

Received 2 September 2022, accepted 12 September 2022, date of publication 26 September 2022, date of current version 11 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210168

## APPLIED RESEARCH

# Computational Methods for Scheduling the Charging and Assignment of an On-Site Shared Electric Vehicle Fleet

JOHANNES VARGA<sup>1</sup>, GÜNTHER R. RAIDL<sup>1</sup>, AND STEFFEN LIMMER<sup>2</sup>

<sup>1</sup>Institute of Logic and Computation, TU Wien, 1040 Vienna, Austria

<sup>2</sup>Honda Research Institute Europe GmbH, 63073 Offenbach, Germany

Corresponding author: Johannes Varga (jvarga@ac.tuwien.ac.at)

This work was supported in part by the Honda Research Institute Europe GmbH, and in part by TU Wien Bibliothek through the Open Access Funding Programme.

**ABSTRACT** We investigate a fleet scheduling problem arising when a company has to manage its own fleet of electric vehicles. Aim is to assign given usage reservations to these vehicles and to devise a suitable charging plan for all vehicles while minimizing a cost function. We formulate the problem as a compact mixed integer linear program, which we strengthen in several ways. As this model is hard to solve in practice, we perform a Benders decomposition, which separates the problem into a master problem and a subproblem and solves them iteratively in an alternating manner. We perform the decomposition in two different ways. First we follow a more classical way, then we enrich the master problem making it stronger but also more complex and the subproblem smaller and simpler to solve. To improve the overall performance, we propose a problem-specific General Variable Neighborhood Search metaheuristic for solving the master problem in earlier iterations. Experimental results show that directly solving the complete mixed integer linear program usually performs well for small to some medium sized problem instances. For larger instances, however, it is not able to find any reasonable primal solutions anymore, while the Benders decomposition scales much better. Especially the variant with the heuristic delivers high quality solutions in reasonable time. The Benders decomposition with the more complex master problem also yields reasonable dual bounds and thus practically relevant quality guarantees for the larger instances.

**INDEX TERMS** Electric vehicles, fleet scheduling, mathematical programming, Benders decomposition, heuristic algorithms.

## I. INTRODUCTION

In many companies employees have to perform business trips on a regular basis, e.g., to visit customers. Usually the company provides a fleet of vehicles that are used for these trips. To utilize these vehicles well, they are most commonly shared among the employees, i.e., not every employee gets his or her own dedicated vehicle. As the overall costs per kilometer of electric cars dropped below the costs of cars with combustion engines [1], it seems reasonable to have a fleet that is primarily composed of electric vehicles. Also the reduction of carbon dioxide, which is targeted by more and

more companies, speaks for an electrification of a company's vehicle fleet.

By having own charging stations and generating energy on-site at the company, e.g., with a photovoltaic system, it is possible to reduce the costs further. Nowadays, the return of investment for photovoltaic systems can be as low as three years, if providing the energy to the grid [2]. Using the generated energy to at least partly charge the own vehicles can further speed up the return of investment.

Long charging times, however, pose a challenge for the scheduling of vehicles within a carsharing system that relies on electric vehicles (EVs). Variable on-site energy generation and time-of-use tariffs for the energy from the grid, i.e., where the energy price is time-dependent, make an effective scheduling even harder. Advanced algorithmic approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao<sup>1</sup>.

are necessary to schedule the charging of a fleet of EVs together with its usage in the sketched context in order to minimize cost while maximizing the availability of the EVs.

We formulate and investigate the EV Fleet Charging and Allocation Problem (EVFCAP) that arises in the carsharing context with EVs and optional on-site energy generation. In this problem we consider a discretized finite time horizon. Given is a fleet of identical EVs and a set of reservations for trips, each having a departure and an arrival time from and back to the site, respectively, and an estimated energy demand. Aim is to assign reservations to vehicles and to determine a feasible charging plan for each vehicle, while minimizing a cost function. It is also possible to leave reservations unassigned, however this is associated with additional cost, e.g., for fulfilling this demand with conventional vehicles with combustion engines or to lend vehicles. In practice the solution methods for the problem can be utilized in a receding horizon approach, in which the scheduling is done for all currently known reservations and a rescheduling is performed whenever a new reservation comes in or there are other changes in the problem input.

Charging can be performed with energy from the grid or surplus energy from the on-site energy generation. For the grid energy a variable time-of-use tariff is assumed. Contrary to grid energy, surplus energy is assumed to not cause any costs as it is the case, e.g., with a photovoltaic system. However only a time-dependent limited amount of surplus energy is available. For simplicity, we do not consider any limit on the maximum power that may be used from the grid or the number of vehicles that can be charged at the same time. However, a respective extension of our models and solution approaches would be straight-forward.

We assume that the time-dependent surplus energy and the energy price are known in advance. Energy prices in time-of-use tariffs are often made public beforehand. If using a photovoltaic plant, the surplus energy can be forecasted using the weather prediction. We consider a myopic scheduling approach, which only considers reservations currently available. If new reservations come in, a rescheduling is supposed to be done.

As we will show later, the problem is NP-hard as well as challenging to solve in practice. As the main contribution of this work we develop several approaches to solve the formalized problem exactly as well as heuristically. We first formulate the problem as Mixed Integer Linear Program (MILP), strengthen it with valid inequalities and, as it turns out to be challenging to solve for larger instances directly, derive a Benders decomposition [3] from this formulation. It is, to our knowledge, the first time a Benders decomposition approach is considered for a problem of this kind. Benders decomposition decomposes a MILP into a master problem (MP) and a subproblem (SP), and both, the MP and the SP are solved iteratively in an alternating fashion. For solving the MP approximately in a faster way, we further propose a heuristic based on General Variable Neighborhood Search (GVNS). For the last iterations, we finally switch from the

GVNS to a MILP solver for solving the MP to ensure exactness of the overall approach. We further investigate a second variant of the Benders decomposition. In the classic Benders decomposition the MP contains all integer variables and all other variables are handled in the SP. In the alternative variant we move some variables from the SP to the MP. This way we can also move some of the constraints of the SP to the MP, which strengthens the MP. As a result, the number of required iterations is reduced, but on the downside, using the GVNS for the MP is not possible anymore, so the MP is solved with the MILP solver in all iterations.

We experimentally compare the two Benders decomposition approaches with the original MILP solved by a state-of-the-art solver on a set of artificially generated benchmark instances. The benchmark instances range from small size with up to 5 vehicles and 80 reservations via medium size with up to 20 vehicles and 320 reservations through to large size with up to 100 vehicles and 1600 reservations. While for small and medium-sized instances the direct solving of the MILP performs well and instances with up to 5 vehicles and 80 reservations can be solved to optimality within one hour, for large instances the Benders decompositions in general perform significantly better. On the one hand, especially the Benders decomposition with the GVNS heuristic is able to find sooner much better heuristic solutions for these instances. For instances with 100 vehicles and 800 reservations the gaps to the best known dual bounds are around 7.5% for the Benders decomposition with heuristic, but around 180% when directly solving with the MILP. On the other hand, the variant of the Benders decomposition with the extended MP is able to identify significantly better dual bounds than the original MILP for our largest instances being up to a factor of 10 lower. This article is based on the first author's master thesis [4], to which we also refer for further details on the implementation and evaluation.

To increase the readability we summarize the most important acronyms and symbols used in this article in Table 1.

The remainder of this article is organized as follows. The next section introduces the EVFCAP in a formal way and states it as MILP. Section III discusses related work. In Section IV the EVFCAP is proven to be NP-hard. Section V derives a class of strengthening inequalities. The two variants of the Benders decomposition are presented in Section VI. Section VII introduces the General Variable Neighborhood Search for solving the MP heuristically in a faster way. Subsequently, Section VIII discusses and compares experimental results of the different approaches on artificial benchmark instances. Finally, Section IX concludes everything and gives an outlook on promising future work.

## II. PROBLEM FORMALIZATION AND MILP MODEL

In the EV Fleet Charging and Allocation Problem (EVFCAP) we are given

- a discretized planning time horizon  $T = \{1, \dots, t_{\max}\}$  with each time step having the same length  $\Delta t$ ;

TABLE 1. Acronyms and symbols.

Acronym / Symbol	Meaning
EVFCAP	Electric Vehicle Fleet Charging and Allocation Problem
GVNS	General Variable Neighborhood Search
ILS	Iterated Local Search
MP	Master Problem
SoC	State of Charge
SP	Subproblem
VND	Variable Neighborhood Descend
VSP	Vehicle Scheduling Problem
$\alpha$	Weight of future costs in the objective function
$c_t, t \in T$	Electricity costs for each time step
$C_v^{\text{feas}}, v \in V$	Feasibility cuts for each vehicle
$C^{\text{opt}}$	Optimality cuts
$c^{\text{uncov}}$	Costs for each unit of energy required by unassigned reservations
$\Delta t$	Duration of one time step
$E^{\text{cap}}$	Battery capacity of an EV
$E_t^{\text{grid}}, t \in T$	Energy consumed from the grid for each time step
$E^{\text{sur}}, t \in T$	Energy consumed from surplus energy for each time step
$E_t^{\text{surmax}}, t \in T$	Predicted available surplus energy for each time step
$E_{v,0}, v \in V$	Initial energy level for each EV
$P^{\text{max}}$	Maximal possible charging power
$R = \{1, \dots, r_{\text{max}}\}$	Set of reservations
$T = \{1, \dots, t_{\text{max}}\}$	Set of time steps
$T_v^{\text{avail}}, v \in V$	Set of time steps in which a vehicle is available
$T^{\text{ends}}$	Set of time steps in which a reservation ends
$T_v^{\text{home}}, v \in V$	Set of time steps in which a vehicle is available and not used for a reservation
$T^{\text{starts}}$	Set of time steps in which a reservation starts
$T_r^{\text{res}} = \{t_r^{\text{start}}, \dots, t_r^{\text{end}}\}, r \in R$	Set of time steps in which a vehicle is needed for each reservation
$V = \{1, \dots, n\}$	Set of vehicles

- a set  $V = \{1, \dots, n\}$  of  $n$  uniform EVs with maximal possible charging power  $P^{\text{max}} > 0$  and battery capacity  $E^{\text{cap}} > 0$ , and for each vehicle  $v \in V$  the subset of time steps  $T_v^{\text{avail}} \subseteq T$  at which the vehicle is available and the initial energy level  $E_{v,0} \in [0, E^{\text{cap}}]$ ;
- a set  $R = \{1, \dots, r_{\text{max}}\}$  of reservations of vehicles, with the set of consecutive time steps  $T_r^{\text{res}} = \{t_r^{\text{start}}, \dots, t_r^{\text{end}}\} \subseteq T$  in which a vehicle is needed for a trip and the expected energy  $E_r^{\text{res}} \geq 0$  that will be (approximately) consumed during the reservation for each  $r \in R$ ; note that this energy demand  $E_r^{\text{res}}$  can be estimated based on a planned route of the trip or, if the destination is unknown, crudely on the duration of the reservation;
- the grid's costs for electricity  $c_t > 0$  per unit of energy during each time step  $t \in T$ ;
- a predicted maximum surplus energy  $E_t^{\text{surmax}} \geq 0$  available during each time step  $t \in T$ , for example from a photovoltaic system;
- and costs  $c^{\text{uncov}} > 0$  for covering each unit of energy required by unassigned reservations, for example by cars with combustion engines.

A solution comprises

- a partial assignment from reservations to vehicles, modeled by variables  $x_{r,v} \in \{0, 1\}$  for  $r \in R, v \in V$ , indicating with value one that reservation  $r$  is to be fulfilled by vehicle  $v$ , and
- a charging plan, modeled by variables  $p_{v,t} \geq 0$  indicating the power by which vehicle  $v \in V$  is charged during time step  $t \in T$ .

The objective is to minimize the costs spent for grid energy and uncovered reservations as well as expected future costs. Expected future costs are formulated as the amount of energy that would be needed to fully charge all vehicles after the last time step weighted with a factor of  $\alpha \geq 0$ . These future costs arise from expected reservations after the time horizon that cannot be covered due to a low State-of-Charge (SoC) of the vehicles.

The EVFCAP can be expressed by the MILP (1)–(15). The total amount of energy consumed from the grid during time step  $t$  is modeled by variables  $E_t^{\text{grid}}$  and the total amount of surplus energy consumed during time step  $t$  by variables  $E_t^{\text{sur}}$ . Moreover, to keep the model clear we use additional variables  $y_r$ , which indicate if reservation  $r \in R$  is uncovered ( $y_r = 1$ ) or covered ( $y_r = 0$ ). Note that these variables can be kept continuous in the model as they will automatically get integral values due to equalities (2). By  $T^{\text{starts}}$  and  $T^{\text{ends}}$  we denote the sets of time steps in which reservations start and end, respectively, i.e.,

$$T^{\text{starts}} := \{t_r^{\text{start}} \mid r \in R\}, \quad T^{\text{ends}} := \{t_r^{\text{end}} \mid r \in R\}.$$

$$\min \sum_{t \in T} c_t E_t^{\text{grid}} + c^{\text{uncov}} \sum_{r \in R} E_r^{\text{res}} y_r + \alpha \sum_{v \in V} (E^{\text{cap}} - E_{v,t_{\text{max}}}) \quad (1)$$

$$\text{s.t.} \sum_{v \in V} x_{r,v} + y_r = 1 \quad r \in R \quad (2)$$

$$x_{r,v} = 0 \quad v \in V, r \in R \mid T_r^{\text{res}} \not\subseteq T_v^{\text{avail}} \quad (3)$$

$$p_{v,t} = 0 \quad v \in V, t \in T \setminus T_v^{\text{avail}} \quad (4)$$

$$p_{v,t} \leq P^{\text{max}} \cdot (1 - \sum_{r \in R \mid t \in T_r^{\text{res}}} x_{r,v}) \quad v \in V, t \in T \quad (5)$$

$$E_{v,t_{\text{max}}} = E_{v,0} + \Delta t \sum_{t \in T} p_{v,t} - \sum_{r \in R} E_r^{\text{res}} x_{r,v} \quad v \in V \quad (6)$$

$$E_{v,0} + \Delta t \sum_{k=1}^t p_{v,k} - \sum_{r \in R \mid t_r^{\text{end}} \leq t-1} E_r^{\text{res}} x_{r,v} \leq E^{\text{cap}} \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\text{max}}\} \quad (7)$$

$$E_{v,0} + \Delta t \sum_{k=1}^{t-1} p_{v,k} - \sum_{r \in R \mid t_r^{\text{start}} \leq t} E_r^{\text{res}} x_{r,v} \geq 0 \quad v \in V, t \in T^{\text{starts}} \quad (8)$$

$$E_t^{\text{grid}} + E_t^{\text{sur}} = \Delta t \sum_{v \in V} p_{v,t} \quad t \in T \quad (9)$$

$$E_t^{\text{grid}} \geq 0 \quad t \in T \quad (10)$$

$$0 \leq E_t^{\text{sur}} \leq E_t^{\text{surmax}} \quad t \in T \quad (11)$$

$$0 \leq E_{v,t_{\text{max}}} \leq E^{\text{cap}} \quad v \in V \quad (12)$$

$$x_{r,v} \in \{0, 1\} \quad v \in V, r \in R \quad (13)$$

$$0 \leq y_r \leq 1 \quad r \in R \quad (14)$$

$$0 \leq p_{v,t} \leq P^{\text{max}} \quad v \in V t \in T \quad (15)$$

The meaning of the constraints is as follows. Equations (2) prohibit the assignment of a reservation  $r$  to multiple EVs, and it forces  $y_r = 1$  if reservation  $r$  is not assigned to a vehicle. Due to (3), reservations cannot be assigned to EVs that are not available in a time step of the reservation interval. Constraints (4) and (5) prevent charging EVs in time steps in which they are not available or are used in reservations. Here the symbol “\” denotes the set difference operator, i.e.,  $T \setminus T_v^{\text{avail}}$  represents all time steps in  $T$  that are not in  $T_v^{\text{avail}}$ . If overlapping reservations are assigned to a vehicle, the right hand side of (5) becomes negative for the time steps in which the reservations overlap. As this is not possible with the domain of  $p_{v,t}$ , constraints (5) also prevent the assignment of overlapping reservations to a vehicle. Note that splitting up the two aspects of (5) into two types of constraints would yield a weaker formulation. Equations (6) define the SoC  $E_{v,t_{\text{max}}}$  of each vehicle after the last time step. Inequalities (7) and (8) determine the SoC of each vehicle for each time step and ensure that it stays in  $[0, E^{\text{cap}}]$ . It is irrelevant for the correctness of the formulation whether the energy of a reservation is subtracted at the start or at the end of the reservation interval or somewhere in between. In order to make the inequalities stronger, we use the end for (7) and the start for (8). Moreover, in (7),  $p_{v,t}$  was added, further strengthening the constraint. Analogously,  $p_{v,t}$  has been omitted in (8). If for some time step the SoC cannot exceed the battery capacity and no reservation ends at this time step, then the SoC also cannot exceed the battery capacity for the previous time step. Therefore it is sufficient to limit the SoC only in time steps before a reservation ends and at the last time step. The SoC can only drop when the energy of a reservation is subtracted. Therefore, (8) is only needed for time steps in which a reservation starts. Equations (9) link the power  $p_{v,t}$  by which the EVs are charged at each time step with the variables for the energy used from the grid  $E_t^{\text{grid}}$  and the used surplus energy  $E_t^{\text{sur}}$ . Finally, all variable domains are defined in (10) to (15). Note that we use continuous values for the charging rates  $p_{v,t}$  as most modern charging infrastructures give this flexibility. In contrast, former work such as [5] often restricts charging to only one or very few power levels, which can ease the mathematical modeling but restricts the possibilities in the scheduling severely. With the larger flexibility of continuously selectable charging rates, we may expect better solutions.

