



HAL
open science

Optimal Charging Scheduling of Electric Vehicles: the Co-Charging Case

Paul Malisani, Jiamin Zhu, Philippe Pognant-Gros

► **To cite this version:**

Paul Malisani, Jiamin Zhu, Philippe Pognant-Gros. Optimal Charging Scheduling of Electric Vehicles: the Co-Charging Case. *IEEE Transactions on Power Systems*, 2023, 38 (2), pp.1069-1080. 10.1109/TPWRS.2022.3172286 . hal-04115000

HAL Id: hal-04115000

<https://ifp.hal.science/hal-04115000>

Submitted on 2 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal charging scheduling of electric vehicles: the co-charging case

P. Malisani*, J. Zhu, P. Pognant-Gros

Abstract—A *co-charging* parking system is a charging system in which the chargers are automated, i.e., they can automatically connect to or disconnect from vehicles presented in the parking. This system makes it possible to have less chargers than the number of parking places. This paper studies an optimal scheduling problem aiming at charging all vehicles to required battery level while minimizing the total charging cost. We formulate this scheduling problem as a non-linear optimal control problem with mixed integer constraints and mixed discrete and continuous time. The problem is solved using a Branch and Bound Strategy coupled with Pontryagin Maximum Principle (PMP) and Interior Point Methods (IPMs) for optimal control. The presented method is tested on numerous configurations and shows promising results both in term of precision and in term of convergence rate.

Index Terms—Co-charging stations, electric vehicle, battery charging, optimal scheduling, Branch and Bound methods, mixed integer optimal control

I. INTRODUCTION

The increasing use of electric vehicles (EVs) is a major component of improving air quality in cities. To support this development actual car parks must be renewed with charging equipment. In many cases, this renewal consists in equipping a certain number of parking places with chargers. This solution can be expensive due to the large number of chargers to be deployed and lacks of flexibility in separating parking spaces for electric and thermal vehicles. In this context, so-called *co-charging* solutions can help to alleviate these drawbacks. In a co-charging system, chargers can move throughout the parking and plug-in and recharge chosen electric vehicles. This autonomous moving chargers allows to recharge a number of vehicles greater than the number of chargers (typically one charger for 3 to 5 parking places) by means of planning the sequence of plug-in/charging/unplug cycles of each charger. In the case we are interested in, the autonomous chargers can switch from a vehicle to another only at a regular time steps. In order to operate a co-charging system one has to schedule the charging of the EVs in order to ensure that they leave the parking lot with required state of charge while minimizing a user defined objective function, typically the

charging cost. At present, this charging technology is still in the early planning stage and has not yet been tested. However, some manufacturers are starting to take an interest in it which motivates this contribution [1].

Optimal charging of EVs has been extensively studied for years. On the one hand, some of this literature focus on single EV optimal charging using different methods ranging from genetic algorithm to deep-learning-based methods [2]–[4]. On the other hand, another part of the literature focus on EV fleet charging scheduling [5], [6]. The problem we are interested in lies in between both cases and concerns the optimal charging of a few EVs, i.e. more than one vehicle but not enough to consider the statistical properties of the EVs (not enough to make a fleet). Indeed, the charging scheduling of a small group of EVs has also been addressed in recent works. In [7], the authors studied a stochastic problem of optimal charging of plug-in electric vehicles at a parking-lot. In [8], [9] the optimal charging problem of electric vehicles that are able to exchange energy between them and with the electrical grid (V2G) has been addressed. However, these studies did not consider the case where the number of vehicles to be charged is larger than the number of chargers whereas it is a key feature regarding the co-charging case. In [10] the problem of the scheduling of several chargers distributed over a an electrical grid is addressed and the scheduling is handled using a heuristic prioritizing the vehicles with smaller departure times. This charging method does not allow to take advantage of a time varying price of electricity or does not allow the charging strategy to reduce energy losses when possible. In [11], the authors address the optimal assignment of tours to be processed of a mixed fleet of combustion and electric vehicles together with the optimal charging scheduling for the latter. The battery model used in the study is a discrete integrator model, which does not take into account the non-linearities in the battery dynamics. But, optimal control strategy of linear systems consists in bang-bang control [12], that is to say always charging EVs at maximal power. This bang-bang strategy does not allow to take advantage of the higher efficiency of a low power charging and hence yields an over consumption of energy.

The dependence of the charging efficiency on the state-of-charge and on the charging power makes the total amount of energy required to charge a vehicle depends on the charging trajectory, i.e. the optimization variables. Indeed, the faster the vehicle is recharged, the greater the energy losses. Therefore, the problem we are interested in is not a combinatorial problem such as the bin-packing problem but must be considered as a mixed integer nonlinear optimal control problem. Several

contributions in the literature focus on mixed integer optimal control problems referred as *hybrid optimal control* [13], [14] or *mixed integer optimal control problems* [15], [16]. In [13], [14], the authors considered a pure continuous time problem, in which switching times of the integer control variable are free and must be computed. In [15], [16], approximations methods such as sum-up rounding strategies are presented and successfully applied. These methods consist of a time-discretization of the mixed integer optimal control problem together with a continuous relaxation of the integer constraints followed by *rounding* them into integer solutions. These methods converge to the optimal solution as the switching time steps converge to 0. Unfortunately, in the co-charging case, these switching times are fixed and are much larger than the time scale of the battery dynamics. In this situation, there is no guarantee that a rounding strategy converges to an optimal solution or even a feasible one. Consequently, the co-charging problem is a Non-Linear Optimal Control Problem with Mixed Integer constraints and Mixed discrete and continuous Time (NLOCPMIMT). The integer part of the problem stems from the fact that only some of the vehicles can be charged at the same time (since there are more parking slots than chargers). The continuous time optimal control part stems from the fact that the charging of electrical batteries can be modeled as a non-linear dynamical system. Finally, the discrete time part stems from the fact that the chargers can move from a vehicle to another at certain time steps. To the knowledge of the authors, this problem has not been addressed in the literature.

In this paper, we tackle the NLOCPMIMT in its full complexity. More precisely, we consider a non-linear battery model calibrated on experimental data, arbitrary numbers of vehicles and chargers. Our main contribution is an algorithm that converges to the global optimum of the NLOCPMIMT. Our approach is as follows, we rely on a Branch and Bound approach, since the large switching time steps ensure a reasonable depth of the associated search-tree. This Branch and Bound method is combined with a continuous time interior point method for optimal control allowing one to compute the optimal solution of the Branch and Bound lower bounding procedure with high precision and small execution time. In addition, the presented methodology is still quite computationally expensive and additional work might be necessary for an online implementation. Nevertheless, the global convergence of the presented algorithm makes this methodology a meaningful benchmark against which other scheduling strategy can be compared. Such a benchmark is particularly useful to assess the performances of heuristics based methods.

This paper is organized as follows, in Section II the nomenclature is presented. In Section III, the optimal scheduling of vehicle charging problem is presented. In detail, in Section III-A the non-linear battery dynamics is presented, in Section III-B the NLOCPMIMT is presented. In Section IV, the Branch and Bound algorithm used to handle the integer variables is described. In Section V, the lower bounding part of the Branch and Bound algorithm is detailed. In Section VI the convergence of the Branch and Bound algorithm is investigated. A numerical study of an optimal vehicle charging scheduling is given in Section VII. Finally, conclusions and

perspectives are discussed in Section IX.