### III. RELATED WORK

We are aware of only two papers addressing problems that are closer related to the EVFCAP. These works are discussed

at the end of this section. Common problems in the literature that are similar to the EVFCAP are Vehicle Scheduling Problems (VSPs). In these a set of trips, each having a start and end location and a time for departure and arrival, is given [6]. The aim is to find a feasible assignment from trips to vehicles that minimizes costs. Vehicles start and end at depots. Problem variants either consider a single depot or multiple depots. The traditional single-depot VSP is solvable in polynomial time, e.g. by using a network flow formulation [7]. When considering multiple depots however, the VSP gets NP-hard [8]. Recently a new variant of the VSP arose that considers an electric fleet of vehicles. The electric VSP has been solved exactly by utilizing MILP solvers [9], [10], [11], and with various heuristics such as Adaptive Large Neighborhood Search [12], GRASP and a memetic algorithm [13]. The EVFCAP can be seen as a variant of the electric VSP with a single depot, where all trips start and end at the depot and charging is only possible at the depot. Note however that the availability of surplus energy and the possibility that trips are not covered have not been considered in works to the electric VSP we are aware of. Furthermore having multiple locations substantially restricts the solution space, since vehicles have to start the next trip where the previous one ended. Thus VSPs are quite different from our problem and solution methods and results can hardly be compared.

Different kinds of works regarding EVs concentrate on the charging process of a vehicle fleet. On the one hand this concerns smart charging of a vehicle fleet in the context of smart grids, see, e.g., [14] or [15] for a survey from the algorithmic perspective and [16] for a review that additionally considers photovoltaic power production. A variety of problems arise in this context from the different perspectives of the grid operator, the aggregator and the EV user. Proposed algorithms are often distributed to increase scalability and reduce communication overhead. On the other hand non-linearities of the charging process have been considered and investigated. For instance and in particular when considering fast charging, the maximum power that can be used to charge a vehicle depends significantly on its current SoC. This is, for example, considered for the following charging problem from the literature. A single vehicle or a fleet of vehicles is given. Each vehicle has a current SoC, a minimum required SoC, an SoC-dependent maximal charging power, and possibly a departure time. Aim is to find a charging schedule, s.t. the vehicles reach the required SoCs by their departure times or the end of the planning horizon, respectively. Applied solving approaches are the cutting plane method for convex maximal charging power functions [17], [18], different MILP formulations [18], [19], as well as mixed integer nonlinear programming [20].

Although the EVFCAP has overlaps with the above problems, we are only aware of two strongly related works. The following paragraphs give an overview on these.

Sassi and Oulamara [5] consider a problem variant that differs from our EVFCAP setting primarily in the optimization objective. First, they maximize the distance that is traveled

with EVs. Then the charging costs are minimized, s.t. the distance stays constant on its optimal value. Another difference is that the number of alternative vehicles that can be used for uncovered reservations, e.g., cars with combustion engines, is limited. Moreover, there are also some differences regarding the charging process: Surplus energy is not considered, i.e., all the energy is taken from the grid. However the grid is limited in the total power it can deliver, so the charging processes of different EVs must also be optimized together. Additionally to the maximum power of the grid and the maximum charging power for each EV, there also exists a lower power limit for charging. Thus, an EV is either charged within its lower and upper power limits or not charged.

The authors propose two heuristics to solve their problem, which they name Sequential Heuristic (SH) and Global Heuristic (GH). The SH repeats the following steps for each EV:

- Select a subset of reservations that are non-conflicting s.t. the driven distance is as large as possible and assign them to the EV.
- Find a charging schedule for the EV that minimizes charging costs.

Afterwards, uncovered reservations are assigned to the alternative vehicles. The GH works basically the same way but calculates a charging schedule for all vehicles at once after all reservations have been assigned to vehicles. These two heuristics were experimentally compared to a MILP on random instances. Both, GH and SH, outperformed the MILP on these instances in terms of the quality of heuristic solutions. Compared to SH, GH in general found feasible solutions on more instances and better solutions on the instances where both heuristics found solutions. However, SH was significantly faster.

Betz *et al.* [21] consider a problem variant that differs from our EVFCAP by considering a limited number of available chargers. These chargers can have different characteristics and are located at specific parking lots. If necessary, relocations of vehicles between parking lots are possible but cause additional cost. Betz *et al.* also formulate their problem as a MILP and solve it using the MATLAB integrated MILP solver. The hardest instance that they could solve within two hours included 30 requests and eight vehicles. Moreover, the authors also investigate the effect on operational costs and environmental impact when using a fleet of EVs compared to a fleet of vehicles with combustion engines.

#### IV. NP-HARDNESS OF THE EVFCAP

Sassi and Oulamara [5] tried to prove the NP-hardness of their problem variant which they called Electric Vehicle Scheduling and Charging Problem (EVSCP). As the EVFCAP is similar to the EVSCP, the basic idea of their proof can be adopted for the EVFCAP. However, we found that the proof in [5] contains a severe error because the reduction from the used partition problem may result in an exponentially large EVSCP instance as the size of the time horizon  $T$  depends on

the input values of the partition problem. In the following we correct this issue and adapt the proof for the EVFCAP.

In our proof, we reduce the partition problem, which is known to be NP-complete [22], to the decision variant of the EVFCAP. The decision variant of the EVFCAP asks whether or not there is a solution that has an objective value less than or equal to a given  $c_{\max}$ .

The partition problem can be stated as follows.

**PARTITION PROBLEM**

**Input:**

A multiset  $P = \{a_1, \dots, a_m\}$  of positive integers that sum up to  $2s$  with  $s \in \mathbb{N}$ .

**Question:**

Is there a partition  $\{M_1, M_2\}$  of  $\{1, \dots, m\}$  s.t.

$$\sum_{i \in M_1} a_i = s = \sum_{i \in M_2} a_i?$$

*Theorem 1:* The EVFCAP is NP-hard.

*Proof:* Given an instance of the partition problem, an instance of the decision variant of the EVFCAP is constructed as follows.

- For each integer in  $P$  there is a time step.
- There are two EVs. Both EVs have a battery capacity and an initial energy level of  $s$ , no charging capabilities and an unlimited availability.
- For each integer in  $P$  there is a reservation. All reservations are non-overlapping and the energy consumption of reservation  $r$  is  $a_r$ .
- There are only costs for uncovered reservations.
- The costs has to be less than or equal to zero.
- No surplus energy is available.

Assume we have a solution for this problem instance. This solution cannot have uncovered reservations as they would increase the objective value to a positive value. Therefore, the two EVs have to cover all reservations. The total energy consumption of all reservations is  $2s$ . Both vehicles have an initial energy level of  $s$  and no recharging possibilities. Therefore the energy level of both vehicles is zero after time step  $m$ , and the energy consumptions of the reservations served by a single vehicle sum up to  $s$ . This gives a solution for the corresponding partition problem which is

$$M_v := \{i \mid x_{i,v} = 1, i = 1, \dots, m\} \quad v \in V. \quad (16)$$

Given a solution to the partition problem, a corresponding solution to the EVFCAP is constructed by assigning the reservations corresponding to  $M_v$  to vehicle  $v$  for both  $v \in V$ . Due to a similar reasoning as above, the solution to the EVFCAP has to be feasible.

Overall, there is a solution to the partition problem, iff there is a solution to the corresponding EVFCAP. As all the described transformations can be done in polynomial time, it follows that the EVFCAP is NP-hard.  $\square$

#### V. STRENGTHENING INEQUALITIES FOR CONFLICTING RESERVATIONS

We now introduce strengthening inequalities for the MILP model (1)–(15) for the EVFCAP, which prevent the assignment of sets of reservations to a vehicle if those reservations

cannot be assigned together to the vehicle due to charging constraints. Let  $R' \subseteq R$  be a set of non-overlapping reservations and assume that these reservations are considered for joined assignment to vehicle  $v \in V$ . The maximal achievable energy level  $E_{v,t}^{\max}(R')$  of vehicle  $v$  for  $t \in T \cup \{0\}$  in dependence of  $R'$  can be calculated by

$$E_{v,t}^{\max}(R') = \begin{cases} E_{v,0} & \text{if } t = 0 \\ \min(E_{v,t-1}^{\max}(R') + P^{\max} \Delta t, E^{\text{cap}}) & \text{if } t \in T_v^{\text{avail}} \setminus \bigcup_{r \in R'} T_r^{\text{res}} \\ E_{v,t-1}^{\max}(R') - \sum_{r \in R' | t = t_r^{\text{start}}} E_r^{\text{res}} & \text{otherwise.} \end{cases} \quad (17)$$

If  $E_{v,t}^{\max}(R')$  is negative at a time step, not all reservations of  $R'$  can be assigned together to vehicle  $v$ . In other words at least one reservation in  $R'$  may not be assigned to vehicle  $v$  in this case, which yields the inequality

$$\sum_{r \in R'} x_{r,v} \leq |R'| - 1. \quad (18)$$

This kind of inequality in general strengthens the MILP, i.e., there are solutions to the linear programming (LP) relaxation of the model that do not fulfill (18). Take for example the problem instance with one vehicle of capacity and initial SoC 3 and two reservations 1 and 2 requiring energy of  $E_1^{\text{res}} = E_2^{\text{res}} = 2$  in time steps  $T_1^{\text{res}} = \{1\}$  and  $T_2^{\text{res}} = \{2\}$ , respectively. Then the LP relaxation of the MILP has a solution with  $x_{1,1} = x_{2,1} = 0.75$ . However,  $E_{1,2}^{\max}(\{1, 2\}) = -1$  is less than zero and inequality 18 is violated for  $R' = \{1, 2\}$ . Its addition to the MILP therefore indeed strengthens the LP relaxation.

Clearly, the number of these inequalities in general is exponentially large in  $|R|$ . Therefore we consider the dynamic separation of such inequalities in a branch-and-cut approach. The respective separation problem can be stated for a specific vehicle  $v \in V$  as follows.

#### SEPARATION PROBLEM

##### Input:

Let  $x_{r,v}^{\text{LP}}$  be the values of variables  $x_{r,v}$  in the solution of the LP relaxation for all  $r \in R$ . Moreover, we use  $E_r^{\text{res}}, T_r^{\text{res}}, P^{\max}, \Delta t, E^{\text{cap}}, E_{v,0}$  and  $T_v^{\text{avail}}$ .

##### Question:

Does there exist a non-overlapping set of reservations  $R' \subseteq R$  with  $\bigcup_{r \in R'} T_r^{\text{res}} \subseteq T_v^{\text{avail}}$ , s.t.  $E_{v,t}^{\max}(R') < 0$  for some  $t \in T \setminus \{1\}$  and  $\sum_{r \in R'} x_{r,v}^{\text{LP}} > |R'| - 1$ ?

For the special case of  $P^{\max} = 0$ ,  $E^{\text{cap}}$  sufficiently large, and unlimited availabilities ( $T_v^{\text{avail}} = T$ ), the maximal achievable energy can be calculated in an easy way by

$$E_{v,t}^{\max}(R') = E_{v,0} - \sum_{r \in R' | t_r^{\text{start}} \leq t} E_r^{\text{res}}. \quad (19)$$

The condition  $E_{v,t}^{\max}(R') < 0$  then becomes  $\sum_{r \in R'} E_r^{\text{res}} > E_{v,0}$  as it is sufficient to check for  $t = t_{\text{max}}$ . If no reservations in  $R$  overlap with each other, the separation problem corresponds

to (the decision version of) the 0–1 knapsack problem with item weights  $1 - x_{r,v}^{\text{LP}}$ , capacity one, and item values  $E_r^{\text{res}}$ . Therefore the separation problem is NP-hard.

One way to deal with the separation problem in practice is to discretize all values  $x_{r,v}^{\text{LP}}$  and to apply dynamic programming over  $\sum_{r \in R'} (1 - x_{r,v}^{\text{LP}})$  with the reservations ordered by their start times, maximizing  $E_{v,t}^{\max}(R')$  for each intermediate result. Alternatively, one could discretize the energy and minimize  $\sum_{r \in R'} (1 - x_{r,v}^{\text{LP}})$ .

In practice it seems reasonable to apply a fast greedy heuristic to possibly identify a violated inequality and the more complex dynamic programming approach only when the heuristic does not succeed. We implemented a greedy heuristic and tested it with the MILP solved by Gurobi<sup>1</sup> in version 9.1. However this did not lead to a speedup in practice, most likely due to the strong generic cuts Gurobi already considers. Therefore we did not use this approach in further steps.

Alternatively, one can initially check all pairs of non-overlapping reservations and provide respective valid inequalities already from the beginning instead of later separating them dynamically. We also tried this approach but again could not observe a practically relevant speedup.

## VI. BENDERS DECOMPOSITION

Benders decomposition is a general approach to tackle MILPs of a certain structure and was proposed almost six decades ago by Benders [3]. It works by splitting up the problem into a master problem (MP) and a subproblem (SP). For details regarding the Benders decomposition in general, variants of it as well as different speedup strategies and extensions we refer to the review by Rahmaniani *et al.* [23].