II. NOMENCLATURE

- n_x : number of vehicles to be charged
- n_c : maximum number of chargers
- $P_c = \{p_1, \dots, p_{n_{cs}}\}$: set of maximal charging powers
- n_{cs} : number of groups of same maximum power chargers
- G_j : Group of chargers of same maximum power, $j = 1, \dots, n_{cs}$
- $P_{G_j}^+$: Maximum power of group charger $j = 1, \dots, n_{cs}$
- $\mathbf{m} \in \{0, 1\}^{n_x \times n_{cs}}$: Vehicle-charger connection matrix. Vehicle i is connected to group charger j if and only if the element $m_{i,j}$ of \mathbf{m} is 1.
- $P_s \in \mathbb{R}$: maximal power available for the whole charging station
- $t_{0,k}$: earliest plugging time of the vehicle k
- $t_{f,k}$: latest unplugging time of the vehicle k
- t_f : optimization horizon
- ΔT : minimum duration of a vehicle connection to a charger
- $k_f = \lfloor t_f / \Delta T \rfloor$: maximum number of switching times
- $x_{0,k}$: initial battery energy of vehicle k
- $x_{f,k}$: desired battery energy of vehicle k
- $p_{c,k} \in C_2(\mathbb{R}, \mathbb{R})$: charging power of vehicle k
- $\lambda_k \in L^\infty([0, t_f], [0, 1])$: charging rate of vehicle k
- $x_k \in C_0([0, t_f], \mathbb{R})$: battery state of energy of vehicle k
- $p_{c,k}^+ \in C_2(\mathbb{R}, \mathbb{R})$: maximal charging power of vehicle k
- $f_{c,k} : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$: dynamics of vehicle battery k
- $\theta_x \in \mathbb{R}^+$: weight of state of charge penalization
- $e_{opt}(n_x) = \int \sum_{k=1}^{n_x} \lambda_k p_{c,k}^+(x^{\lambda_k}) dt$: total energy consumption with optimal charging rates λ .
- $c_{opt}(n_x) = \int \text{price} \sum_{k=1}^{n_x} \lambda_k p_{c,k}^+(x^{\lambda_k}) dt$: total charging cost with optimal charging rates λ .
- $e_{ref}(n_x) = \int \sum_{k=1}^{n_x} p_{c,k} dt$: Sequence of total energy consumption with reference charging powers p_c .
- $c_{ref}(n_x) = \int \text{price} \sum_{k=1}^{n_x} p_{c,k} dt$: total charging cost with reference charging powers p_c .
- \top, \perp : True and False boolean variables respectively
- \neg : logical not
- RMES : Relative Mean Energy Savings
- ERSD : Energy Relative Standard Deviation
- RMCS : Relative Mean Cost Savings
- CRSD : Cost Relative Standard Deviation

In the rest of the paper, bold lowercase variables such as \mathbf{m} indicate matrices of $\mathbb{R}^{n_x \times n_{cs}}$. Bold capital variables such as \mathbf{M}^k indicates a sequence of such matrices also noted

$$\mathbf{M}^k = \langle \mathbf{m}^0, \dots, \mathbf{m}^k \rangle, \mathbf{m}^i \in \mathbb{R}^{n_x \times n_{cs}}, i = 0, \dots, k \quad (1)$$

And we note the concatenation between a sequence \mathbf{M}^k and a matrix $\mathbf{p} \in \mathbb{R}^{n_x \times n_{cs}}$ as follows

$$\mathbf{M}^k : \mathbf{p} = \langle \mathbf{m}^0, \dots, \mathbf{m}^k, \mathbf{p} \rangle \quad (2)$$

The concatenation of two sequences $\mathbf{M}^k = \langle \mathbf{m}^0, \dots, \mathbf{m}^k \rangle$ and $\mathbf{S}^l = \langle \mathbf{s}^0, \dots, \mathbf{s}^l \rangle$ writes as follows

$$\mathbf{M}^k : \mathbf{S}^l = \langle \mathbf{m}^0, \dots, \mathbf{m}^k, \mathbf{s}^0, \dots, \mathbf{s}^l \rangle \quad (3)$$

III. PROBLEM PRESENTATION

The problem we are interested in consists in charging n_x electric vehicles using n_b autonomous charging stations that can automatically move from one parked vehicle to another with a regular time-step ΔT . The question of planing the order of vehicles charging arise naturally in this context. This paper provides an optimal control based algorithm to compute such a planing. This algorithms aims at minimizing a time-of-use cost of charging seen from the charging station owner/operator viewpoint. The battery model used for the optimal control algorithm is presented in Section III-A. Given the battery model, Section III-B describes the optimal control problem corresponding to the vehicles charging planing. That is to say, the cost to minimize, the equations of the dynamical systems and the optimization constraints of the problem.

A. Battery Model

In order to represent the battery dynamics, an equivalent circuit model comprising a voltage source u_{ocv} , an electric resistance r_c is used [17] and a maximal charging power p_c^+ . These variables are dependant on the battery state of energy (SOE) x . In addition, these functions are approximated by polynomial functions of the SOE using experimental data. Using these functions and noting $p_{c,k}$ the charging power of vehicle k and $x^{p_{c,k}}$ its corresponding SOE, one has the following dynamics

$$\begin{aligned} \dot{x}_k &= \frac{u_{ocv}(x_k)}{2r_c(x_k)} \times \\ &\quad \left(-u_{ocv}(x_k) + \sqrt{u_{ocv}^2(x_k) + 4r_c(x_k)p_{c,k}} \right) \\ &= f_c(x_k, p_{c,k}) \end{aligned} \quad (4)$$

Taking into account the SOE dependant charging power limitation one has

$$p_{c,k} \leq p_{c,k}^+(x_k) \quad (5)$$

B. Optimal charging problem

In the problem we are interested in, chargers can be of different maximum charging power. Therefore, these chargers are gathered in n_{cs} groups G_j , $j = 1, \dots, n_{cs}$, of same charging maximal charging power $P_{G_j}^+$. Therefore, the total amount of chargers is equal to the sum of elements of each groups G_j :

$$\sum_j \text{card}(G_j) = n_c \quad (6)$$

The problem we are interested in consists in charging n_x vehicles using n_c chargers able to switch position to charge another vehicle every ΔT . Therefore, for all $t \in [s\Delta T, (s+1)\Delta T)$, the connections between vehicles and chargers are represented by a matrix $\mathbf{m}^s \in \{0, 1\}^{n_x \times n_{cs}}$. Each element $m_{i,j}^s$ of this matrix is equal to 1 if and only if vehicle i is connected to group charger j . Since each vehicle can be connected to at most one charger, one has

$$\sum_{j=1}^{n_{cs}} m_{i,j}^s \leq 1 \quad ; \quad \forall i = 1, \dots, n_x, \quad \forall s \in 0, \dots, k_f \quad (7)$$

In addition, the number of vehicles connected to the chargers of group j must be less or equal to the number of chargers within this group, which writes

$$\sum_{i=1}^{n_x} m_{i,j}^s \leq \text{card}(G_j) \quad ; \quad \forall j = 1, \dots, n_{cs}, \quad \forall s \in 0, \dots, k_f \quad (8)$$

Moreover, the total number of connected vehicles cannot exceed the minimum between the amount of vehicles or the amount of chargers:

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_{cs}} m_{i,j}^s \leq \min\{n_x, n_c\} \quad ; \quad \forall s = 0, \dots, k_f \quad (9)$$

Finally, the quantity to minimize is a trade-off between the co-charging operator cost and the charging speed. We assume a time-of-use charging rate to compute the charging cost and the charging speed is represented using a quadratic cost on difference between the current and the target state of charge. This trade-off cost writes

$$\begin{aligned} \min_{p_{c,i}, \mathbf{M}^{k_f}} \sum_{s=0}^{k_f} \int_{s\Delta T}^{(s+1)\Delta T} & \text{price}(t) \sum_{i=1}^{n_x} \sum_{j=1}^{n_{cs}} m_{i,j}^s p_{c,i}(t) \dots \\ & + \theta_x (x_i(t) - x_i(t_{f,i}))^2 dt \end{aligned} \quad (10)$$

with $k_f = \lceil T/\Delta T \rceil$. The weight θ_x is a tuning parameter. For zero or small values of θ_x emphasis is given to the charging cost while high values of θ_x favor a faster charging. For the sake of simplicity, the following change of variable is performed

$$p_{c,i}(t) = \lambda_i(t) p_{c,i}^+(x_i(t)) \quad (11)$$

where $\lambda_i(t) \in [0, 1]$. This change of variable allows the control variables to be all valued in the same set $[0, 1]$. Moreover, when connected to a station from group G_j , the charging power of vehicle i is bounded as follows:

$$\lambda_i(t) p_{c,i}^+(x_i(t)) \leq \min\{p_{c,i}^+(x_i(t)), P_{G_j}^+\} \quad (12)$$