First we apply the Benders decomposition to our problem in a more classical way. Afterwards we strengthen the MP by moving variables from the SP to the MP. In order to ultimately have the whole objective function in this extended MP, we use a reformulated but equivalent objective function. To derive this reformulation from the original objective function (1) the following steps are performed besides basic simplifications:

- Replace  $E_{v,t_{\text{max}}}$  according to (6).
- Replace  $\Delta t \sum_{v \in V} p_{v,t}$  with  $E_t^{\text{grid}} + E_t^{\text{sur}}$  according to (9).
- Replace  $\sum_{v \in V} x_{r,v}$  with  $1 - y_r$  according to (2).

This leads to the expression

$$\sum_{t \in T} (c_t - \alpha) E_t^{\text{grid}} + (c^{\text{uncov}} - \alpha) \sum_{r \in R} E_r^{\text{res}} y_r - \alpha \sum_{t \in T} E_t^{\text{sur}} + \alpha \sum_{v \in V} (E^{\text{cap}} - E_{v,0}) + \alpha \sum_{r \in R} E_r^{\text{res}}. \quad (20)$$

As the last two terms are constant, we abbreviate them by

$$\tilde{c}^{\text{const}} := \alpha \sum_{v \in V} (E^{\text{cap}} - E_{v,0}) + \alpha \sum_{r \in R} E_r^{\text{res}}, \quad (21)$$

further introduce the new cost factors

$$\tilde{c}_t := c_t - \alpha \quad \text{and} \quad \tilde{c}^{\text{uncov}} := c^{\text{uncov}} - \alpha, \quad (22)$$

<sup>1</sup><https://www.gurobi.com/>

and arrive at the reformulated objective function

$$\min \sum_{t \in T} \tilde{c}_t E_t^{\text{grid}} + \tilde{c}^{\text{uncov}} \sum_{r \in R} E_r^{\text{res}} y_r - \alpha \sum_{t \in T} E_t^{\text{sur}} + \tilde{c}^{\text{const}}. \quad (23)$$

Variables  $E_{v,t_{\max}}$  are not needed anymore when using this reformulation and can therefore be removed together with their defining equation (6).

### A. BASIC BENDERS DECOMPOSITION

The following master problem (MP) only considers the assignment variables  $x_{r,v}$ ,  $r \in R$ ,  $v \in V$  and  $y_r$ ,  $r \in R$  and abstracts from the actual charging details. The latter will only be dealt with in the subproblem. The subproblem, however, yields feasibility cuts and optimality cuts which are added to the MP over time to ultimately account for the charging.

$$\text{(MP)} \quad \min \tilde{c}^{\text{uncov}} \sum_{r \in R} E_r^{\text{res}} y_r + \mu + \tilde{c}^{\text{const}} \quad (24)$$

$$\text{s.t.} \quad \sum_{v \in V} x_{r,v} + y_r = 1 \quad r \in R \quad (25)$$

$$x_{r,v} = 0 \quad v \in V, r \in R \mid T_r^{\text{res}} \not\subseteq T_v^{\text{avail}} \quad (26)$$

$$\sum_{r \in R \mid t \in T_r^{\text{res}}} x_{r,v} \leq 1 \quad v \in V, t \in T \quad (27)$$

$$\sum_{r \in R} \gamma_{c,r} x_{r,v} + \gamma'_c \leq 0 \quad v \in V, c \in C_v^{\text{feas}} \quad (28)$$

$$\sum_{r \in R} \sum_{v \in V} \beta_{c,r,v} x_{r,v} + \beta'_c \leq \mu \quad c \in C^{\text{opt}} \quad (29)$$

$$\mu \geq \sum_{t \in T \mid \tilde{c}_t < 0} \tilde{c}_t n \Delta t P^{\text{max}} - \alpha \sum_{t \in T} E_t^{\text{surmax}} \quad (30)$$

$$x_{r,v} \in \{0, 1\} \quad (31)$$

$$0 \leq y_r \leq 1 \quad (32)$$

Feasibility cuts (28) are formulated independently for each vehicle and thus the set of feasibility cuts  $C^{\text{feas}} = \bigcup_{v \in V} C_v^{\text{feas}}$  is partitioned into smaller sets, each dedicated to one vehicle. A feasibility cut is represented by the vehicle  $v$  it is dedicated to, a vector  $\gamma_c$  of coefficients for  $x$ , and a constant  $\gamma'_c$ . In (29), each optimality cut  $c \in C^{\text{opt}}$  is represented by a coefficient matrix  $\beta_c$  of  $x$  and a constant  $\beta'_c$ . Hereby, variable  $\mu$  represents additional costs for the charging that are to be contributed by the optimality cuts (29). Initially,  $C^{\text{feas}}$  and  $C^{\text{opt}}$  are empty. Equation (30) prevents the MP to be unbounded before optimality cuts have been added by providing a trivial lower bound for  $\mu$ .

The subproblem is determined for a given solution  $\bar{x} = (\bar{x}_{r,v})_{r \in R, v \in V}$  to the MP. We define

- for each vehicle  $v \in V$  the set of time steps when the vehicle is available and not reserved

$$T_v^{\text{home}} := T_v^{\text{avail}} \setminus \bigcup_{r \in R \mid \bar{x}_{r,v} = 1} T_r^{\text{res}} \quad (33)$$

- and for each vehicle  $v \in V$  the energy difference up to time step  $t \in T$  caused by assigned reservations. The

energy of a reservation may be subtracted at the start or the end of the reservation interval, leading to the two definitions

$$E_{v,t}^{\text{res,start}} := \sum_{r \in R \mid t \leq t_r^{\text{start}} \wedge \bar{x}_{r,v} = 1} E_r^{\text{res}} \quad \text{and} \quad (34)$$

$$E_{v,t}^{\text{res,end}} := \sum_{r \in R \mid t \leq t_r^{\text{end}} \wedge \bar{x}_{r,v} = 1} E_r^{\text{res}}. \quad (35)$$

The primal subproblem  $\text{SP}(\bar{x})$  can now be stated as the following LP.

$$\text{(SP}(\bar{x})) \quad \min \sum_{t \in T} \tilde{c}_t E_t^{\text{grid}} - \alpha \sum_{t \in T} E_t^{\text{sur}} \quad (36)$$

$$\text{s.t.} \quad p_{v,t} = 0 \quad v \in V, t \in T \setminus T_v^{\text{home}} \quad (37)$$

$$E_{v,0} + \Delta t \sum_{k=1}^t p_{v,k} - E_{v,t-1}^{\text{res,end}} \leq E^{\text{cap}} \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\max}\} \quad (38)$$

$$E_{v,0} + \Delta t \sum_{k=1}^{t-1} p_{v,k} - E_{v,t}^{\text{res,start}} \geq 0 \quad v \in V, t \in T^{\text{starts}} \quad (39)$$

$$E_t^{\text{grid}} + E_t^{\text{sur}} = \sum_{v \in V} \Delta t p_{v,t} \quad t \in T \quad (40)$$

$$E_t^{\text{grid}} \geq 0 \quad t \in T \quad (41)$$

$$0 \leq E_t^{\text{sur}} \leq E_t^{\text{surmax}} \quad t \in T \quad (42)$$

$$0 \leq p_{v,t} \leq P^{\text{max}} \quad v \in V, t \in T \quad (43)$$

Further feasibility cuts (28) and optimality cuts (29) are determined by solving the dual SP for specific reservation assignments  $\bar{x} = (\bar{x}_{r,v})_{r \in R, v \in V}$  of a current solution to the MP. More specifically, feasibility cuts are inferred from extreme rays of the dual SP in case this problem is unbounded. Since feasibility cuts can be formulated independently for each vehicle, identifying such cuts is more efficient due to the smaller problem size. Optimality cuts are inferred from extreme points of the dual subproblem. For the dual subproblem and the mathematical derivation of the cuts see Appendix A.

We further remark that it is also possible to formulate the SP as Minimum Cost Flow Problem. We did that and compared the Minimum Cost Flow Problem formulation solved with the network simplex implementation of CPLEX<sup>2</sup> in version 12.8. to the LP formulation solved by Gurobi. As it turned out, the network simplex approach was in practice not faster than solving the LP with Gurobi. Therefore we use Gurobi to solve the SP in all further steps.

### B. EXTENSION OF THE MP

By adding meaningful information about the SP to the MP from the beginning, we may expect to reduce the number of

<sup>2</sup><https://www.ibm.com/analytics/cplex-optimizer>

iterations and, as long as solving the MP does not become substantially harder, the overall runtime. We do this by moving the variables  $E_t^{\text{grid}}$  and  $E_t^{\text{sur}}$ , for all  $t \in T$ , from the SP into the MP. The number of these variables is low enough (in  $\mathcal{O}(|T|)$ ) to still keep the MP reasonably small, but now it becomes possible to add further strengthening inequalities to the MP that make use of  $E_t^{\text{grid}}$  and  $E_t^{\text{sur}}$ .

Besides moving the variables  $E_t^{\text{grid}}$  and  $E_t^{\text{sur}}$  from the SP into the MP, the MP is updated as follows to the formulation (44)–(56) shown in Figure 1.

- As  $E_t^{\text{grid}}$  and  $E_t^{\text{sur}}$  are available now, the extended MP can use the complete reformulated objective function (23).
- Summing inequalities (7) and (8) over the vehicles yields constraints, which do not contain  $p_{v,t}$  anymore and therefore can be added to the extended MP.
- Optimality cuts are not necessary anymore, as the objective function of the extended MP already comprises all parts of the original objective.
- Feasibility cuts (52) also contain the variables  $E_t^{\text{grid}}$  and  $E_t^{\text{sur}}$ . Note however that now, they cannot be formulated independently for each vehicle anymore.

These changes in the MP cause also some changes in the subproblem and the derivation of the feasibility cuts for which we refer again to Appendix A. Also note that it is still possible to add the feasibility cuts (51) of the basic Benders decomposition, which we will also do in our experiments. Recall that the feasibility cuts of the basic Benders decomposition can be formulated independently for each vehicle, which is not possible after extending the MP. Thus the feasibility cuts of the basic Benders decomposition are not necessarily weaker than the feasibility cuts of the Benders decomposition with the extended MP.

### C. SOLVING THE MP

In the classical Benders decomposition approach, the MP is solved with a MILP solver to optimality in each iteration. The process of iteratively solving the MP, the dual SP, and adding derived Benders cuts to the MP eventually stops with a proven optimal solution to the original problem when no further cuts can be derived, given sufficient time and memory. During each iteration, the dual bound derived from the solution to the MP is only valid also for the original problem if the MP was solved to optimality. Solving the MP always to proven optimality is in general time consuming and, at least in earlier iterations, unnecessary, if afterwards a cut is added, especially as in practice typically a large portion of the time is required to prove the optimality of a found solution. We examine two different strategies to speed up the overall Benders decomposition approach.

On the one hand we turn the classical scheme into a *Branch-and-Check* approach. Here, the MP is only solved once by a MILP solver, but whenever a new feasible solution is found within this search, the respective SP is solved for this solution and obtained feasibility/optimality cuts are

dynamically added to the MP. For more information on Branch-and-Check we refer to [24].

On the other hand we speed up the classical Benders decomposition approach by solving the MP first in an inexact way and only switch to an exact approach at a later point when no further cuts can be found by the respective SPs. In this way, the whole procedure stays exact in the end. This idea has already successfully been applied in the literature multiple times, see for example [25] or [26]. In our first inexact approach, we solve the MP by the MILP solver terminating early when a solution is obtained with an optimality gap within a certain specified limit. We will see that, although a speedup is achieved for the individual iterations, this ultimately does not lead to substantially better final optimality gaps as more iterations are required. We therefore also consider a metaheuristic approach for solving the MP, which is described in the following section.

## VII. GENERAL VARIABLE NEIGHBORHOOD SEARCH HEURISTIC

We suggest a General Variable Neighborhood Search (GVNS) [27] as a heuristic to solve the MP. The GVNS metaheuristic combines basic Variable Neighborhood Search based on shaking moves in larger neighborhoods with Variable Neighborhood Descent (VND). Variable Neighborhood Descent is a generalization of local search that utilizes a sequence of multiple neighborhood structures. Upon finding a better solution in one neighborhood, VND restarts with the first neighborhood, otherwise it continues with the next neighborhood. The eventually returned solution thus is a local optimum w.r.t. all used neighborhood structures. In the outer Variable Neighborhood Search, a shaking operator takes a random solution from a typically larger neighborhood  $N_k \in \{N_1, \dots, N_{k_{\max}}\}$  of the current solution. Algorithm 1 shows a pseudo code of GVNS.

---

### Algorithm 1: General Variable Neighborhood Search

---

```

x ← initial solution;
repeat
    k ← 1;
    repeat
        x' ← random element in Nk(x);
        x'' ← VND(x');
        if x'' better than x then
            x ← x'';
            k ← 1;
        else
            k ← k + 1;
        end
    until k > kmax;
until stopping criteria satisfied;
return x

```

---

The specific components of our GVNS for the MP are described in the following. Note that this metaheuristic is dedicated only to the first Benders decomposition variant



$$\begin{aligned}
 \min \quad & \sum_{t \in T} \tilde{c}_t E_t^{\text{grid}} + \tilde{c}^{\text{uncov}} \sum_{r \in R} E_r^{\text{res}} y_r - \alpha \sum_{t \in T} E_t^{\text{sur}} + \tilde{c}^{\text{const}} & (44) \\
 \text{s.t.} \quad & \sum_{v \in V} x_{r,v} + y_r = 1 & r \in R \quad (45) \\
 & x_{r,v} = 0 & v \in V, r \in R \mid T_r^{\text{res}} \not\subseteq T_v^{\text{avail}} \quad (46) \\
 & \sum_{r \in R \mid t \in T_r^{\text{res}}} x_{r,v} \leq 1 & v \in V, t \in T \quad (47) \\
 & E_t^{\text{grid}} + E_t^{\text{sur}} \leq \sum_{v \in V} \Delta t P^{\text{max}} \left( 1 - \sum_{r \in R \mid t \in T_r^{\text{res}}} x_{r,v} \right) & t \in T \quad (48) \\
 & \sum_{k=1}^t (E_k^{\text{grid}} + E_k^{\text{sur}}) + \sum_{v \in V} \left( E_{v,0} - \sum_{r \in R \mid t_r^{\text{end}} \leq t-1} E_r^{\text{res}} x_{r,v} \right) \leq n E^{\text{cap}} & t \in T^{\text{ends}} \cup \{t_{\text{max}}\} \quad (49) \\
 & \sum_{k=1}^{t-1} (E_k^{\text{grid}} + E_k^{\text{sur}}) + \sum_{v \in V} \left( E_{v,0} - \sum_{r \in R \mid t_r^{\text{start}} \leq t} E_r^{\text{res}} x_{r,v} \right) \geq 0 & t \in T^{\text{starts}} \quad (50) \\
 & \sum_{r \in R} \gamma_{c,r} x_{r,v} + \gamma'_c \leq 0 & v \in V, c \in C_v^{\text{feas}} \quad (51) \\
 & \sum_{r \in R} \sum_{v \in V} \gamma_{c,r,v}^{\text{ext}} x_{r,v} + \sum_{t \in T} (\gamma_{c,t}^{\text{grid}} E_t^{\text{grid}} + \gamma_{c,t}^{\text{sur}} E_t^{\text{sur}}) + \gamma_c^{\text{ext}} \leq 0 & c \in C^{\text{feas,ext}} \quad (52) \\
 & E_t^{\text{grid}} \geq 0 & t \in T \quad (53) \\
 & 0 \leq E_t^{\text{sur}} \leq E_t^{\text{surmax}} & t \in T \quad (54) \\
 & x_{r,v} \in \{0, 1\} & (55) \\
 & 0 \leq y_r \leq 1 & (56)
 \end{aligned}$$

FIGURE 1. Extended master problem after moving the variables  $E^{\text{grid}}$  and  $E^{\text{sur}}$  from the SP to the MP.

whose MP (24)–(32) contains the variables  $x_{r,v}$  and  $y_r$  only. This is because only then it is possible to reasonably speed up searching the neighborhoods with delta evaluation. Delta evaluation computes the objective value of a neighbor by efficiently computing the difference to the objective value of the current solution based on the move that was used to obtain the neighbor. Due to the possibility to apply delta evaluation we expect the GVNS heuristic to work well for the MP.

### A. MP INSTANCES

Remember that an instance of the MP consists of

- the instance of the EVFCAP as well as
- a set of feasibility cuts and
- a set of optimality cuts.

For each optimality cut the factors  $\beta_c$  of  $x$  and the constant  $\beta'_c$  are stored. Since the matrix  $\beta_c$  usually is dense, all coefficients of the matrix are stored. The same is done for feasibility cuts, with the difference that each cut is targeted to a specific vehicle  $v$  and therefore also the vehicle has to be stored besides the factors  $\gamma_c$  and the constant  $\gamma'_c$ .

### B. SOLUTION REPRESENTATION

Solutions of the MP are represented by the assignment of reservations to vehicles. Within the GVNS, such an assignment is stored as vector  $(a_r)_{r \in R} \in (V \cup \{\varepsilon\})^{|R|}$ , where  $a_r$

indicates the vehicle by which reservation  $r$  is to be fulfilled or  $\varepsilon$  if no EV is assigned to this reservation. Note that this representation is more compact and efficient to use than the binary representation with the  $x_{r,v}$  and  $y_v$  variables from the MILP formulation.

To make delta evaluation possible, the GVNS stores additional data for its candidate solutions. In particular

- the objective value that corresponds to the assignment,
- for each optimality and each feasibility cut the value in which the left hand side of the respective inequality results, and
- for each vehicle the set of time steps in which the vehicle is used in reservations

are stored. Suppose an MP solution  $a$  and another solution  $a'$  that only differs from  $a$  in the assignment of one reservation are given. Utilizing the additionally stored data for solution  $a$  it is possible to decide the feasibility and calculate the objective value of solution  $a'$  in time  $\mathcal{O}(|C^{\text{feas}}| + |C^{\text{opt}}| + |T|)$ .

### C. VND NEIGHBORHOOD STRUCTURES

The VND starts with the trivial solution that does not assign any reservations and uses two different neighborhood structures, based on the following moves.

- 1) Change the assignment of any single reservation. This neighborhood can, for a given solution  $a$ ,

be written as

$$N_1(a) = \{a' \in F \mid \Delta(a, a') = 1\}, \quad (57)$$

where  $\Delta$  refers to the hamming distance between the two vectors. Set  $F$  contains all feasible assignments, i.e., assignments that neither assign overlapping reservations to the same vehicle nor invalidate Benders cuts. Note that also  $a_r = \varepsilon$  or  $a'_r = \varepsilon$  is possible for the reservation  $r \in R$  whose assignment is changed. The size of this neighborhood is in  $|N_1(a)| = \mathcal{O}(|R| \cdot |V|)$ .

2) Swap the assignments of any single pair of reservations:

$$N_2(a) = \{a' \in F \mid a' = a \circ \sigma_{rr'}, r, r' \in R\}, \quad (58)$$

where permutation  $\sigma_{rr'} \in S_R$  swaps  $r$  with  $r'$  and leaves all other elements at their positions, and  $F$  denotes the set of feasible assignments again. The size of this neighborhood is in  $|N_2(a)| = \mathcal{O}(|R|^2)$ .

For the instances we considered  $|N_2(a)|$  is generally larger than  $|N_1(a)|$ , and therefore we apply  $N_2$  after  $N_1$  in the VND. A more general version of the second neighborhood that considers more than two reservations was also considered but turned out to be too time consuming for larger instances and did not yield substantially better results. Therefore we did not use it further.

All moves can be decomposed into alterations of single assignments of a reservation. Thus efficient delta evaluation can be applied to search the neighborhoods that result from the given moves.

#### D. SHAKING

Shaking is used to escape local optima and to diversify the search. In our case, shaking deletes assignments of a number of reservations, i.e., sets the respective  $a_r$  variables to  $\varepsilon$ . This allows the VND afterwards to in general converge to a different solution. Which assignments are deleted is determined in two different ways. In the first shaking method a fixed number of reservations is selected randomly and all their assignments are deleted, and in the second shaking method a fixed number of vehicles is selected randomly and all assignments to these are deleted.

In preliminary experiments we found that the following three shaking configurations, in this order, are performing generally well on the kind of instances we used in our tests.

- Deleting five random assignments from reservations to vehicles.
- Deleting ten random assignments from reservations to vehicles.
- Deleting all assignments to one random vehicle.

We therefore used these in all further experiments.

### VIII. EXPERIMENTS AND RESULTS

We first describe how we created artificial benchmark instances as actual real world instances were not available to us. Then, the results of solving the original MILP as well as the different Benders decomposition variants are presented and compared.

#### A. BENCHMARK INSTANCES

The benchmark instances were created randomly, with properties of a real-world application at the Honda Research Institute in mind. We consider the following combinations of the number of time steps  $t_{\max}$  and the number of EVs  $n$ :

- $t_{\max} = 32$  and  $n \in \{1, 2, 5\}$
- $t_{\max} = 192$  and  $n \in \{5, 10, 20\}$
- $t_{\max} = 768$  and  $n \in \{20, 50, 100\}$

For each of these combinations we consider  $r_{\max} \in \{4n, 8n, 16n\}$  reservations. The other input values are set as follows.

- $\Delta t = 15\text{min}$ , i.e.,  $T$  corresponds to 8 hours, 2 days, or 8 days.
- The maximum charging power is  $P^{\max} = 3.3\text{ kW}$  and the battery capacity is  $E^{\text{cap}} = 20\text{ kWh}$ .
- For each vehicle  $v \in V$ ,  $T_v^{\text{avail}} = T$  and  $E_{v,0}$  is selected uniformly at random from  $[0, E^{\text{cap}}]$ .
- For each reservation  $r \in R$  the reservation energy  $E_r^{\text{res}}$  is selected uniformly at random from  $[0, E^{\text{cap}}]$ . To determine the reservation interval its length is selected uniformly at random from  $[1, \frac{t_{\max}}{4}]$  then the time interval is placed randomly within the bounds 1 and  $t_{\max}$ ; actual starting and ending time steps are obtained by rounding down and up, respectively.
- Data on energy prices and surplus energies came from the Honda Research Institute. They were also randomly generated but according to models mimicking some real expectations such as a day-night-cycle and the usage of a photovoltaic system. The available surplus energy is scaled according to the number of vehicles to obtain meaningful instances.
- Costs for uncovered reservations and the factor  $\alpha$  in the objective function are set to  $c^{\text{uncov}} = 150 \frac{\text{Yen}}{\text{kWh}}$  and  $\alpha = 75 \frac{\text{Yen}}{\text{kWh}}$ .

Thirty instances were created for each combination of  $t_{\max}$ ,  $n$ , and  $r_{\max}$ , yielding 810 instances in total, and we made them available online.<sup>3</sup> Some of the more specific tests were performed on a reduced instance set consisting of only the first ten instances of each instance size to keep the computational requirements reasonable. This will be indicated by referring to the reduced instance set.

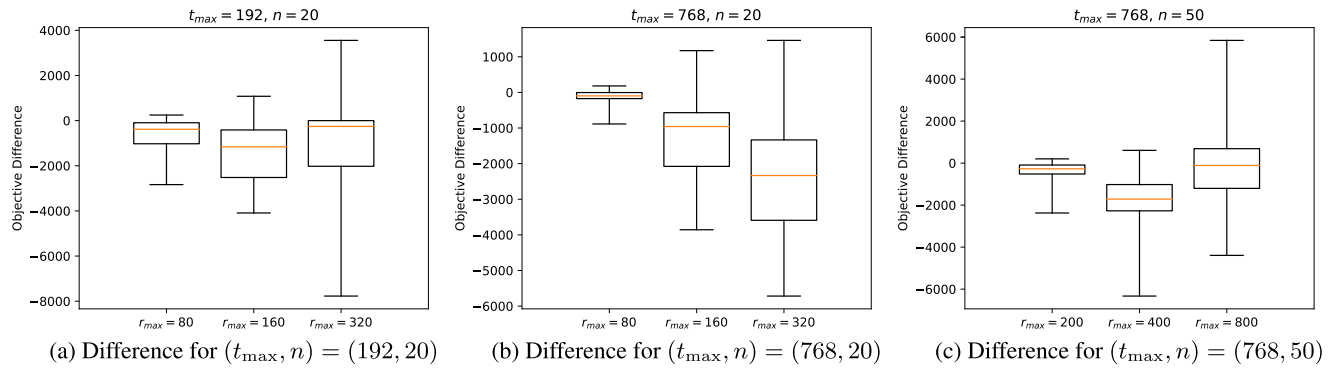
#### B. IMPLEMENTATION AND COMPUTING ENVIRONMENT

We used the MILP solver Gurobi 9.1<sup>4</sup> [28], which is a leading commercial product. The models as well as the Benders decomposition approach were implemented in Julia 1.6<sup>5</sup> [29]. The Julia package JuMP was used as interface to the Gurobi [30]. Each experiment was performed on a single core of an Intel Xeon E5-2640 v4 with a time limit of one hour. We furthermore imposed a memory limit of 36 GB. The time given to the GVNS heuristic is limited to the time that was needed to solve the SP in the previous iteration.

<sup>3</sup><https://www.ac.tuwien.ac.at/research/problem-instances/#evfcap>

<sup>4</sup><https://www.gurobi.com>

<sup>5</sup><https://julialang.org>



**FIGURE 2.** Difference in the objective values of solutions obtained from GVNS and ILS. A negative value means that the objective value of the GVNS is lower (better) than the value of the ILS.

We solved the test instances with the introduced approaches in different configurations. These configurations and the results are presented in the following.

### C. SOLVING THE MP

Before coming to results directly concerning the Benders decomposition we compare our GVNS to an Iterated Local Search (ILS), which uses the same neighborhood structures as the GVNS for the local search but replaces the more advanced shaking of the GVNS with a perturbation operation that always removes the assignment of 70 reservations in the perturbation step. The value of 70 was determined as a robust choice by a preliminary experimental tuning on independent problem instances. To generate instances for the comparison we solved all instances of the EVFCAP with *BDH* and of each run we selected one MP instance.

The results of the comparison for a selection of instance sizes are shown in Figure 2. We gave the heuristics five, ten, and 50 seconds for the sizes  $(t_{\max}, n) = (192, 20)$ ,  $(t_{\max}, n) = (768, 20)$ , and  $(t_{\max}, n) = (768, 50)$ , respectively, similar to the time they would have when running as part of the Benders decomposition. Each boxplot shows the differences between the final objective values obtained by the GVNS and the ILS. Negative values mean that the objective value from the GVNS is lower and therefore better. As can be seen the GVNS performs slightly better on almost all the instance sizes with a median difference in the objective value of around 2500 for  $(t_{\max}, n, r_{\max}) = (768, 20, 320)$ . But as we will see later, these are still rather minor differences – both approaches can be said to perform reasonably well.

### D. IMPACTS OF INDIVIDUAL IMPROVEMENTS

For the rest of this section we will refer to the Benders decomposition with *BD* if the MP is solved to optimality in each iteration, with *BDG* if the MP is solved by Gurobi only up to a certain gap limit in the first iterations, and with *BDH* if the MP is solved by the GVNS heuristic in the first iterations. Concerning *BDG*, we remark that in our implementation the constant offset  $\tilde{c}^{\text{const}}$  in the reformulated objective function was not considered as it does not affect the search space and the feasibility and optimality of solutions; thus, the gap-limit

for the early termination was determined with  $\tilde{c}^{\text{const}} = 0$ . However, in all results listed in the following, the correct values for  $\tilde{c}^{\text{const}}$  as calculated from the original instances are taken into account. Superscripts to *BD*, *BDG*, or *BDH* denote the following:

- ext: the extended MP is used
- cuts: the inequalities from Section V for pairs of vehicles are added from the beginning to the MP
- bnch: Branch-and-Check was used instead of the classical Benders decomposition scheme

We first evaluate for *BDG*<sup>ext</sup> the impact on the number of solved instances, i.e. instances for which a feasible solution was found by the approach, and final optimality gaps when allowing different gap-limits for solving the MP in the beginning. The results for the reduced test set and two selected instance sizes are shown in Figure 3. Note that if the gap is zero the approach is identical to *BD*. As can be seen, only one instance out of ten can be solved with *BD* for the large instance size, but more than five instances out of ten can be solved for any of the other evaluated MP gaps. The number of solved instances increases slightly with the used gap, rising from six solved instances for 0.5% MP gap to nine solved instances for 10% MP gap. Contrary to this, the final gap gets worse with an increased MP gap for smaller instance sizes. As a compromise we select an MP gap of 2% for further experiments.

Next we compare *BDG*<sup>ext</sup> to *BDG*<sup>ext,cuts</sup>, *BDG*<sup>ext,bnch</sup>, *BDG*, and *BDH*. All experiments were conducted on the reduced test set. A comparison of the number of iterations respectively number of solved instances between *BDG*<sup>ext</sup> and *BDG*<sup>ext,cuts</sup> on a small, medium-sized, and large instance size can be seen in Figure 4. The number of iterations on the small instances is up to 15% lower for *BDG*<sup>ext,cuts</sup> and this also leads to a lower runtime. However the version that does not add static cuts in the beginning is able to solve more instances of large size. Especially for size  $(t_{\max}, n, r_{\max}) = (768, 100, 1600)$  *BDG*<sup>ext</sup> can solve eight instances out of ten while *BDG*<sup>ext,cuts</sup> cannot solve any instance. We are more interested in the performance on large instances and therefore we chose to not add these cuts for the further experiments.