Finally, the NLOCPMIMT writes

$$\begin{aligned} \min_{\lambda_i, \mathbf{M}^{k_f}} \sum_{s=0}^{k_f} \int_{s\Delta T}^{(s+1)\Delta T} & \text{price}(t) \sum_{i=1}^{n_x} \sum_{j=1}^{n_{cs}} m_{i,j}^s \times \dots \\ & \lambda_i(t) p_{c,i}^+(x_i(t)) \dots \\ & + \theta_x (x_i(t) - x_i(t_{f,i}))^2 dt \end{aligned} \quad (13)$$

under the following constraints

$$\dot{x}_i(t) = f_{c,i}(x_i(t), \lambda_i(t) p_{c,i}^+(x_i(t))) \quad (14)$$

$$x_i(t) \in [0, E_i^+] \quad (15)$$

$$\lambda_i(t) \in [0, 1] \quad (16)$$

$$\mathbf{M}^{k_f} = \langle \mathbf{m}^0, \dots, \mathbf{m}^{k_f} \rangle \quad (17)$$

$$\mathbf{m}^s \in \{0, 1\}^{n_x \times n_{cs}} \quad (18)$$

$$\sum_{j=1}^{n_{cs}} m_{i,j}^s \leq 1 \quad (19)$$

$$\sum_{i=1}^{n_x} m_{i,j}^s \leq \text{card}(G_j) \quad (20)$$

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_{cs}} m_{i,j}^s \leq \min\{n_x, n_{cs}\} \quad (21)$$

$$P_{G_j}^+ \geq \lambda_i(t) p_{c,i}^+(x_i(t)) m_{i,j}^s \quad (22)$$

$$P_s \geq \sum_{i=1}^{n_x} \lambda_i(t) \sum_{j=1}^{n_{cs}} m_{i,j}^s p_{c,i}^+(x_i(t)) \quad (23)$$

where P_s is the maximum power capacity of the charging station.

IV. A BRANCH AND BOUND APPROACH TO INTEGER OPTIMAL CONTROL PROBLEMS

A. Branch and Bound method overview

The problem presented in Section III-B is not a pure mixed integer optimal control problem such as in [16]. Indeed, \mathbf{M}^{k_f} is a discrete time variable which can switch from a discrete value to another only at regular time steps ΔT which can be large compared to the battery dynamics. Now, the method presented in [16] converges to the optimal solution only if the time between consecutive switching times converges to zero. This convergence condition is not met for the presented application.

The mixed integer optimal control problem consists in finding both the optimal sequence of discrete valued control variable \mathbf{M}^{k_f} and the continuous optimal controls λ_i . To do so, we use a Branch and Bound approach [18]. Generally speaking, these techniques rely on a systematic exploration of a search-tree representing all possible values of the discrete variables. Throughout this exploration, at each node, the Branch and Bound algorithm completes two sequential tasks:

- **Bounding** : Computing the optimal solution of a relaxed problem for a node of depth k is an optimization problem where the discrete variables have been set over k stages, i.e. from the root to the current node of the search-tree. This relaxed optimization problem provides the algorithm a lower-bound of the actual optimal cost.
- **Branching** : Once the bounding part of the node has been computed, the branching part can be performed. The branching of a node consists in two mutually exclusive actions :

- **Exploring** : If the branch rooted at the current node of depth k is worth exploring, the node's children are computed.

- **Pruning** : If the branch rooted at the current node is not worth exploring, i.e. if the optimal solution of the relaxed problem of the current node is higher than the cost of an admissible trajectory for the considered problem, the branch is then pruned and will not be explored.

The efficiency of a Branch and Bound approach requires to have a relaxed problem which approximates the integer value problem and an exploring method, which explores primarily the nodes located on the optimal path of the tree. Now, let us see how to apply this general method to Problem (13)-(23).

B. Lower bounding

The lower bounding part of the algorithm consists in solving a relaxed problem, i.e. a continuous problem deriving from the original problem where the integer constraints have been relaxed. In order to introduce the relaxed problem let us define the following useful functions

$$g_1(\lambda_i(t), x_i(t), j, \mathbf{M}^k) = \dots \begin{cases} \lambda_i(t) p_{c,i}^+(x_i) - \max_j P_{G_j}^+ & \text{if } t < k\Delta T \\ \lambda_i(t) p_{c,i}^+(x_i) m_{i,j}^s - P_{G_j}^+, & \text{if } t \geq k\Delta T \end{cases} \quad (24)$$

$$g_2(\lambda(t), x(t), \mathbf{M}^k) = \dots \begin{cases} \sum_i \lambda_i(t) p_{c,i}^+(x_i) - P_s, & \text{if } t < k\Delta T \\ \sum_i \lambda_i(t) p_{c,i}^+(x_i) \sum_j m_{i,j}^s - P_s, & \text{if } t \geq k\Delta T \end{cases} \quad (25)$$

Function g_1 (resp. g_2) corresponds to the relaxation of constraint (22) (resp. (23)) over the time interval $[0, k\Delta T]$

$$\begin{aligned} \min_{\lambda_i} J(\lambda, \mathbf{M}^k) = \dots \\ \sum_{i=1}^{n_x} \left[\sum_{s=0}^k \int_{s\Delta T}^{(s+1)\Delta T} \text{price}(t) \lambda_i(t) p_{c,i}^+(x_i(t)) dt \dots \right. \\ \left. + \sum_{s=k+1}^{k_f} \int_{s\Delta T}^{(s+1)\Delta T} \text{price}(t) \sum_{j=1}^{n_{cs}} m_{i,j}^s \lambda_i(t) p_{c,i}^+(x_i(t)) dt \dots \right. \\ \left. + \int_0^{t_f} \theta_x(x_i(t) - x_i(t_{f,i}))^2 dt \right] \quad (26) \end{aligned}$$

under constraints (14)-(16) and where constraints (22)-(23) are relaxed as follows:

$$g_1(\lambda_i(t), x_i(t), j, \mathbf{M}^k) \leq 0 \quad \forall t, i, j \quad (27)$$

$$g_2(\lambda(t), x(t), \mathbf{M}^k) \leq 0 \quad \forall t, i \quad (28)$$

$$\mathbf{M}^k = \langle \mathbf{m}^0, \dots, \mathbf{m}^k \rangle \quad (29)$$

$$\mathbf{m}^s \in \{0, 1\}^{n_x \times n_{cs}} \quad (30)$$

$$\sum_{j=1}^{n_{cs}} m_{i,j}^s \leq 1 \quad (31)$$

$$\sum_{i=1}^{n_x} m_{i,j}^s \leq \text{card}(G_j) \quad (32)$$

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_{cs}} m_{i,j}^s \leq \min\{n_x, n_{cs}\} \quad (33)$$

At this point, several remarks can be made regarding this optimal control problem

- Any node of depth k from the search-tree is uniquely associated to a particular sequence \mathbf{M}^k .
- Notice from equations (24)-(25) that the problem at depth k respects the integer constraints from stage $k_f - k$ to stage k_f and is a continuous problem from stage 0 to stage $k_f - k$. At depth k_f the solution of the relaxed problem satisfies the integer constraints at all stages. Therefore, in this algorithm the setting of the integer matrices \mathbf{m}^k is a time-backward procedure. Although other methods of setting the integer variables are possible (forward time, by random,...), this method proved to be very efficient in practice.

Each node generation consists in solving a problem of the form (26)-(33). Therefore, the computational tractability of the Branch and Bound method strongly relies on the ability to efficiently solve this problem and the solving algorithm must be designed with great attention. The chosen algorithm used to solve this problem is presented in details in Section V.

C. Upper bounding

A leaf node of the tree, i.e. a node of depth k_f is a node for which all discrete variables have been set. Therefore this node satisfies the discrete constraints (17)-(23), i.e. any optimal solution of a leaf node is admissible for problem (13)-(23). Thus, the optimal cost corresponding to the relaxed problem of a leaf node is an upper bound on the optimal cost of the original problem and at least one the leaf nodes is the optimal solution for Problem (13)-(23).

D. Generating children of depth $k + 1$ from node of depth k

At this point, the relaxed problem of depth k has been described. One still has to define the branching part of the algorithm, i.e. how to generate the children of a node of depth k . Since Branch and Bound is an exhaustive exploration method, generating the children of depth $k+1$, of a parent node $n(\mathbf{M}^k)$ actually consists in generating all possible matrices $\mathbf{m}^{k+1} \in \{0, 1\}^{n_x \times n_{cs}}$ satisfying (19)-(21) and solving the problem for each matrix. At first glance generating these matrices is not straightforward. But, there is a one to one correspondence between this set and the set of all possible vehicle to charger connection configurations and one can directly derive the connection matrix \mathbf{m}^s corresponding to any of these configurations.

E. Exploring the tree

The branching part consisting in lower-bounding the optimal cost and generating the children of a node has been covered. To fully describe the Branch and Bound algorithm, one has to specify the exploration part of this algorithm, that is to say in what order the nodes are going to be explored.