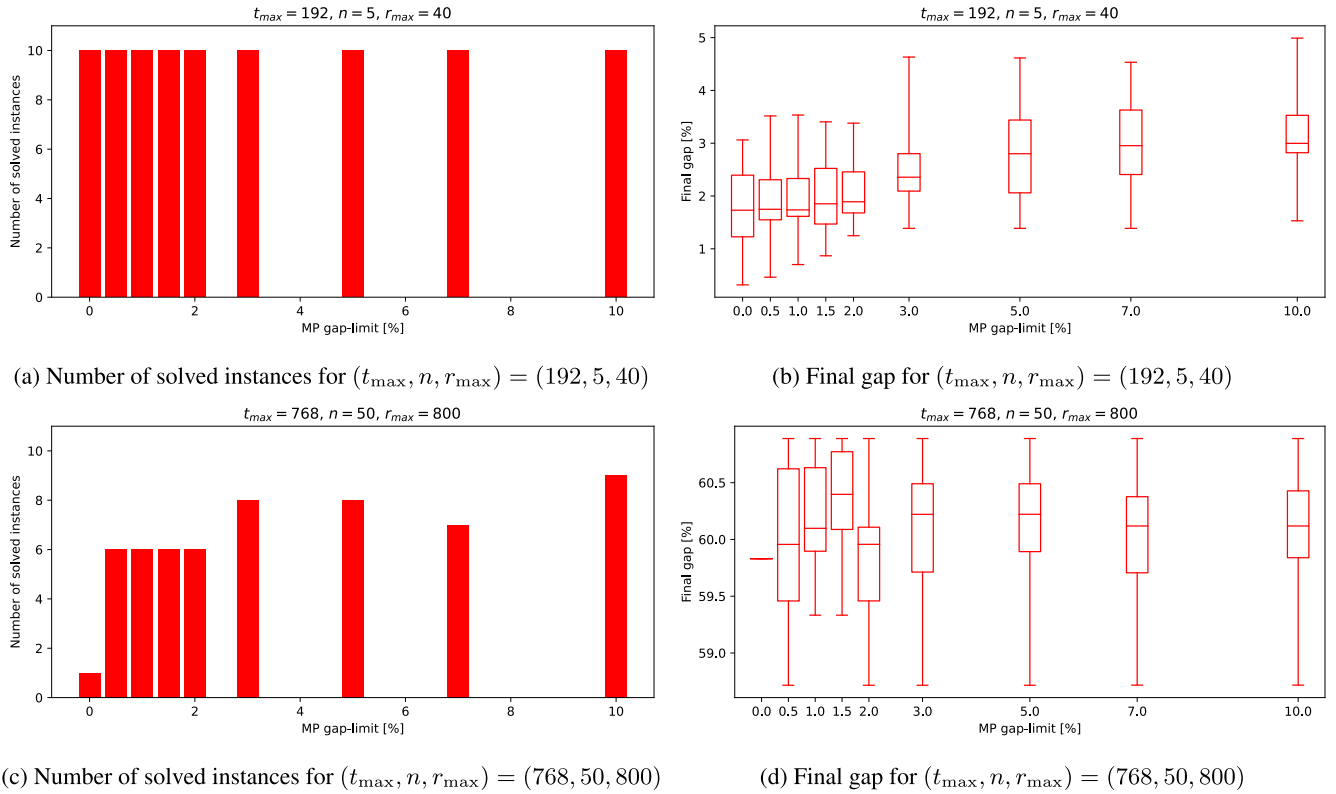


FIGURE 3. Numbers of solved instances and final gaps for different MP gap-limits in  $BDG^{\text{ext}}$  for instances of two sizes.

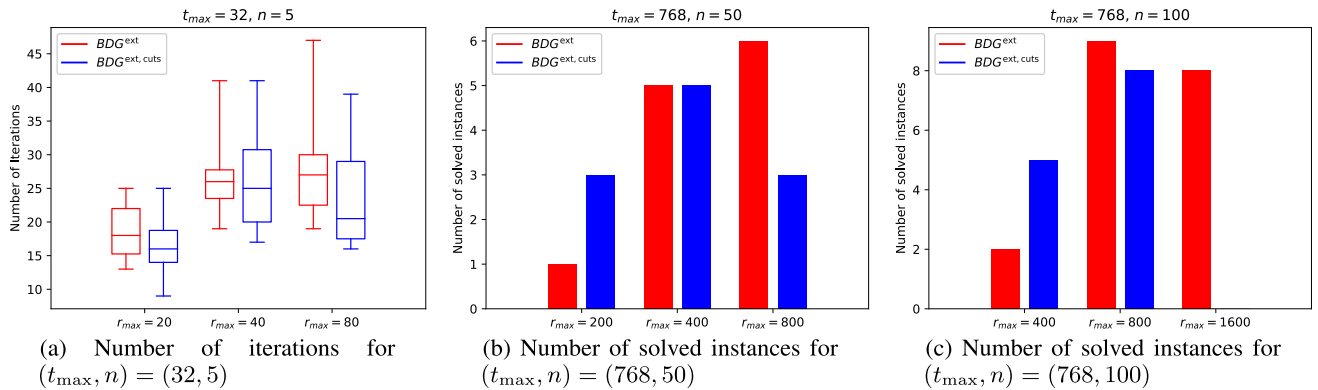


FIGURE 4. Comparison of  $BDG^{\text{ext}}$  with and without additionally adding the static cuts from Section V.

Figure 5 shows a comparison of runtime, final gap, number of iterations respectively callback calls, and the number of solved instances between  $BDG^{\text{ext}}$  and  $BDG^{\text{ext},\text{bnch}}$ . Note that the number of callback calls in  $BDG^{\text{ext},\text{bnch}}$  correspond to the number of so far feasible solutions for which the respective dual SP was solved in order to possibly obtain further cuts. We can see that the number of callback calls in  $BDG^{\text{ext},\text{bnch}}$  is significantly higher than the number of iterations in  $BDG^{\text{ext}}$  for most of the instances. While for small instances  $BDG^{\text{ext},\text{bnch}}$  is faster than  $BDG^{\text{ext}}$  by a factor of two to three, the gap is larger by a factor of up to four on some of the medium-sized instances and no large instances with  $r_{\max} \geq 800$  were solved by  $BDG^{\text{ext},\text{bnch}}$  whereas  $BDG^{\text{ext}}$  is able to solve most of these instances.

The reason for the worse performance of  $BDG^{\text{ext},\text{bnch}}$  on larger instances is that the SP needs to be solved much more often and with many inferior solutions, which wastes more of time.

We now investigate the impact on the dual bound when shifting the variables  $E^{\text{grid}}$  and  $E^{\text{sur}}$  from the SP to the MP as done in the extended MP. Figure 6 compares the dual bounds produced by  $BDG^{\text{ext}}$  and  $BDG$  on some selected instance sizes. Since the objective function is minimized, the higher the dual bound the better. While the dual bounds are similar for smaller instances,  $BDG^{\text{ext}}$  gives significantly better dual bounds for larger instances, especially when the ratio between  $r_{\max}$  and  $n$  is large. For  $r_{\max} \geq 400$   $BDG$  is only able to give negative dual bounds, which are not reasonable because

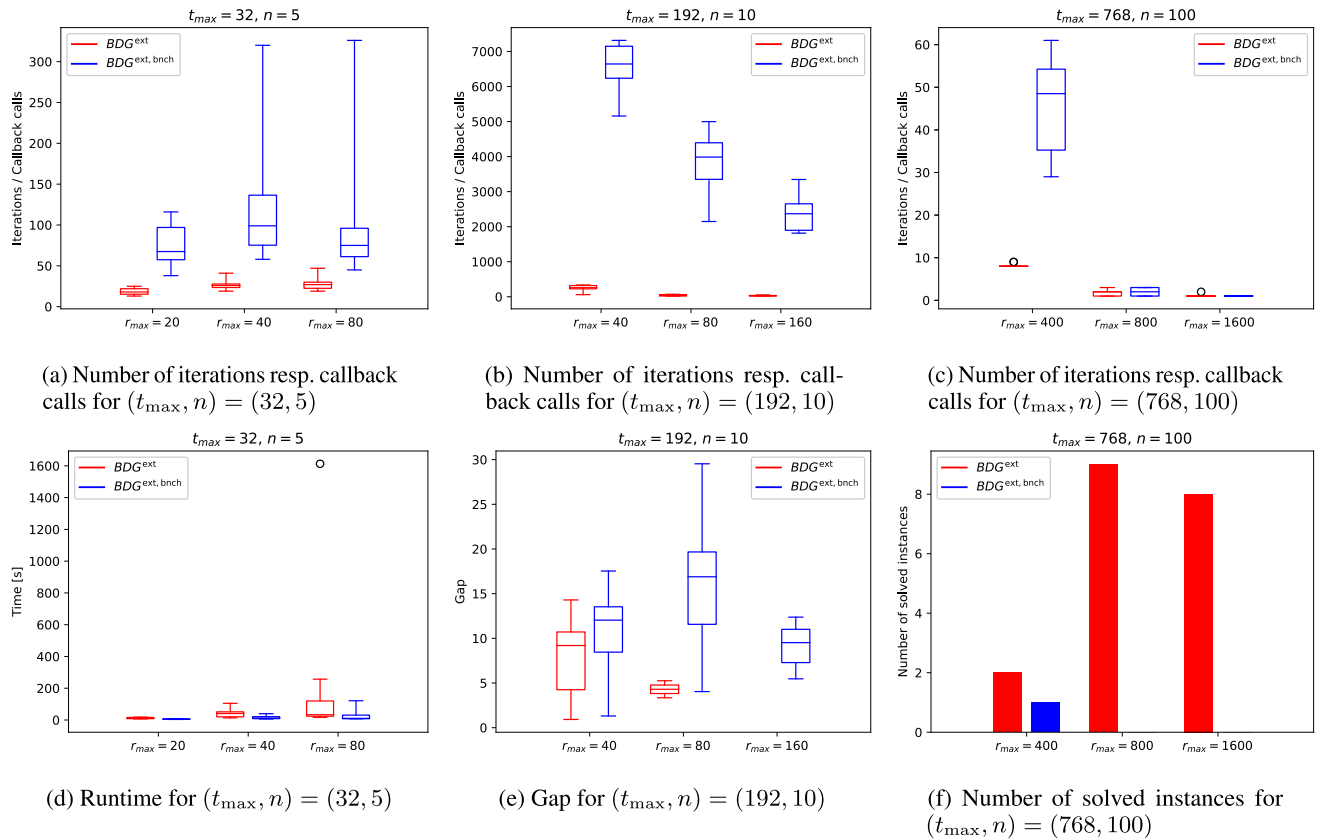


FIGURE 5. Comparison of  $BDG^{\text{ext}}$  with the branch-and-check variant  $BDG^{\text{ext, bnch}}$ .

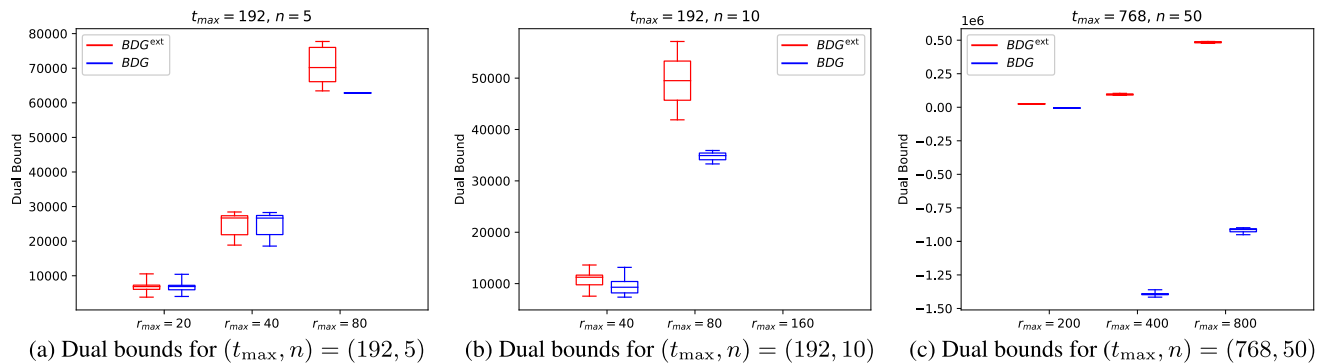


FIGURE 6. Comparison of the dual bounds obtained from  $BDG^{\text{ext}}$  and  $BDG$ .

the original objective function (1) cannot yield values less than zero. Recall that the Benders decomposition is based on the reformulated objective function (23), and the trivial dual bound for the MP with this objective function can be negative. The variant  $BDG^{\text{ext}}$  is able to find positive dual bounds on these instances and we will see in the next subsection that these bounds are nearly optimal.

Finally we also investigate the impact of using the GVNS heuristic for the MP in the first iterations. Figure 7 compares  $BDH$  with  $BDG^{\text{ext}}$  on different instance sizes. On most of the small instances with  $t_{\max} = 32$  the runtime for  $BDH$  is by 50 to 100 % larger than for  $BDG^{\text{ext}}$ . Contrary to this, both, the number of solved instances and the objective values

are in general significantly better for larger instances. For  $(t_{\max}, n) = (192, 20)$ , e.g.,  $BDG^{\text{ext}}$  is not able to solve any instances with  $r_{\max} \geq 160$  while  $BDH$  solves them all. Also the objective values obtained by  $BDG^{\text{ext}}$  for instances with  $(t_{\max}, n) = (768, 50)$  are up to a factor of nine higher than the ones obtained by  $BDH$ . In other words Gurobi is not able to find any reasonable solutions for the MP for these large instances, while the GVNS performs quite well.

Of all the considered improvements, the ones discussed in the previous two paragraphs are most notable. Moving the variables  $E^{\text{grid}}$  and  $E^{\text{sur}}$  from the SP to the MP seems to significantly improve the resulting dual bound on large

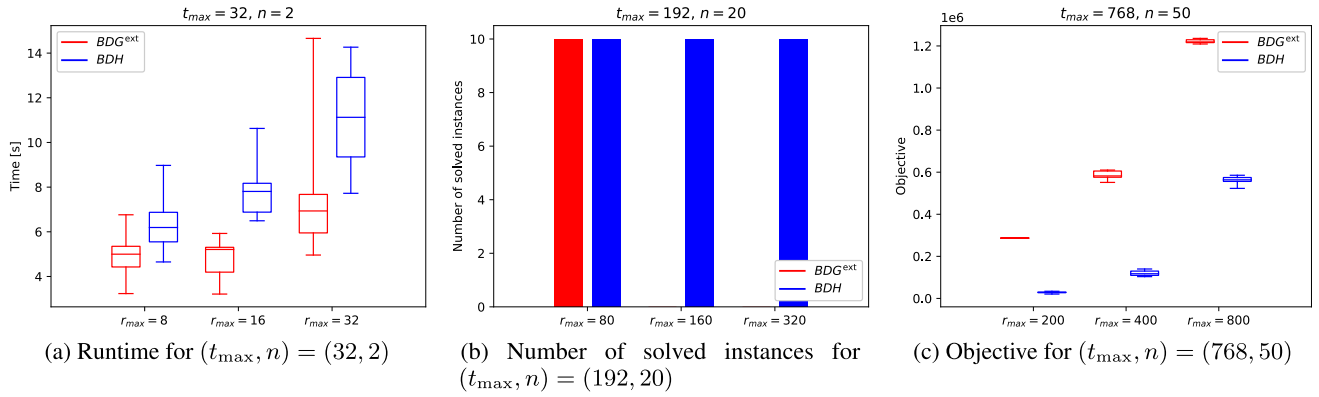


FIGURE 7. Comparison of  $BDG^{ext}$  and  $BDH$ .

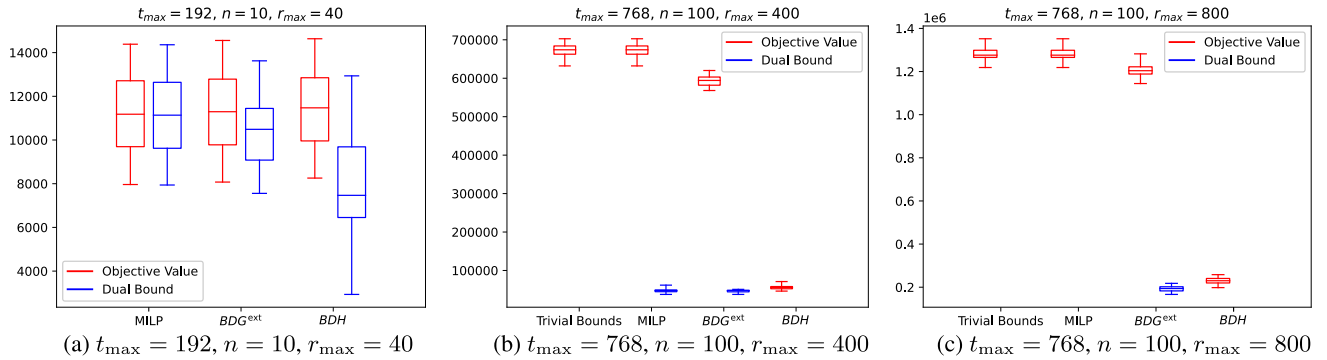


FIGURE 8. Comparison of final objective values and dual bounds of the different methods for three selected instance sizes. Trivial primal bounds are obtained from the solutions that leave all reservations unassigned. The trivial dual bounds in 8b and 8c are calculated by the right hand side expression in (30).

instances. Similarly using the heuristic for the MP seems to significantly improve the objective value for large instances. Therefore, in the following we will compare in more detail the two approaches  $BDG^{ext}$  and  $BDH$  with each other and with the MILP formulation (1)–(15) directly solved by Gurobi.

### E. COMPARISON OF THE MILP AND THE BENDERS DECOMPOSITION APPROACHES

As we saw in the previous section, especially the Benders decomposition variants  $BDG^{ext}$  and  $BDH$  are most promising. We therefore evaluate and compare these with the direct solving of the original MILP, however, also using the reformulated objective (23) to get rid of the variables  $E_{v,t_{max}}$ , on the whole benchmark instance set. For short we abbreviate the latter direct MILP solving just by MILP. Detailed results are given in Table 2. These results are visualized for three selected instance sizes in Figure 8. Each boxplot represents the final objective values respectively dual bounds for a specific approach applied to the test instances of the given size. Let us first look at Figure 8a, where the approaches are compared for the medium sized instances with ten EVs and 40 reservations. The MILP outperforms the two Benders decomposition approaches for these instances with a median gap of 0.1% compared to median gaps of 3.3% and 13.5%, respectively. This changes for larger instance sizes as can be seen in Figure 8b and Figure 8c, where the

approaches are compared on instances with 100 EVs and 400 and 800 reservations, respectively. Here  $BDH$  provides excellent solutions with gaps to the dual bounds found by  $BDG^{ext}$  around 15% while the other two approaches only find solutions that are not much better than the trivial solutions that leave all reservations unassigned. These insights coincide with those of Sassi and Oulamara [5]. They also find that the MILP formulation solved by a solver does not lead to (reasonable) solutions for large instances, whereas their heuristic is able to find good solutions for these instances.  $BDG^{ext}$  gives good dual bounds while  $BDH$  finds worse or no dual bounds. The original MILP only yields a meaningful dual bound for some of the instances, while for others like the ones with  $r_{max} = 800$  it only finds poor trivial dual bounds. When looking at the detailed results in the table, they confirm that for large instances  $BDH$  finds the best solutions, while  $BDG^{ext}$  finds the best dual bounds. One exception are the largest instances with  $(t_{max}, n, r_{max}) = (768, 100, 1600)$  where  $BDH$  does not find feasible solutions. The reason is that not enough iterations are performed to get to an MP solution that is also feasible in respect to the charging constraints.  $BDG^{ext}$  performs better here because the MP solutions are worse but already feasible with charging constraints from the beginning.

We also want to point out that the fraction between  $r_{max}$  and  $n$ , which indicates whether the system is overbooked, has a high influence on the runtime and gap. This can be seen

**TABLE 2.** Comparison of the MILP,  $BDG^{ext}$  and  $BDH$ . The number of feasibly solved instances / optimally solved instances, the median final objective values, and the median dual bounds are listed. Bold numbers indicate the best values among the three compared approaches.

$t_{max}$	$n$	$r_{max}$	Solved			Objective Value			Dual Bound		
			MILP	$BDG^{ext}$	$BDH$	MILP	$BDG^{ext}$	$BDH$	MILP	$BDG^{ext}$	$BDH$
32	1	4	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>4 590</b>	<b>4 590</b>	<b>4 590</b>	<b>4 590</b>	<b>4 590</b>	<b>4 590</b>
32	1	8	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>9 761</b>	<b>9 761</b>	<b>9 761</b>	<b>9 761</b>	<b>9 761</b>	<b>9 761</b>
32	1	16	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>20 856</b>	<b>20 856</b>	<b>20 856</b>	<b>20 856</b>	<b>20 856</b>	<b>20 856</b>
32	2	8	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>8 966</b>	<b>8 966</b>	<b>8 966</b>	<b>8 966</b>	<b>8 966</b>	<b>8 966</b>
32	2	16	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>20 516</b>	<b>20 516</b>	<b>20 516</b>	<b>20 516</b>	<b>20 516</b>	<b>20 516</b>
32	2	32	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>45 650</b>	<b>45 650</b>	<b>45 650</b>	<b>45 650</b>	<b>45 650</b>	<b>45 650</b>
32	5	20	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>18 699</b>	<b>18 699</b>	<b>18 699</b>	<b>18 699</b>	<b>18 699</b>	<b>18 699</b>
32	5	40	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 30</b>	<b>47 557</b>	<b>47 557</b>	<b>47 557</b>	<b>47 557</b>	<b>47 557</b>	<b>47 557</b>
32	5	80	<b>30 / 30</b>	<b>30 / 30</b>	<b>30 / 29</b>	<b>106 759</b>	<b>106 759</b>	107 225	<b>106 755</b>	<b>106 756</b>	106 677
192	5	20	<b>30 / 30</b>	<b>30 / 0</b>	<b>30 / 1</b>	<b>7 129</b>	<b>7 129</b>	<b>7 129</b>	<b>7 128</b>	6 716	7 023
192	5	40	<b>30 / 30</b>	<b>30 / 1</b>	<b>30 / 1</b>	<b>24 662</b>	24 683	24 695	<b>24 661</b>	24 125	24 010
192	5	80	<b>30 / 16</b>	21 / 0	<b>30 / 0</b>	<b>67 618</b>	70 485	69 126	67 454	<b>69 726</b>	66 087
192	10	40	<b>30 / 4</b>	<b>30 / 0</b>	<b>30 / 0</b>	<b>11 180</b>	11 294	11 472	<b>11 136</b>	10 488	7 467
192	10	80	<b>30 / 0</b>	8 / 0	<b>30 / 0</b>	<b>44 207</b>	46 301	47 622	41 838	<b>44 069</b>	32 387
192	10	160	<b>30 / 0</b>	0 / 0	<b>30 / 0</b>	<b>144 460</b>	–	154 820	<b>140 230</b>	–	123 208
192	20	80	<b>30 / 1</b>	29 / 0	<b>30 / 0</b>	<b>20 955</b>	21 624	22 523	<b>20 601</b>	19 448	773
192	20	160	<b>30 / 0</b>	0 / 0	<b>30 / 0</b>	<b>92 296</b>	–	95 321	<b>82 077</b>	–	–
192	20	320	<b>30 / 0</b>	0 / 0	<b>30 / 0</b>	327 636	–	<b>301 269</b>	<b>272 335</b>	–	–
768	20	80	<b>30 / 15</b>	<b>30 / 0</b>	<b>30 / 0</b>	<b>11 200</b>	11 330	11 700	<b>11 045</b>	10 596	2 528
768	20	160	<b>30 / 0</b>	4 / 0	<b>30 / 0</b>	176 397	248 346	<b>53 030</b>	<b>44 733</b>	42 540	19 754
768	20	320	<b>30 / 0</b>	0 / 0	<b>30 / 0</b>	494 045	–	<b>218 802</b>	<b>-441 719</b>	–	–
768	50	200	<b>30 / 0</b>	3 / 0	<b>30 / 0</b>	<b>27 341</b>	286 716	29 285	<b>25 650</b>	24 318	-19 281
768	50	400	<b>30 / 0</b>	13 / 0	<b>30 / 0</b>	637 527	593 116	<b>118 576</b>	-1 400 677	<b>97 239</b>	–
768	50	800	<b>30 / 0</b>	20 / 0	<b>30 / 0</b>	1 250 581	1 215 269	<b>553 960</b>	-1 087 250	<b>479 707</b>	–
768	100	400	<b>30 / 0</b>	9 / 0	29 / 0	674 014	594 281	<b>55 971</b>	45 924	<b>47 832</b>	-65 293
768	100	800	<b>30 / 0</b>	29 / 0	19 / 0	1 276 038	1 203 941	<b>230 068</b>	-2 796 330	<b>193 888</b>	–
768	100	1600	<b>29 / 0</b>	28 / 0	0 / 0	2 474 916	<b>2 400 555</b>	–	-2 202 808	<b>924 414</b>	–

**TABLE 3.** Mean objective values of solutions obtained by  $BDH$  and their composition in respect of the three terms of the objective function (1).

$t_{max}$	$n$	$r_{max}$	Tot. objective value in Yen	Grid costs in Yen	Costs for uncovered reservations in Yen	Future costs in Yen
32	1	4	4 764	440	3 803	520
32	1	8	9 738	419	8 262	1 056
32	1	16	21 073	464	19 518	1 090
32	2	8	8 657	875	5 981	1 800
32	2	16	20 235	946	16 990	2 298
32	2	32	45 443	996	41 902	2 544
32	5	20	18 388	2 250	10 792	5 344
32	5	40	47 151	2 359	38 585	6 205
32	5	80	106 212	2 494	97 011	6 705
192	5	20	7 002	4 048	2 064	888
192	5	40	24 515	6 313	16 499	1 702
192	5	80	70 162	8 378	59 344	2 439
192	10	40	11 568	8 491	1 198	1 878
192	10	80	47 941	12 712	30 653	4 575
192	10	160	153 965	15 293	132 416	6 255
192	20	80	23 230	17 410	1 239	4 581
192	20	160	94 389	25 692	58 589	10 107
192	20	320	301 505	30 681	256 722	14 101
768	20	80	11 152	9 946	328	877
768	20	160	54 292	22 335	29 788	2 168
768	20	320	219 676	34 624	181 571	3 480
768	50	200	28 446	25 952	26	2 467
768	50	400	120 398	56 977	58 734	4 686
768	50	800	555 499	90 029	455 969	9 500
768	100	400	55 744	50 023	0	5 721
768	100	800	229 726	120 569	97 257	11 899
768	100	1600	–	–	–	–

particularly well for  $BDG^{ext}$  for the instances with  $n = 10$  and  $n = 20$ . Here, solutions to almost all instances with  $r_{max} = 4n$

could be found, but not many solutions for  $r_{max} = 8n$  and none for  $r_{max} = 16n$ .

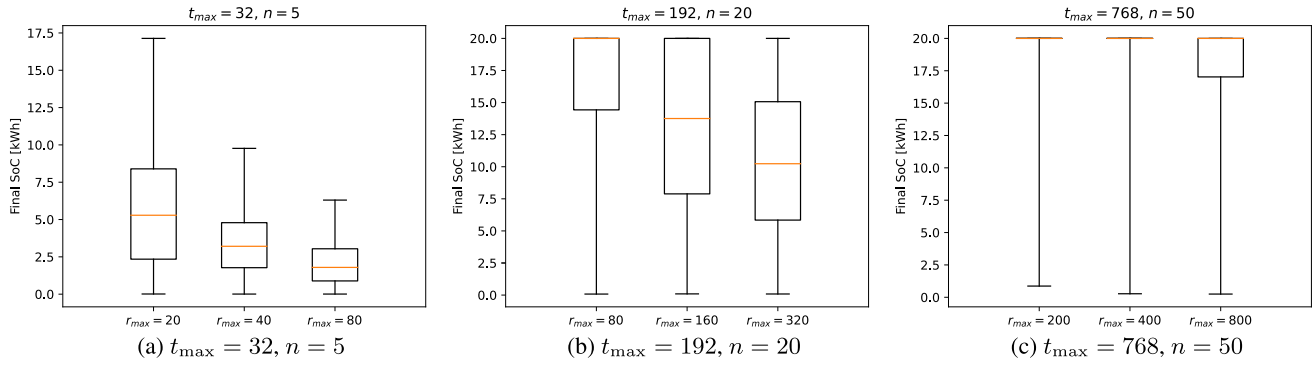


FIGURE 9. Final SoC of EVs for some selected instance sizes.

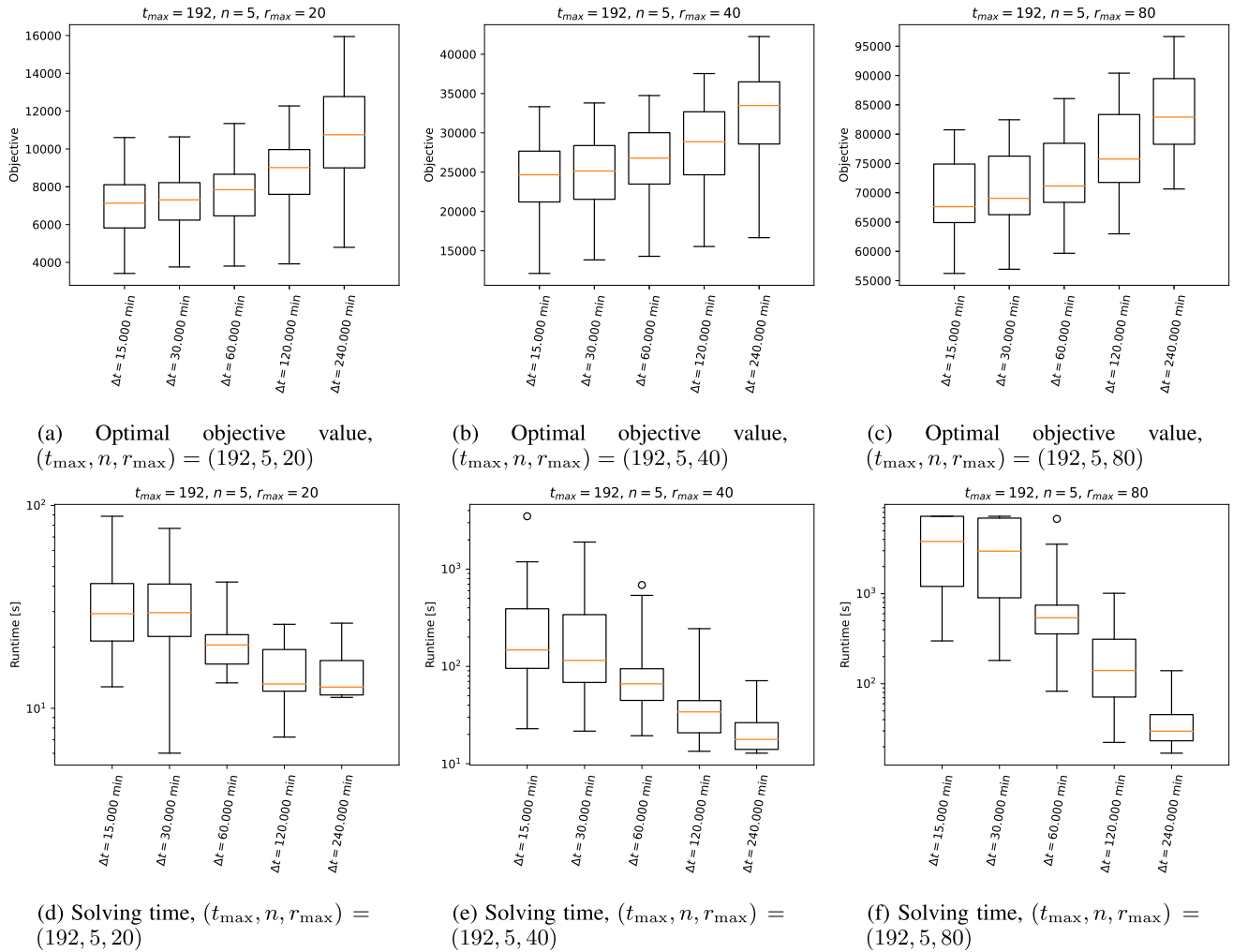


FIGURE 10. Impact of changing  $\Delta t$  on the objective value and the solving time when directly applying the MILP solver.