1) *Depth-First-Search like strategy*: In order to have an efficient pruning of the tree, it is very important to get a good upper bound on the optimal cost as fast as possible. Indeed, as long as no upper bound has been computed no pruning can be performed. In the same time, the convergence speed of the Branch and Bound method strongly relies on

the ability to prune branches in early stages of the algorithm execution. But, as described in Section IV-C, upper bounds on the optimal cost are computed only in the leaf nodes of the tree. Therefore, it is very important to reach a leaf node as fast as possible. To do so, a Depth-First-Search like technique of exploration is used. Depth-First-Search techniques consist in putting generated children of a node in a stack and then exploring the node on top of the stack. Proceeding that way ensures that the current explored node is always one step deeper than the previous one, and that a leaf node is reached in k_f steps. But, in order to get a good upper bound on the optimal cost, the generated nodes are stacked in decreasing order of their relaxed optimal cost. Doing this way ensures that the Depth-First-Search method follows a path of best children per generation. Indeed, a node is explored, then its best child is explored, then its best child's best child and so on. This heuristic allows the algorithm to find a good upper bound in k_f steps. Moreover, using a backward strategy for setting the integer variable (see equation (24)-(25)) often allows this strategy to find the optimal solution when reaching a leaf node.

2) *Best first search strategy*: Once the Depth-First-Search strategy has reached a leaf node, an upper bound is available. To complete the optimization it is now time to prune as many branch as possible to ensure the optimal solution is reached as fast as possible. To do so, the exploration method is now a Best-First-Search method, the node with the lower relaxed optimal cost is explored first. The data structure behind this Best-First-Search is a min-priority-queue [19] sorting nodes in increasing order of their relaxed optimal cost.

F. Branch and Bound algorithm

The complete Branch and Bound algorithm is presented in Algorithm 1.

Remark 1. *At the end of the Depth-First-Search part of the Branch and Bound algorithm a good solution (often optimal in practice) is reached and can be used as an approximate solution of the problem. This property is particularly useful if the execution time must be controlled.*

V. SOLVING THE LOWER BOUNDING PROBLEM : AN INTERIOR POINT METHOD APPROACH

As stated in Section IV-B, the solving algorithm of the lower-bound problem must be chosen carefully. First, the algorithm must be fast enough to ensure that the node generation will not slow down the tree exploration. Interestingly, at each node generation, one can notice that the solution of the lower-bounding procedure of the parent node is a good initialization for any of its child nodes. Indeed, the only difference between the optimization problems of node k and its children of depth $k + 1$ are functions g_1 and g_2 from equations (24)-(25) on the time interval $[(k - 1)\Delta T, k\Delta T]$. In order to compute the lower bound of each node, Pontryagin Maximum Principle based methods are suitable [12] because of their convergence speed when provided with a good initialization. However, in the framework of optimal control, handling pure state and control constraints such as in equations (15) and (16) is

Algorithm 1: Branch and Bound algorithm

```
1:  $stack \leftarrow Stack()$ 
2:  $ub \leftarrow +\infty$ 
3:  $lb \leftarrow -\infty$ 
4:  $E^{-1} \leftarrow \emptyset$ 
5:  $root \leftarrow node(E^{-1})$ 
6:  $best\_node \leftarrow null$ 
7:  $stack.push(root)$ 
8: while  $ub = +\infty$  do
9:    $parent \leftarrow stack.pop()$ 
10:   $sorted\_children \leftarrow parent.gen\_child\_decreased\_order()$ 
11:  for  $child$  in  $sorted\_children$  do
12:    if  $child.relaxed\_cost < +\infty$  then
13:       $stack.push(child)$ 
14:    end if
15:    if  $child.depth = k_f$  then
16:       $ub \leftarrow child.relaxed\_cost$ 
17:    end if
18:  end for
19: end while
20:  $minpq \leftarrow minPQ()$ 
21: while  $stack.size > 0$  do
22:   $node = stack.pop()$ 
23:  if  $node.relaxed\_cost < ub$  then
24:     $minpq.insert(node)$ 
25:  end if
26: end while
27: while  $minpq.size > 0$  do
28:   $node = minpq.delmin()$ 
29:   $children \leftarrow parent.generate\_children()$ 
30:  for  $child$  in  $children$  do
31:    if  $child.relaxed\_cost < ub$  then
32:      if  $child.depth = t_f$  then
33:         $ub \leftarrow child.relaxed\_cost$ 
34:         $best\_node \leftarrow child$ 
35:      else
36:         $minpq.insert(child)$ 
37:      end if
38:    end if
39:  end for
40: end while
41: return  $best\_node$ 
```

not straightforward [20] and classic off the shelves TPBVP solvers fail to compute the solution of such problems¹. To overcome this difficulty, one can use Interior Point Methods (IPMs) adapted for state and input constraints optimal control problems.

A. Presentation of the interior penalized optimal control problem

In order to solve the lower bounding problem (26)-(33) of the Branch and Bound algorithm, an interior point approach

¹The adjoint variables of the two-point boundary value problems can be discontinuous when the state constraint reaches 0.

[21] is used. This method consists in relaxing constraints (15)-(16) and (27)-(28) through an augmentation of the cost (26) using interior penalization functions. This relaxation depends on a parameter sequence (ε_n) converging to 0. As this sequence converges to 0 the solution of the relaxed optimal solution converges to the solution of the original problem. The penalized problem writes as follows

$$\min_{\lambda \in \Lambda} M(\lambda, \mathbf{M}^k, \varepsilon) = \dots \int_{t_0}^{t_f} \text{price} \sum_i p_{c,i}^+(x_i) \lambda_i + \theta_x (x_i(t) - x_i(t_{f,i}))^2 dt + \varepsilon p_{\text{int}}(x, \lambda, \mathbf{M}^k) \quad (34)$$

under constraints (14) and where

$$p_{\text{int}}(x, \lambda, \mathbf{M}^k) = \int_0^{t_f} \left[\sum_i (\gamma_x(-x_i) + \gamma_x(x_i - E_i^+)) + \gamma_x \circ g_2(\lambda, x, \mathbf{M}^k) + \sum_i \gamma_\lambda(\lambda_i) + \sum_i \sum_j \gamma_x \circ g_1(\lambda_i, x_i, j, \mathbf{M}^k) \right] dt \quad (35)$$

$$\Lambda = L^\infty([0, T]; [0, 1]^{n_x}) \quad (36)$$

$$\gamma_x(x) = \begin{cases} (-x)^{-1.1} & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

$$\gamma_\lambda(x) = -\log(-x) - \log(1-x) \quad (38)$$

This particular choice of penalty is given in [21] and ensures that the optimal solution of the penalized problem strictly satisfies the constraints.

Theorem 1 (From [21]). *If penalty functions γ_x and γ_λ are chosen according to equations (37)-(38) then any optimal solution λ^* of problem (34) with $\varepsilon > 0$ is such that*

$$x_i^*(t) \in (0, E_i^+) \quad (39)$$

$$\lambda_i^*(t) \in (0, 1) \quad (40)$$

$$g_1(\lambda_i^*(t), x_i^*(t), j, \mathbf{M}^k) < 0 \quad (41)$$

$$g_2(\lambda^*(t), x^*(t), \mathbf{M}^k) < 0 \quad (42)$$

B. Solving algorithm

1) *Saturation functions for a fully unconstrained problem:* In order to get a fully unconstrained problem, we use saturation functions on the control:

$$\lambda = \phi(\nu) = \frac{1}{2}(1 + \tanh(\nu)) \quad (43)$$

Using this change of variables the penalized optimal control problem (34) becomes

$$\min_{\nu} M(\phi(\nu), \mathbf{M}^k, \varepsilon) = \dots \int_0^{t_f} \text{price} \sum_i p_{c,i}^+(x_i) \phi(\nu_i) + \theta_x (x_i(t) - x_i(t_{f,i}))^2 dt \dots + \varepsilon p_{\text{int}}(x, \phi(\nu), \mathbf{M}^k) \quad (44)$$

under the following dynamic constraint adapted from (14):

$$\dot{x}_i = f_{c,i}(x_i, \phi(\nu_i)) \quad (45)$$

Thanks to Theorem 1, this change of variables is well posed since the optimal control λ^* for problem (34) + (14) is in $L^\infty([0, t_f]; (0, 1))$, i.e. is strictly interior.

2) *Pontryagin Maximum Principle approach for solving the unconstrained optimal control problem*: Solving the lower bounding problem consists in solving a sequence of unconstrained optimal control problems (UOCs) (44)-(45). To solve this sequence of UOCs, a Pontryagin Maximum Principle (PMP) based approach is used. To do so, let us first define the Hamiltonian [12] of problem (44)-(45)

$$\begin{aligned} H(t, x, \phi(\nu), p, \mathbf{M}^k, \varepsilon) = & \dots \\ \text{price} \sum_i p_{c,i}^+(x_i) \phi(\nu_i) & + \theta_x (x_i(t) - x_i(t_{f,i}))^2 \\ & + \varepsilon p_{\text{int}}(x, \phi(\nu), \mathbf{M}^k) + \sum_i p_i^T f_{c,i}(x_i, \phi(\nu_i)) \end{aligned} \quad (46)$$

The PMP states that any optimal solution from problem (44)-(45) is solution of the following TPBVP

$$\dot{x}_i = f_{c,i}(x_i, \phi(\nu_i)) \quad (47)$$

$$\dot{p}_i = - \frac{\partial H(t, x, \phi(\nu), p, \mathbf{M}^k, \varepsilon)}{\partial x_i} \quad (48)$$

$$0 = \frac{\partial H(t, x, \phi(\nu), p, \mathbf{M}^k, \varepsilon)}{\partial \nu_i} \quad (49)$$

$$x_i(0) = x_{0,i} \quad (50)$$

$$x_i(t_f) = x_{t_f,i} \quad (51)$$

To solve this TPBVP numerous algorithms such as shooting or collocation algorithms exist and an exhaustive presentation is given in [22]. For the problem under consideration, a collocation based solving algorithm such as [23] is a relevant choice. Indeed, these methods allow one to achieve a good compromise between computational speed and numerical sensitivity.

3) *Presentation of the lower bounding algorithm*: In the following we present the resolution algorithm for the lower bounding part of the Branch and Bound Algorithm 1

Algorithm 2: Lower bounding algorithm

- 1: $\varepsilon = \varepsilon_0 > 0, \alpha > 1, \text{tol} > 0$
 - 2: Initialization of (ν, x) such that $x_i \in (0, E_i^+)$;
 $\phi(\nu_i) p_{c,i}^+(x_i) \leq \max_j P_{b,j}$ and $\sum_i \phi(\nu_i) p_{c,i}^+(x_i) \leq P_s$
 - 3: $C_{\text{int}} \leftarrow \perp$
 - 4: **while** $\neg C_{\text{int}}$ **do**
 - 5: (ν, x, p) is solution of (47)-(51)
 - 6: **if** $\varepsilon p_{\text{int}}(x, \phi(\nu), \mathbf{M}^k) / J \circ \phi(\nu) < \text{tol}$ **then**
 - 7: $C_{\text{int}} \leftarrow \top$
 - 8: **else**
 - 9: $\varepsilon = \varepsilon / \alpha$
 - 10: **end if**
 - 11: **end while**
 - 12: **return** $\int_0^{t_f} \text{price} \sum_i p_{c,i}^+(x_i) \phi(\nu_i) dt$
-

The exit condition from Algorithm 2 allows to stop the algorithm when the perturbation on the optimal cost provided

by the penalty function $\varepsilon p_{\text{int}}(x, \phi(\nu), \mathbf{M}^k)$ is negligible with respect to the original cost. The working justification of this exit condition will be detailed in Section VI-A.

VI. CONVERGENCE ANALYSIS

Before proving convergence of the presented method, let us introduce the following Definition

Definition 1 (Lower bound solution). *The lower bound solution of a node n of depth k is a 2-uple $(\mathbf{M}_n^k, \lambda^n)$, where \mathbf{M}_n^k is the sequence of integer matrices corresponding to node n defined in equations (29)-(33) and where λ^n is an optimal solution of problem (26)-(33) with $\mathbf{M}^k = \mathbf{M}_n^k$.*

Now, the proof of convergence is presented in three parts. First, the convergence of the interior point method from Section V is investigated. Then, proof of path-increasing cost is given. Finally the proof of convergence of the method directly stems from the aforementioned proofs.

A. Convergence of the interior point method

In order to prove the convergence of the interior point method, one first needs to prove that the solution of the Two Point Boundary Value Problem (TPBVP) (47)-(51) is indeed the optimal solution of the considered problem. This is the object of the following proposition

Proposition 1. *There exists a unique optimal solution λ^* to Problem (34)-(38) and λ^* is solution of TPBVP (47)-(51)*

Proof: See Appendix A ■

The proof of convergence of the interior point method is a direct use of [21].

Theorem 2 (From [21]). *If the penalty functions are chosen according to equations (37)-(38), any optimal solution $\nu_\varepsilon^*(\mathbf{M}^k)$ of problem (44)-(45) is such that, $\forall k \leq k_f$*

$$\lim_{\varepsilon \rightarrow 0} M(\phi(\nu_\varepsilon^*(\mathbf{M}^k)), \mathbf{M}^k, \varepsilon) = \min_\lambda J(\lambda, \mathbf{M}^k) \quad (52)$$

$$\lim_{\varepsilon \rightarrow 0} J(\phi(\nu_\varepsilon^*(\mathbf{M}^k)), \mathbf{M}^k) = \min_\lambda J(\lambda, \mathbf{M}^k) \quad (53)$$

$$\lim_{\varepsilon \rightarrow 0} \varepsilon p_{\text{int}}(x, \phi(\nu_\varepsilon^*(\mathbf{M}^k)), \mathbf{M}^k) = 0 \quad (54)$$

under constraints (14)-(16)+(27)-(33).

This result yields $\lim_{\varepsilon \rightarrow 0} \frac{\varepsilon p_{\text{int}}(x, \phi(\nu_\varepsilon^*(\mathbf{M}^k)), \mathbf{M}^k)}{J(\phi(\nu_\varepsilon^*(\mathbf{M}^k)), \mathbf{M}^k)} = 0$, thus the exit condition from Algorithm 2

B. Path increasing lower bound procedure

Proposition 2. *Let n_1 and n_2 be two nodes on the same simple path from the root to a leaf of the tree. Without loss of generality, assume that n_1 (resp. n_2) has depth k (resp. s , $s > k$) and let $(\mathbf{M}_{n_1}^k, \lambda^{n_1})$ (resp. $(\mathbf{M}_{n_2}^s, \lambda^{n_2})$) be its lower bounding solution from Definition 1. One has $J(\lambda^{n_1}, \mathbf{M}_{n_1}^k) \leq J(\lambda^{n_2}, \mathbf{M}_{n_2}^s)$. In other words, the lower bounding cost is increasing along any simple path of the search-tree.*

Proof: Since n_1 and n_2 are on the same path connecting the root and a leaf of the search tree, one has $M_{n_2}^s = M_{n_1}^k : \langle m^{k+1}, \dots, m^s \rangle$. Hence, using equation (24) one has

$$g_1(\lambda_i^{n_1}, x_i^{n_1}, j, M_{n_1}^k) \geq \dots \quad (55)$$

$$g_1(\lambda_i^{n_2}, x_i^{n_2}, j, M_{n_2}^k) \forall t \in [0, t_f], \dots \quad (56)$$

$$i = 1, \dots, n_x; j = 1, \dots, n_b \quad (57)$$

where $x^{n_1} = (x_i^{n_1}, \dots, x_{n_x}^{n_1})^T$ is solution of equations (14), with control variable $\lambda^{n_1} = (\lambda_i^{n_1}, \dots, \lambda_{n_x}^{n_1})^T$ and using equation (25) one also has

$$g_2(\lambda^{n_1}, x^{n_1}, M_{n_1}^k) \geq \dots \quad (58)$$

$$g_2(\lambda^{n_2}, x^{n_2}, M_{n_2}^k) \forall t \in [0, t_f] \quad (59)$$

Therefore, the search space of problem (26)-(33) associated to $M_{n_1}^k$ contains the search space associated to $M_{n_2}^s$. This yields $J(\lambda^{n_1}, M_{n_1}^k) \leq J(\lambda^{n_2}, M_{n_2}^s)$. ■