Last but not least, Table 3 breaks the mean objective values down into the values of its terms, i.e., the costs for energy taken from the grid, the costs for uncovered reservations and future costs. The solutions of *BDH* were taken for that purpose, since they are consistently near optimal. Naturally the costs for uncovered reservations dominate the two other terms if  $r_{max} = 16n$  since then the system is overbooked and most of the reservations cannot be served with the available EVs.

Furthermore the larger the instances get, the lower gets the ratio of costs for uncovered reservations versus other costs, provided that the ratio between  $r_{max}$  and  $n$  stays the same. While for instances with  $(t_{max}, n, r_{max}) = (768, 100, 400)$  these costs are zero, for instances with  $(t_{max}, n, r_{max}) = (32, 1, 4)$  costs for uncovered vehicles are roughly four times the other costs together. We believe the reason for this behavior is that in larger instances random fluctuations become



less relevant and there are more possibilities to compensate for them and thus more reservations can be assigned to EVs. In practice this means that a larger system reaches a higher utilization of EVs. It should also be mentioned that in general, future costs do not rise in proportion to the number of EVs. For example, set  $r_{\max} = 8n$ ; then the average future costs are 1056 for  $n = 1$ , 4575 for  $n = 10$  and 11899 for  $n = 100$ . A larger system therefore also tends to be able to leave the EVs at a higher SoC at the end of the time horizon. This can also be seen in more detail in Figure 9, where each boxplot represents the final SoCs of all vehicles of solutions to instances of a specific instance size, when solved with *BDH*. The vehicles in the solutions to all instances with  $t_{\max} = 192$  or  $t_{\max} = 768$  have a median final SoC of more than half of the battery capacity and can therefore be considered well prepared to serve reservations of the next time horizon when expecting a similar demand of requests. In a schedule of the next time horizon EVs can already serve reservations at the beginning leading to a higher utilization of the EVs. We furthermore observe that a higher ratio between the number of reservations and the number of vehicles leads to a lower final SoC, which is an expected behavior.

#### F. IMPACT OF CHANGING THE DEGREE OF TIME DISCRETIZATION

Finally we investigated the impact on the solution quality and hardness of the instances when using coarser time steps. To obtain a coarser instance from an existing instance we arranged the time steps in blocks of 2, 4, 8, and 16 time steps, respectively, and merged the time steps of each block into one. Reservation intervals and vehicle availabilities are extended to span whole blocks, surplus energies of time steps in a block are added up, and for the electricity cost the mean value of the cost of the individual time steps in the block is used. The derived problem instances are more restrictive in the sense that solutions to them can be transformed into feasible solutions to the original instance but not vice versa. Therefore these coarser instances have worse or equally good solutions while being smaller.

Experimental results are shown in Figure 10. As expected the coarser instances are easier to solve, with a factor on the median running time of up to 100 for the instance size  $(t_{\max}, n, r_{\max}) = (192, 5, 80)$  when having blocks of 16 time steps. However merging the time steps increases the median optimal objective value by more than 50% for the instance size  $(t_{\max}, n, r_{\max}) = (192, 5, 20)$  and still by more than 22% for the instance size  $(t_{\max}, n, r_{\max}) = (192, 5, 80)$ . While the reduced running time for the coarser instances looks promising, the optimal objective value of these instances is significantly worse.

#### IX. CONCLUSION

We considered a fleet scheduling problem that was not addressed in the scientific literature so far. After the formal

definition we proved the problem to be NP-hard. We formulated the problem as a Mixed Integer Linear Program (MILP), taking care in the details to achieve a strong linear programming relaxation. Moreover, we suggested an additional class of problem-specific strengthening inequalities, although it turned out that in practice they do only have a minor impact when using a modern MILP solver.

As an alternative to directly solving this MILP, we proposed a Benders decomposition for the problem. The basic variant of this Benders decomposition was enhanced by multiple measures. For one of these measures the variables relating to the energy consumption at each time step were moved from the subproblem (SP) to the master problem (MP). This allowed to also move related inequalities from the SP to the MP making the linear programming relaxation of the MP stronger. Moreover, we proposed a heuristic based on General Variable Neighborhood Search (GVNS). The heuristic is used to solve the MP in the first iterations only approximately but significantly faster. By switching to an exact method for the MP afterwards the approach ultimately still yields an optimal solution, given enough time and memory. Furthermore, we investigated the effect on the performance when using Branch-and-Check, which solves the MP only once and adds Benders cuts as lazy constraints, instead of the classic iterative approach.

We experimentally compared the Benders decomposition to solving the complete MILP on a set of artificial benchmark instances and evaluated the effect of the different performance improvement measures of the Benders decomposition. It turned out that directly solving the MILP performs better than the Benders decomposition on small to medium sized instances. On large instances, however, the solving of the original MILP fails to find reasonable solutions and meaningful dual bounds. The Benders decomposition based on the extended MP is here able to give better dual bounds and the Benders decomposition with the GVNS finds better primal solutions. Using Branch-and-Check and adding our strengthening inequalities did not improve the performance of the Benders decomposition. Note that the GVNS heuristic does not work for the extended MP and therefore it is not directly possible to combine the Benders decomposition based on the extended MP with the GVNS.

This aspect may be addressed in future work since such an approach might yield both, good solutions and good dual bounds at the same time. However, appropriately extending the GVNS is not straight-forward. One possibility to solve the issue would be a nested Benders decomposition, i.e., solving the MP of the Benders decomposition with extended MP by another Benders decomposition. While the outer Benders decomposition is able to give good bounds, it is still possible to use our heuristic to find good solutions to the MP of the inner Benders decomposition.

Future work may also investigate a more fine-grained approach for the amount of effort that is used to solve the MP in the different phases of the whole optimization. It seems reasonable to start with little effort and raising it whenever

$$\begin{aligned}
 (\text{SP}(\bar{x})) \quad & \min \quad \sum_{t \in T} \tilde{c}_t E_t^{\text{grid}} - \alpha \sum_{t \in T} E_t^{\text{sur}} & (59) \\
 \text{s.t.} \quad & p_{v,t} = 0 & (\lambda_{v,t}^{p=0}) \quad v \in V, t \in T \setminus T_v^{\text{home}} & (60) \\
 & E_{v,0} + \Delta t \sum_{k=1}^t p_{v,k} - E_{v,t-1}^{\text{res,end}} \leq E^{\text{cap}} & (\lambda_{v,t}^{\text{ub}}) \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\text{max}}\} & (61) \\
 & E_{v,0} + \Delta t \sum_{k=1}^{t-1} p_{v,k} - E_{v,t}^{\text{res,start}} \geq 0 & (\lambda_{v,t}^{\text{lb}}) \quad v \in V, t \in T^{\text{starts}} & (62) \\
 & E_t^{\text{grid}} + E_t^{\text{sur}} = \sum_{v \in V} \Delta t p_{v,t} & (\lambda_t^{\text{E}}) \quad t \in T & (63) \\
 & E_t^{\text{grid}} \geq 0 & & t \in T & (64) \\
 & 0 \leq E_t^{\text{sur}} \leq E_t^{\text{surmax}} & (\lambda_t^{\text{sur}}) \quad t \in T & (65) \\
 & 0 \leq p_{v,t} \leq P^{\text{max}} & (\lambda_{v,t}^{\text{p}}) \quad v \in V, t \in T & (66)
 \end{aligned}$$

FIGURE 11. Primal subproblem.

the derived cut does not cut off or worsen the found MP solution. Finally, the algorithm will again solve the MP with the MILP solver to optimality to guarantee exactness of the whole approach. When using the MILP solver for the MP, one may adapt the allowed gap from higher to lower values. For the heuristic the allowed runtime may be adapted.

Last but not least, it would also be interesting to investigate generalizations and extensions of the problem in future work. Practically relevant aspects are to consider uncertainties, in particular in the energy price or the available surplus energy, or to consider non-linear maximal charging rates. Further possible extensions are the consideration of also delivering energy from the vehicles back to the grid, or to allow different vehicle types. Vehicles could, e.g., have different battery capacities or even specific characteristics that may be required by some of the reservations. In the original MILP model vehicle-specific battery capacities can be directly considered in a straight-forward way. Different further vehicle-specific attributes, such as passenger capacities, can also be given for the vehicles and respective constraints can be added. While adapting the models in these ways is relatively easy, actual running times for solving them may differ significantly, either in the negative or positive sense.

## APPENDIX A DERIVATION OF THE BENDERS CUTS

This part of the appendix deals with the derivation of the Benders cuts. First feasibility and optimality cuts for the basic Benders decomposition are derived. Afterwards we derive feasibility cuts for the Benders decomposition with the extended master problem.

### A. BASIC BENDERS DECOMPOSITION

Benders cuts will be derived by solving the subproblem for specific reservation assignments  $\bar{x} = (\bar{x}_{r,v})_{r \in R, v \in V}$  in a current solution to the MP given by (24) to (32).

The primal subproblem  $\text{SP}(\bar{x})$  from Section VI is shown again in Figure 11. To get feasibility and optimality cuts, we have to derive the dual of this subproblem. In the primal problem  $\text{SP}(\bar{x})$ , the  $\lambda$ -variables in parentheses denote the dual variables that correspond to each constraint. The resulting dual subproblem is shown in Figure 12.

This dual subproblem can be simplified in multiple regards as follows.

- Variable  $\lambda_{v,t}^{p=0}$  appears only once in (71) and is unbounded. Therefore, it can always be chosen in a way that the inequality is fulfilled. Thus,  $\lambda_{v,t}^{p=0}$  as well as (71) can be removed.
- Variables  $\lambda_t^{\text{sur}}$  and  $\lambda_{v,t}^{\text{p}}$  appear in the objective function with positive coefficients and in two constraints that only give upper limits. The value of such a variable will always be the smaller upper limit in an optimal solution.

This leads to the equivalent formulation given in Figure 13.

For reasons of clarity we define for a given solution to  $\text{DSP}(\bar{x})$ , denoted by  $\bar{\lambda}_t^{\text{E}}$ ,  $\bar{\lambda}_{v,t}^{\text{lb}}$  and  $\bar{\lambda}_{v,t}^{\text{ub}}$ ,

$$C_{v,t} := \min \left( 0, \bar{\lambda}_t^{\text{E}} - \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \bar{\lambda}_{v,k}^{\text{ub}} - \sum_{k \in T^{\text{starts}} | k \geq t+1} \bar{\lambda}_{v,k}^{\text{lb}} \right). \quad (78)$$

### 1) FEASIBILITY CUTS

Feasibility cuts are inferred from extreme rays of the dual subproblem in case this problem is unbounded. Let  $\bar{\lambda}_{v,t}^{\text{lb}}$ ,  $\bar{\lambda}_{v,t}^{\text{ub}}$ , and  $\bar{\lambda}_t^{\text{E}}$  represent such an extreme ray. We then have to have  $\bar{\lambda}_t^{\text{E}} \leq 0$  and the objective function without constants (i.e., without  $\alpha$ ) has to be positive. The third term of the objective function is zero in this case. Furthermore,  $\bar{\lambda}_t^{\text{E}} = 0$  due to a similar argument as before. The formulation can now be split into different parts for the different vehicles, which are independent from each other. As the total objective value is positive, some of these parts also have to provide positive contributions and therefore also form rays.

$$\begin{aligned} \max \quad & \sum_{v \in V} \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + E_{v,t-1}^{\text{res, end}} - E_{v,0} \right) \lambda_{v,t}^{\text{ub}} + \sum_{v \in V} \sum_{t \in T^{\text{starts}}} \left( E_{v,t}^{\text{res, start}} - E_{v,0} \right) \lambda_{v,t}^{\text{lb}} + \\ & \sum_{t \in T} E_t^{\text{surmax}} \lambda_t^{\text{sur}} + \sum_{v \in V} \sum_{t \in T} P^{\text{max}} \lambda_{v,t}^{\text{p}} \end{aligned} \quad (67)$$

$$\text{s.t.} \quad \lambda_t^{\text{E}} \leq \tilde{c}_t \quad (E_t^{\text{grid}}) \quad t \in T \quad (68)$$

$$\lambda_t^{\text{E}} + \lambda_t^{\text{sur}} \leq -\alpha \quad (E_t^{\text{sur}}) \quad t \in T \quad (69)$$

$$\lambda_{v,t}^{\text{p}} - \Delta t \lambda_t^{\text{E}} + \Delta t \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \lambda_{v,k}^{\text{ub}} + \Delta t \sum_{k \in T^{\text{starts}} | k \geq t+1} \lambda_{v,k}^{\text{lb}} \leq 0 \quad (p_{v,t}) \quad v \in V, t \in T_v^{\text{home}} \quad (70)$$

$$\lambda_{v,t}^{\text{p}=0} + \lambda_{v,t}^{\text{p}} - \Delta t \lambda_t^{\text{E}} + \Delta t \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \lambda_{v,k}^{\text{ub}} + \Delta t \sum_{k \in T^{\text{starts}} | k \geq t+1} \lambda_{v,k}^{\text{lb}} \leq 0 \quad (p_{v,t}) \quad v \in V, t \in T \setminus T_v^{\text{home}} \quad (71)$$

$$\lambda_{v,t}^{\text{p}=0} \in \mathbb{R} \quad v \in V, t \in T \quad (72)$$

$$\lambda_t^{\text{E}} \in \mathbb{R} \quad t \in T \quad (73)$$

$$\lambda_t^{\text{sur}} \leq 0 \quad t \in T \quad (74)$$

$$\lambda_{v,t}^{\text{p}} \leq 0 \quad v \in V, t \in T \quad (75)$$

$$\lambda_{v,t}^{\text{ub}} \leq 0 \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\text{max}}\} \quad (76)$$

$$\lambda_{v,t}^{\text{lb}} \geq 0 \quad v \in V, t \in T^{\text{starts}} \quad (77)$$

FIGURE 12. Dual subproblem.

$$\begin{aligned} (\text{DSP}(\bar{x})) \quad \max \quad & \sum_{v \in V} \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + E_{v,t-1}^{\text{res, end}} - E_{v,0} \right) \lambda_{v,t}^{\text{ub}} + \sum_{v \in V} \sum_{t \in T^{\text{starts}}} \left( E_{v,t}^{\text{res, start}} - E_{v,0} \right) \lambda_{v,t}^{\text{lb}} + \\ & \sum_{t \in T} E_t^{\text{surmax}} \min(0, -\alpha - \lambda_t^{\text{E}}) + \\ & \sum_{v \in V} \sum_{t \in T_v^{\text{home}}} \Delta t P^{\text{max}} \cdot \min \left( 0, \lambda_t^{\text{E}} - \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \lambda_{v,k}^{\text{ub}} - \sum_{k \in T^{\text{starts}} | k \geq t+1} \lambda_{v,k}^{\text{lb}} \right) \end{aligned} \quad (80)$$

$$\text{s.t.} \quad \lambda_t^{\text{E}} \leq \tilde{c}_t \quad t \in T \quad (81)$$

$$\lambda_{v,t}^{\text{ub}} \leq 0 \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\text{max}}\} \quad (82)$$

$$\lambda_{v,t}^{\text{lb}} \geq 0 \quad v \in V, t \in T^{\text{starts}} \quad (83)$$

FIGURE 13. Simplified dual subproblem.