C. Convergence of the Branch and Bound method

We are now ready to state our main convergence result

Theorem 3. *Let node o be the solution of the Branch and Bound algorithm, let $(\lambda^o, M_o^{k_f})$ its lower bounding solution and let $\nu_\epsilon^*(M_o^{k_f})$ be the optimal solution of the lower bounding problem (47)-(51) for node o . The Branch and Bound algorithm converges to the global solution of problem (13)-(23) in the following sense*

$$\begin{aligned} \lim_{\epsilon \downarrow 0} J(\phi(\nu_\epsilon^*(M_o^{k_f})), M_o^{k_f}) &= J(\lambda^o, M_o^{k_f}) \\ &= \min_{M^{k_f}} \min_{\lambda} J(\lambda, M^{k_f}) \end{aligned} \quad (60)$$

under constraints (14)-(23)

Proof: From Theorem 2 and Definition 1 one has

$$\lim_{\epsilon \downarrow 0} J(\phi(\nu_\epsilon^*(M_o^{k_f})), M_o^{k_f}) = \min_{\lambda} J(\lambda, M_o^{k_f}) = J(\lambda^o, M_o^{k_f}) \quad (61)$$

Now, let us prove that node o is the leaf node with lowest lower bounding cost among all leaf nodes. The proof is by contradiction. Assume that there exists a leaf node l with lower bounding solution $(M_l^{k_f}, \lambda^l)$ such that $J(\lambda^l, M_l^{k_f}) < J(\lambda^o, M_o^{k_f})$. This configuration is possible only if during the Branch and Bound algorithm execution a branch containing l rooted at a node m of depth $s < k_f$ with lower bounding solution (M_m^s, λ^m) is pruned. Let c_{ub} be the current upper bound when this pruning happened, according to Proposition 2 one has $c_{ub} \leq J(\lambda^m, M_m^s) \leq J(\lambda^l, M_l^{k_f})$ otherwise no pruning would have occurred. When the execution is completed, the final upper bound is the lower bounding cost of the leaf node o . From the updating method of the Branch and Bound algorithm upper bound one has $J(\lambda^o, M_o^{k_f}) \leq c_{ub} \leq J(\lambda^m, M_m^s) \leq J(\lambda^l, M_l^{k_f})$ for any leaf node l , which contradicts the initial assumption and yields

$$J(\lambda^o, M_o^{k_f}) = \min_{n \in \mathbb{L}} J(\lambda^n, M_n^{k_f}) \quad (62)$$

where \mathbb{L} is the set of all leaf nodes. By construction of the Branch and Bound search-tree, the set $\{M_n^{k_f}\}_{n \in \mathbb{L}}$ is exactly the set of all discrete sequences of length k_f which yields

$$J(\lambda^o, M_o^{k_f}) = \min_{n \in \mathbb{L}} J(\lambda^n, M_n^{k_f}) = \min_{M^{k_f}} \min_{\lambda} J(\lambda, M^{k_f}) \quad (63)$$

and concludes the proof. ■

VII. NUMERICAL PERFORMANCE ANALYSIS

A. Example presentation

In this section the following n_x -dependent problem is considered

$$n_c = 2; P_c = \{50, 22\}; P_s = 60 \quad (64)$$

$$\Delta T = 30 \text{ minutes} \quad (65)$$

$$\theta_x = 0 \quad (66)$$

$$t_{0,k} = 0, ; t_{f,k} \in U(14, 18) \quad (67)$$

$$E_k^+ \in U(30, 90) \quad (68)$$

$$x_{0,k} \in U\left(\frac{E_k^+}{10}, \frac{E_k^+}{5}\right), ; x_{f,k} \in U\left(\frac{7}{10}E_k^+, \frac{9}{10}E_k^+\right) \quad (69)$$

$$\text{price} = \begin{cases} 0, 095 & \text{if } t \in [6, 9) \cup [12, 14) \\ 0.07 & \text{otherwise} \end{cases} \quad (70)$$

where $U(x, y)$ is the uniform distribution on $[x, y]$. From this setting with n_x vehicles and 2 different chargers, the size of the Branch and Bound search-tree is exactly the number of permutations of 2-elements among n_x to the power of the time steps number, that is to say $\left(\frac{n_x!}{(n_x-2)!}\right)^{36}$.

B. Reference charging strategy

In order to benchmark the proposed method, we consider the following reference strategy. The idea is to charge two vehicles which need the larger amount of energy at the switching moment. More precisely, at switching time t_s , we note $i_1(t_s)$ (resp. $i_2(t_s)$) the vehicle such that $x_{f,i_1(t_s)} - x_{i_1(t_s)}(t_s)$ is maximum (resp. second to maximum) and let $p_{c,k}$ be the charging power of vehicle k . The charging strategy is as follows

$$p_{c,i_1(t_s)}(t) = \min\{50, p_{c,i_1(t_s)}(x_{i_1(t_s)}(t))\} \quad (71)$$

$$p_{c,i_2(t_s)}(t) = \min\{22, p_{c,i_2(t_s)}(x_{i_2(t_s)}(t)); 60 - p_{c,i_1(t_s)}(t)\} \quad (72)$$

$$p_{c,k}(t) = 0 \quad \forall k \in \{1, \dots, n_x\} \setminus \{i_1(t_s), i_2(t_s)\} \quad (73)$$

where $t \in [t_s, t_{s+1})$. Note that once the state of energy of vehicle k has reached $x_{f,k}$, the charging power of this vehicle is always 0.

C. Performance analysis indicators

For each value of n_x we solve 50 instances of problem (64)-(70) both with reference and presented method. Let $e_{opt}^i(n_x)$ (resp. $e_{ref}^i(n_x)$) be the optimal solution of instance $i = 1, \dots, 50$, of problem (64)-(70). In the same way, we note $c_{opt}^i(n_x)$ (resp. $c_{ref}^i(n_x)$) the optimal (resp. reference) charging cost corresponding to the solving of instance

n_x	Energy		Cost		Execution time	
	RMES (%)	ERSD (%)	RMCS (%)	CRSD (%)	mean (s)	std (s)
9	1.59	1.03	6.01	1.14	570	96
8	1.92	0.95	3.79	1.23	261	43
7	2.73	1.28	2.92	1.22	169	29
6	3.21	1.22	3.12	1.22	60	4
5	5.14	1.60	5.05	1.59	47	3
4	7.11	1.61	7.00	1.60	35	3
3	7.37	1.98	7.26	1.96	30	6

TABLE I

PERFORMANCE INDICATORS AS A FUNCTION OF THE NUMBER OF VEHICLES TO BE CHARGED n_x

$i = 1, \dots, 50$, of problem (64)-(70). In addition, we note $\delta_e^i(n_x) = \frac{e_{ref}^i(n_x) - e_{opt}^i(n_x)}{e_{ref}^i(n_x)}$ (resp. $\delta_c^i(n_x) = \frac{c_{ref}^i(n_x) - c_{opt}^i(n_x)}{c_{ref}^i(n_x)}$) the relative energy (resp. cost) savings. In order to quantify the performances of the proposed method, the following indicators are used

- Relative Mean Energy Savings, RMES(n_x):

$$\text{RMES}(n_x) = 100\mathbb{E}_i \{ \delta_e^i(n_x) \} \quad (74)$$

- Energy Relative Standard Deviation, ERSD(n_x):

$$\text{ERSD}(n_x) = 100\sqrt{\mathbb{E}_i \{ \delta_e^i(n_x)^2 \} - \mathbb{E}_i \{ \delta_e^i(n_x) \}^2} \quad (75)$$

- Relative Mean Cost Savings, RMCS(n_x):

$$\text{CRMS}(n_x) = 100\mathbb{E}_i \{ \delta_c^i(n_x) \} \quad (76)$$

- Cost Relative Standard Deviation, CRSD(n_x):

$$\text{CRSD}(n_x) = 100\sqrt{\mathbb{E}_i \{ \delta_c^i(n_x)^2 \} - \mathbb{E}_i \{ \delta_c^i(n_x) \}^2} \quad (77)$$

where \mathbb{E}_i is the mathematical expectation with respect to the instance number i .

D. Compared performance analysis

1) *Energy consumption comparison:* On Table I, one can see that the relative energy savings decreases with the number of vehicle to be charged. This phenomenon is a consequence of the fact that fewer vehicles to charge over a given time period allows to use lower power charging limiting the energy losses in the process. In contrast, the reference method aims at charging EVs as fast as possible to the expense of energy losses.