Hence the problem of finding extreme rays can be decomposed into smaller problems, one for each vehicle. Using the obtained rays, we get the following feasibility cuts for vehicles  $v \in V$ .

$$\begin{aligned} & \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + E_{v,t-1}^{\text{res, end}} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{ub}} \\ & + \sum_{t \in T^{\text{starts}}} \left( E_{v,t}^{\text{res, start}} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{lb}} \\ & + \sum_{t \in T_v^{\text{home}}} \Delta t P^{\text{max}} C_{v,t} \\ = & \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + \sum_{r \in R | t_r^{\text{end}} \leq t-1} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{ub}} \\ & + \sum_{t \in T^{\text{starts}}} \left( \sum_{r \in R | t_r^{\text{start}} \leq t} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{lb}} \end{aligned}$$

$$+ \sum_{t \in T_v^{\text{avail}}} \Delta t P^{\text{max}} C_{v,t} \left( 1 - \sum_{r \in R | t \in T_r^{\text{res}}} x_{r,v} \right) \leq 0 \quad (79)$$

The factors  $\gamma_{c,r}$  and the constant  $\gamma'_c$  can be obtained from these cuts by comparison of the coefficients of  $x$ .

## 2) OPTIMALITY CUTS

Optimality cuts are inferred from extreme points of the dual subproblem. Let  $\bar{\lambda}_{v,t}^{\Delta E}$  and  $\bar{\lambda}_t^{\text{E}}$  be such an extreme point. Unlike for extreme rays, we cannot discard constants so the above simplifications do not work here. The resulting cuts are

$$\begin{aligned} & \sum_{v \in V} \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + \sum_{r \in R | t_r^{\text{end}} \leq t-1} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{ub}} \\ & + \sum_{v \in V} \sum_{t \in T^{\text{starts}}} \left( \sum_{r \in R | t_r^{\text{start}} \leq t} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{lb}} \end{aligned}$$

$$\begin{aligned}
 & (\text{DSP}(\bar{x}, \bar{E}^{\text{grid}}, \bar{E}^{\text{sur}})) \\
 \max \quad & \sum_{v \in V} \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} (E^{\text{cap}} + E_{v,t-1}^{\text{res, end}} - E_{v,0}) \lambda_{v,t}^{\text{ub}} + \sum_{v \in V} \sum_{t \in T^{\text{starts}}} (E_{v,t}^{\text{res, start}} - E_{v,0}) \lambda_{v,t}^{\text{lb}} + \\
 & \sum_{t \in T} (\bar{E}_t^{\text{grid}} + \bar{E}_t^{\text{sur}}) \lambda_t^{\text{E}} + \\
 & \sum_{v \in V} \sum_{t \in T_v^{\text{home}}} \Delta t P^{\text{max}} \cdot \min \left( 0, -\lambda_t^{\text{E}} - \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \lambda_{v,k}^{\text{ub}} - \sum_{k \in T^{\text{starts}} | k \geq t+1} \lambda_{v,k}^{\text{lb}} \right) \quad (85) \\
 \text{s.t.} \quad & \lambda_t^{\text{E}} \in \mathbb{R} \quad t \in T \quad (86) \\
 & \lambda_{v,t}^{\text{ub}} \leq 0 \quad v \in V, t \in T^{\text{ends}} \cup \{t_{\text{max}}\} \quad (87) \\
 & \lambda_{v,t}^{\text{lb}} \geq 0 \quad v \in V, t \in T^{\text{starts}} \quad (88)
 \end{aligned}$$

FIGURE 14. Dual subproblem after extending the MP.

$$\begin{aligned}
 & \sum_{v \in V} \sum_{t \in T^{\text{ends}} \cup \{t_{\text{max}}\}} \left( E^{\text{cap}} + \sum_{r \in R | t_r^{\text{end}} \leq t-1} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{ub}} + \sum_{v \in V} \sum_{t \in T^{\text{starts}}} \left( \sum_{r \in R | t_r^{\text{start}} \leq t} E_r^{\text{res}} x_{r,v} - E_{v,0} \right) \bar{\lambda}_{v,t}^{\text{lb}} + \\
 & \sum_{t \in T} (E_t^{\text{grid}} + E_t^{\text{sur}}) \lambda_t^{\text{E}} + \\
 & \sum_{v \in V} \sum_{t \in T_v^{\text{avail}}} \Delta t P^{\text{max}} \left( 1 - \sum_{r \in R | t \in T_r^{\text{res}}} x_{r,v} \right) \cdot \min \left( 0, -\bar{\lambda}_t^{\text{E}} - \sum_{k \in T^{\text{ends}} \cup \{t_{\text{max}}\} | k \geq t} \bar{\lambda}_{v,k}^{\text{ub}} - \sum_{k \in T^{\text{starts}} | k \geq t+1} \bar{\lambda}_{v,k}^{\text{lb}} \right) \leq 0 \quad (89)
 \end{aligned}$$

FIGURE 15. Feasibility cuts after extending the MP.

$$\begin{aligned}
 & + \sum_{t \in T} E_t^{\text{surmax}} \cdot \min \left( 0, -\alpha - \bar{\lambda}_t^{\text{E}} \right) \\
 & + \sum_{v \in V} \sum_{t \in T_v^{\text{avail}}} \Delta t P^{\text{max}} C_{v,t} \left( 1 - \sum_{r \in R | t \in T_r^{\text{res}}} x_{r,v} \right) \leq \mu. \quad (84)
 \end{aligned}$$

Again,  $\beta_{c,r,v}$  and  $\beta'_c$  can be derived by comparison of coefficients.

### B. EXTENSION OF THE MP

Starting point for the Benders cuts is now the MP given by (44) to (56). The dual subproblem changes in some regards compared to the dual subproblem of the basic Benders decomposition.

- The term with  $\lambda_t^{\text{sur}}$  in the objective drops because  $E_t^{\text{sur}}$  is no longer part of the subproblem.
- $E_t^{\text{sur}}$  and  $E_t^{\text{grid}}$  turn into constants, leading to a new term in the objective function.
- The constraints corresponding to  $E_t^{\text{sur}}$  and  $E_t^{\text{grid}}$  are dropped.

After simplification the dual of the updated subproblem can be stated as shown in Figure 14.

Note that there are no optimality cuts in this case as the variables from the SP do not occur in the objective function of the MP.

### 1) FEASIBILITY CUTS

Let  $\bar{\lambda}_{v,t}^{\text{p}}$ ,  $\bar{\lambda}_{v,t}^{\text{ub}}$  and  $\bar{\lambda}_{v,t}^{\text{lb}}$  be a solution to the above dual subproblem that leads to a positive objective value. The feasibility

cuts can then be formulated as shown in Figure 15. Once again,  $\gamma_{c,r,v}^{\text{ext}}$ ,  $\gamma_{c,t}^{\text{grid}}$ ,  $\gamma_{c,t}^{\text{sur}}$  and  $\gamma_c^{\text{ext}}$  can be obtained by comparison of coefficients.

### REFERENCES

- [1] J. Webb, "The future of transport: Literature review and overview," *Econ. Anal. Policy*, vol. 61, pp. 1–6, Mar. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0313592618300638>
- [2] D. Yang, H. Latchman, D. Tingling, and A. A. Amarsingh, "Design and return on investment analysis of residential solar photovoltaic systems," *IEEE Potentials*, vol. 34, no. 4, pp. 11–17, Jul. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7159231/>
- [3] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math.*, vol. 4, no. 1, pp. 238–252, Dec. 1962.
- [4] J. Varga, "Computational methods for fleet scheduling in e-mobility," M.S. thesis, Dept. Inform., TU Wien, Vienna, Austria, 2021.
- [5] O. Sassi and A. Oulamara, "Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches," *Int. J. Prod. Res.*, vol. 55, no. 2, pp. 519–535, Jan. 2017, doi: [10.1080/00207543.2016.1192695](https://doi.org/10.1080/00207543.2016.1192695).
- [6] S. Bunte and N. Klierer, "An overview on vehicle scheduling models," *Public Transp.*, vol. 1, no. 4, pp. 299–317, 2009, doi: [10.1007/s12469-010-0018-5](https://doi.org/10.1007/s12469-010-0018-5).
- [7] J. L. Saha, "An algorithm for bus scheduling problems," *J. Oper. Res. Soc.*, vol. 21, no. 4, pp. 463–474, Dec. 1970, doi: [10.1057/jors.1970.95](https://doi.org/10.1057/jors.1970.95).
- [8] A. A. Bertossi, P. Carraresi, and G. Gallo, "On some matching problems arising in vehicle scheduling models," *Networks*, vol. 17, no. 3, pp. 271–281, 1987, doi: [10.1002/net.3230170303](https://doi.org/10.1002/net.3230170303).
- [9] B. Messaoudi and A. Oulamara, *Electric Bus Scheduling and Optimal Charging* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2019, vol. 11756, ch. 15, pp. 233–247, doi: [10.1007/978-3-030-31140-7\\_15](https://doi.org/10.1007/978-3-030-31140-7_15).
- [10] M. Rinaldi, E. Picarelli, A. D’Ariano, and F. Viti, "Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications," *Omega*, vol. 96, Oct. 2020, Art. no. 102070. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0305048318314075>

- [11] M. E. van Kooten Niekerk, J. M. van den Akker, and J. A. Hoogeveen, "Scheduling electric vehicles," *Public Transp.*, vol. 9, nos. 1–2, pp. 155–176, 2017, doi: [10.1007/s12469-017-0164-0](https://doi.org/10.1007/s12469-017-0164-0).
- [12] M. Wen, E. Linde, S. Ropke, P. Mirchandani, and A. Larsen, "An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem," *Comput. Oper. Res.*, vol. 76, pp. 73–83, Dec. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0305054816301460>
- [13] J. García-Álvarez, M. A. González, and C. R. Vela, "Metaheuristics for solving a real-world electric vehicle charging scheduling problem," *Appl. Soft Comput.*, vol. 65, pp. 292–306, Apr. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1568494618300164>
- [14] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1500–1517, 2nd Quart., 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7383228/>
- [15] N. I. Nimalisiri, C. P. Mediwiththe, E. L. Ratnam, M. Shaw, D. B. Smith, and S. K. Halgamuge, "A survey of algorithms for distributed charging control of electric vehicles in smart grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4497–4515, Nov. 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8855113/>
- [16] R. Fachrizal, M. Shepero, D. van der Meer, J. Munkhammar, and J. Widén, "Smart charging of electric vehicles considering photovoltaic power production and electricity consumption: A review," *eTransportation*, vol. 4, May 2020, Art. no. 100056. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590116820300138>
- [17] N. Korolko and Z. Sahinoglu, "Robust optimization of EV charging schedules in unregulated electricity markets," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 149–157, Jan. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7244227/>
- [18] B. Schaden, T. Jatschka, S. Limmer, and G. R. Raidl, "Smart charging of electric vehicles considering SOC-dependent maximum charging powers," *Energies*, vol. 14, no. 22, p. 7755, Nov. 2021. [Online]. Available: <https://www.mdpi.com/1996-1073/14/22/7755>
- [19] J. Han, J. Park, and K. Lee, "Optimal scheduling for electric vehicle charging under variable maximum charging power," *Energies*, vol. 10, no. 7, p. 933, 2017. [Online]. Available: <https://www.mdpi.com/1996-1073/10/7/933>
- [20] C. Z. El-Bayeh, I. Mougharbel, M. Saad, A. Chandra, D. Asber, and S. Lefebvre, "Impact of considering variable battery power profile of electric vehicles on the distribution network," in *Proc. 4th Int. Conf. Renew. Energies Developing Countries (REDEC)*, Nov. 2018, pp. 1–8.
- [21] J. Betz, D. Werner, and M. Lienkamp, "Fleet disposition modeling to maximize utilization of battery electric vehicles in companies with on-site energy generation," *Transp. Res. Proc.*, vol. 19, pp. 241–257, Jan. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2352146516308705>
- [22] R. Karp, "Reducibility among combinatorial problems," in *Complexity Computer Computations*. Boston, MA, USA: Springer, 1972, pp. 85–103.
- [23] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The Benders decomposition algorithm: A literature review," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 801–817, 2017. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0377221716310244>
- [24] J. C. Beck, *Checking-Up Branch-and-Check* (Lecture Notes in Computer Science), vol. 6308. Berlin, Germany: Springer, 2010, ch. 10, pp. 84–98, doi: [10.1007/978-3-642-15396-9\\_10](https://doi.org/10.1007/978-3-642-15396-9_10).
- [25] G. R. Raidl, T. Baumhauer, and B. Hu, "Speeding up logic-based benders' decomposition by a metaheuristic for a bi-level capacitated vehicle routing problem," in *Hybrid Metaheuristics*, M. J. Blesa, C. Blum, and S. Voß, Eds. Cham, Switzerland: Springer, Jun. 2014, pp. 183–197.
- [26] M.-C. Lai, H.-S. Sohn, T.-L. Tseng, and C. Chiang, "A hybrid algorithm for capacitated plant location problem," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8599–8605, Dec. 2010. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0957417410006081>
- [27] P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi, "Variable neighborhood search: Basics and variants," *EURO J. Comput. Optim.*, vol. 5, no. 3, pp. 423–454, Sep. 2017. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2192440621000873>
- [28] Gurobi Optimization, LLC. (2021). *Gurobi Optimizer Reference Manual*. [Online]. Available: <https://www.gurobi.com>
- [29] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, 2017, doi: [10.1137/141000671](https://doi.org/10.1137/141000671).
- [30] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Rev.*, vol. 59, no. 2, pp. 295–320, 2017, doi: [10.1137/15M1020575](https://doi.org/10.1137/15M1020575).



**JOHANNES VARGA** received the B.S. degree in technical physics, the B.S. degree in computer engineering, and the M.S. degree in logic and computation from TU Wien, Vienna, Austria, in 2019, 2020, and 2021, respectively. He is currently pursuing the Ph.D. degree in computer science.

Since 2021, he has been a University Assistant at TU Wien. His research interests include heuristic and exact methods for combinatorial optimization problems with a focus on scheduling problems.



**GÜNTHER R. RAIDL** received the M.S. and Ph.D. degrees in computer science from TU Wien, Vienna, Austria, in 1992 and 1994, respectively, and the Habilitation degree in practical computer science, in 2003.

In 2005, he received a University Professor position for combinatorial optimization at TU Wien. Since 2016, he has been also a Faculty Member of the Vienna Graduate School on Computational Optimization. He has coauthored a text book on

hybrid metaheuristics and over 185 reviewed papers in scientific journals, books, and conference proceedings. Moreover, he has co-edited 13 books and coauthored one book on hybrid metaheuristics. His research interests include algorithms and data structures in general and combinatorial optimization in particular, with a specific focus on metaheuristics, mathematical programming, intelligent search methods, and hybrid optimization approaches. His research work typically combines theory and practice for application areas such as scheduling, network design, transport optimization, logistics, and cutting and packing.

Dr. Raidl is a Co-Founder and a Steering Committee Member of the Annual European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP). He is an Associate Editor of the *INFORMS Journal on Computing* and the *International Journal of Metaheuristics* and at the Editorial Board of several journals, including *Algorithms*, *Metaheuristics*, and the *Evolutionary Computation*.



**STEFFEN LIMMER** received the M.S. (Diploma) degree in computer science from the University of Jena, Germany, in 2009, and the Ph.D. degree in engineering from the University of Erlangen–Nürnberg, Germany, in 2016. From 2009 to 2016, he worked as a Scientific Assistant at the Chair of Computer Architecture. Since December 2016, he has been a Senior Scientist at the Honda Research Institute Europe. His current research interests include optimization and data-driven modeling in the context of energy management systems.