2) *Cost savings comparison:* On Table I, the behavior of the RMCS as a function of n_x is more complex, it decreases from $n_x = 3$ to $n_x = 7$ and increases from $n_x = 7$ to $n_x = 9$. First, for $n_x = 3, \dots, 7$, the cost savings is a decreasing function of n_x . On Table II, one can see that the percentage of energy consumed during peak hours is zero² for reference and optimal strategies when $n_x < 7$. Since the whole energy is consumed outside peak hours, the percentage of cost decrease is equal to the percentage of energy decrease. From Table I, the CRMS is now an increasing function of n_x , for $n_x \geq 7$. This increase phenomenon is due to the fact that the main

²In the optimal case, the peak-hours energy is not exactly zero. This stems from using IPMs to solve the optimal control which leaves a residual non zero consumption during these periods.

n_x	9	8	7	6	5	4	3
Optimal (%)	0.32	0.29	0.29	0.26	0.30	0.31	0.33
Reference (%)	13.54	5.76	0.80	0.00	0.00	0.00	0.00

TABLE II

PERCENTAGE OF ENERGY CONSUMED DURING PEAK HOURS FOR OPTIMAL AND REFERENCE STRATEGY.

source of cost savings does not come from limiting energy losses anymore but from shifting energy consumption from the peak periods. On Table II, one can see that the percentage of energy consumed during peak hours increases when $n_x \geq 7$ in the reference case but is still 0 with the optimal strategy. Therefore, the more energy is consumed during peak hours for the reference strategy the greater the cost savings using the proposed method.

3) *Execution time analysis:* All instances of Problem (64)-(70) are solved on a Linux Ubuntu operating system endowed with 35 Intel®Xeon®Gold 5120 CPU @ 2.2 GHz. The solver is based on homemade PYTHON solvers both for the Branch and Bound algorithm and the Differential Algebraic Equations. On Table I one can see that the growth in execution time suddenly increases for $n_x \geq 7$. This phenomenon stems from the fact that for $n_x \geq 7$ there are more children by generation than CPUs available. Hence the children generation part of the Branch and Bound algorithm cannot be fully parallelized. That said, one can see that the mean execution time is always less than 1% of the time horizon. Finally, with this hardware setting the computation time is compatible with an online implementation. Nevertheless, the presented method requires lots of computing power. Specifically, this methods relies on the availability of using many CPUs at the same time which can be a limit for online implementation.

VIII. MPC IMPLEMENTATION SIMULATION

A. MPC implementation algorithm

In this Section, we are interested in comparing the performances of the reference method presented in Section VII-B and the presented algorithm in an online implementation simulation. For this simulation, we implement the algorithm in a straightforward Model Predictive Control manner [24] consisting in the following step:

- At each vehicle arrival solve the optimal charging problem corresponding to the charging of the cars parked in the co-charging station.
- Execute the computed planning until new vehicle arrival.

This implementation does not require any prediction on the arrival time of a vehicle. Each vehicle is taken into account in the optimization problem as soon as it arrives at the charging station. Therefore, if the charging speed is too low handling a large flow of vehicle arrival might be impossible. In order to gain some robustness with respect to the flow of vehicles the weight on the quadratic cost on the state of charge θ_x can be tuned to achieve a good trade-off between minimization of the time-of-use charging cost and the charging speed.

	Energy consumption (kWh)	Charging cost (€)
Optimal case	197.7	14.72
Reference case	210.6	17.82
Variation	-6.13%	-17.46%

TABLE III

COMPARISON OF REFERENCE AND OPTIMAL STRATEGY IN MPC SIMULATION IN TERMS OF ENERGY CONSUMPTION DURING BASE AND PEAK HOURS.

	Peak-hours Energy(kWh)	Base-hours Energy(kWh)
Optimal case	15.09	182.61
Reference case	61.25	149.35
Variation	-75.37%	+22.27%

TABLE IV

COMPARISON OF REFERENCE AND OPTIMAL STRATEGY IN MPC SIMULATION IN TERMS OF ENERGY CONSUMPTION DURING PEAK AND BASE HOURS.

B. Simulation Parameters

The MPC simulation is carried out with the following parameters

$$n_c = 2 ; P_c = \{20, 10\} ; P_s = 25 \quad (78)$$

$$\Delta T = 30 \text{ minutes}, ; \theta_x = 10^{-3} \quad (79)$$

$$E_k^+ \in U(5, 15) \quad (80)$$

$$x_{0,k} \in U\left(\frac{E_k^+}{10}, \frac{E_k^+}{5}\right), ; x_{f,k} \in U\left(\frac{7E_k^+}{10}, \frac{9E_k^+}{10}\right) \quad (81)$$

$$\text{price} = \begin{cases} 0,095 & \text{if } t \in [6, 9) \\ 0,150 & \text{if } t \in [18, 21) \\ 0,07 & \text{otherwise} \end{cases} \quad (82)$$

In addition the probability of a car arrival of a car between two plug-times (30 min) of the charger is 40%. The simulation is carried out on 48 hours.

C. Simulation Results

This comparative simulation required the solving of 41 problems corresponding to 41 vehicle arrival. At max, there were 6 cars charging at the same time. Table IV shows that the optimal strategy is both energy and time-of-use cost efficient. The overall energy savings stems from a slower but more efficient charging of the vehicles. The lower charging rate is particularly visible by comparing Figures 1 and 2 where it is clear that the required final state-of-energy is reached sooner with reference than optimal strategy. Moreover, Table III shows that the optimal strategy achieves to reduce by 75% the energy consumption during peak periods. This energy consumption is simply transferred to off-peak periods. This transfer is the most important factor of charging-cost reduction. In addition, one can see on Figures 3 and 4 that the optimal charging strategy takes advantage of the possibility of regularly changing the vehicles connected to the charging stations. This increase rate of plug-in plug-out of vehicles allows to reduce the charging speed while satisfying the required state of energy target.

IX. CONCLUSION

This paper presents a methodology to compute an optimal charging scheduling of electric vehicles at a co-charging

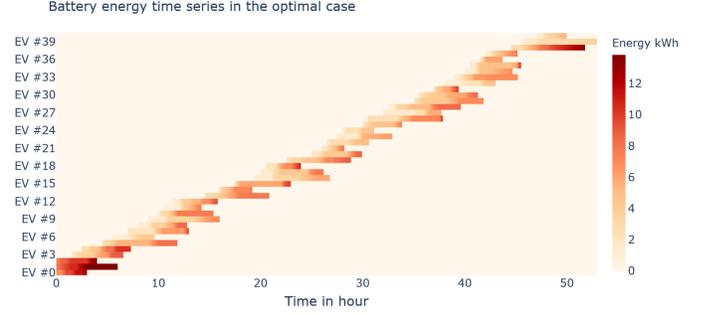


Fig. 1. Optimal energy time series for MPC simulation

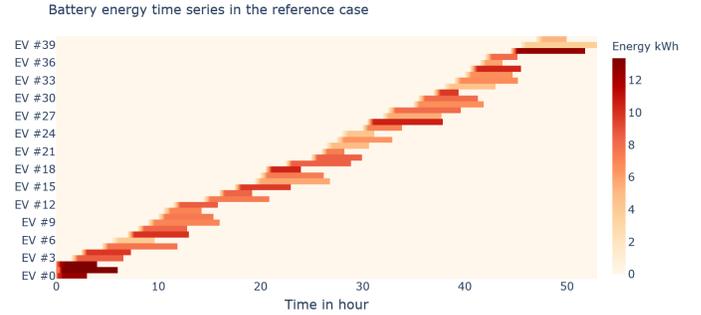


Fig. 2. Reference energy time series for MPC simulation

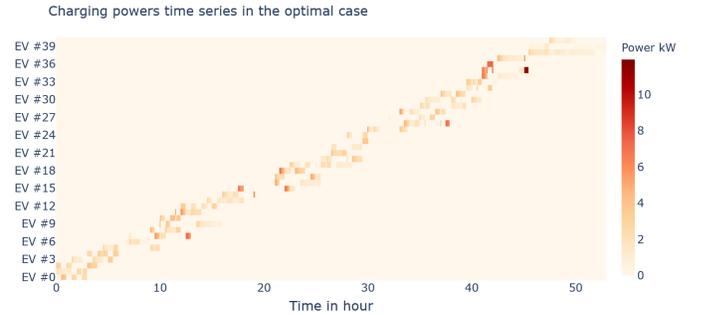


Fig. 3. Optimal charging time series for MPC simulation

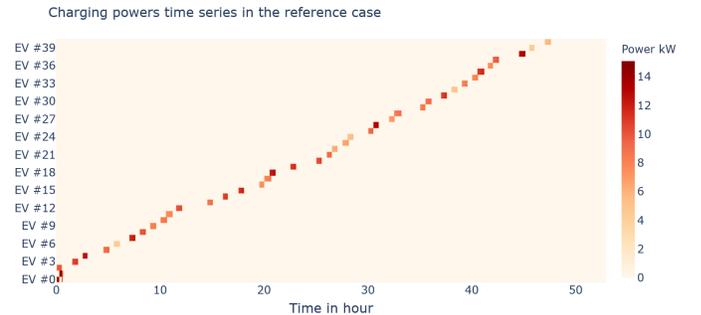


Fig. 4. Reference charging time series for MPC simulation

station. The presented methodology relies on a Branch and Bound strategy allowing to compute the global optimum. This methodology strongly relies on Maximum Principle based optimal control solver allowing to compute quickly a tight lower bound at each node of the Branch and Bound tree. It also relies on the slower time scale of the charging stations compared to the time scale of vehicles batteries to produce an efficient Branch and Bound algorithm. However, in order to be working at full potential this algorithm requires to run on a cluster so that the children generation can be fully parallelized. This need of computing power might hold back online implementations of this method. Nevertheless, thanks to the global convergence of the method, it can be used offline to assess the performances of heuristic-based methods in terms of distance to the optimal solution. In future work, the authors will take into account uncertainties on the departure time and arrival time of the electric vehicles. A progressive hedging approach [25] might be considered to use the presented algorithm in a stochastic framework.

APPENDIX

A. Proof of Proposition 1

First, let us recall the Inverse Function Theorem which will be used throughout the proof.

Theorem 4 (Inverse Function Theorem [26]). *Let $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ be continuously differentiable on some open set containing a , and suppose $\det(\nabla f(a)) \neq 0$. Then there is some open set V containing a and an open set W containing $f(a)$ such that $f : V \mapsto W$ has a continuous inverse $f^{-1} : W \mapsto V$ which is differentiable for all $y \in W$ and one has*

$$\nabla [f^{-1}] (y) = [\nabla f]^{-1} (f^{-1}(y))$$

In addition, the Ky Fan Dominance Theorem recalled here after will also be used

Theorem 5 (K. Fan Dominance Theorem [27]). *Let \mathcal{M}_n be the space of $n \times n$ real matrices. For all $A \in \mathcal{M}_n$, we note $(s_i(A))_{i=1, \dots, n}$ the sequence of singular values in decreasing order. For $1 \leq k \leq n$, the Ky Fan k -norm on \mathcal{M}_n is defined as*

$$\| A \|_{(k)} = \sum_{i=1}^k s_i(A)$$

These norms are unitarily invariant, i.e., $\| UAV \|_{(k)} = \| A \|_{(k)}$ and for given matrices $A, B \in \mathcal{M}_n$

$$\| A \|_{(k)} \leq \| B \|_{(k)} \Leftrightarrow \| A \| \leq \| B \|$$

$\forall k = 1, \dots, n$ and for all unitarily invariant norms $\| \cdot \|$, such as the Euclidian $\| \cdot \|_2$ norm.

The proof of Proposition 1 follows the following outline

- First, we prove that any optimal trajectory from Problem (34)-(38) does not contain any singular arc in Section A1.
- Then, we prove that any optimal solution of Problem (34)-(38) is characterized by the stationarity conditions (47)-(51) in Section A2.
- Finally, we prove that there exists a unique solution from Problem (47)-(51) in Section A3.

1) *Absence of singular arcs:* Since the control variable λ appears explicitly in $\frac{\partial}{\partial \lambda} H(t, x(t), \lambda, p(t), \mathbf{M}^k, \varepsilon)$ for all values of $t, x(t), p(t), \mathbf{M}^k, \varepsilon$, there are not singular arcs along the optimal trajectory, i.e. it is always possible to compute a minimizer of $H(t, x(t), \lambda, p(t), \mathbf{M}^k, \varepsilon)$ with respect to $\lambda \in [0, 1]^{n_x}$.

2) *Stationarity conditions of Pontryagin's Maximum Principle:* From Theorem 1, any optimal solution of Problem (34)-(38) is strictly interior to the constraints. Therefore, any optimal solution λ^* of this problem satisfies the following stationarity condition:

$$\frac{\partial H(t, x(t), \lambda^*(t), p(t), \mathbf{M}^k, \varepsilon)}{\partial \lambda} = 0$$

3) *Uniqueness of solutions of the stationarity conditions:* In the following, for any mapping $\varphi : X \times Y \mapsto Z$, we note $\varphi[x](\cdot)$, the mapping $\varphi(x, \cdot) : Y \mapsto Z$ where x is fixed and considered as a parameter. Using that notation we introduce the following functions

$$\begin{aligned} F[t, x(t), p(t)](\lambda) &= \text{price}(t) + \frac{\partial f_c(x(t), \lambda p_c^+(x(t)))}{\partial \lambda} p(t) \\ D[x(t), \mathbf{M}^k](\lambda) &= \frac{\partial}{\partial \lambda} \sum_i \gamma_\lambda(\lambda_i) \\ E[x(t), \mathbf{M}^k](\lambda) &= \frac{\partial g_2}{\partial \lambda}(\lambda, x(t), \mathbf{M}^k) \gamma'_x \circ g_2(\lambda, x(t), \mathbf{M}^k) \\ &+ \sum_{i,j} \frac{\partial g_1}{\partial \lambda}(\lambda_i, x_i(t), j, \mathbf{M}^k) \gamma'_x \circ g_1(\lambda_i, x_i(t), j, \mathbf{M}^k) \\ \Gamma[x(t), \mathbf{M}^k](\lambda) &= D[x(t), \mathbf{M}^k](\lambda) + E[x(t), \mathbf{M}^k](\lambda) \end{aligned}$$

and one has:

$$\begin{aligned} \frac{\partial H(x(t), \lambda, p(t), \mathbf{M}^k, \varepsilon, t)}{\partial \lambda} &= \dots \\ F[t, x(t), p(t)](\lambda) + \varepsilon \Gamma[x(t), \mathbf{M}^k](\lambda) \end{aligned}$$

$\Gamma[x(t), \mathbf{M}^k](\lambda)$ is the gradient of the sum of convex functions over a convex set. Hence, the corresponding Hessian is strictly definite positive and invertible. Moreover, one has

$$\begin{aligned} \lim_{\lambda \rightarrow 0^+} \Gamma_i[x(t), \mathbf{M}^k](\lambda) &= -\infty, \quad i = 1, \dots, n_x \\ \lim_{\lambda \rightarrow 1^-} \Gamma_i[x(t), \mathbf{M}^k](\lambda) &= +\infty, \quad i = 1, \dots, n_x \end{aligned}$$

Therefore, $\Gamma[x(t), \mathbf{M}^k]$ is a continuous mapping from $(0, 1)^{n_x}$ to \mathbb{R}^{n_x} . Since x, p are bounded and since $F[t, x(t), p(t)]$ is continuous $\frac{F[t, x(t), p(t)]}{\varepsilon}$ is also bounded for all $\varepsilon > 0$. Therefore, one can use the Inverse Function Theorem which gives

$$\begin{aligned} \lambda &= \left[\Gamma[x(t), \mathbf{M}^k] \right]^{-1} \left(-\frac{F[t, x(t), p(t)](\lambda)}{\varepsilon} \right) \\ &= r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda) \end{aligned} \quad (83)$$

In order to prove existence and uniqueness of solutions of (83), one needs to prove that the mapping $r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\cdot)$ is a contraction.

$$\begin{aligned} &\| r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda_1) - r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda_2) \| \dots \\ &\leq \sup_\lambda \| \nabla r[t, x(t), \mathbf{M}^k, p(t), \varepsilon](\lambda) \| \| \lambda_1 - \lambda_2 \| \end{aligned}$$

Plus, one has

$$\sup_{\lambda} \|\nabla r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda)\| = \dots$$

$$\sup_{\lambda} \|\nabla \left[\Gamma[x(t), p(t), \mathbf{M}^k]^{-1} \left(-\frac{F[t, x(t), p(t)](\lambda)}{\varepsilon} \right) \right]\|$$

From the inverse function theorem one has

$$\sup_{\lambda} \|\nabla \left[\Gamma[x(t), p(t), \mathbf{M}^k]^{-1} \left(-\frac{F[t, x(t), p(t)](\lambda)}{\varepsilon} \right) \right]\| = \dots$$

$$\sup_{\lambda} \left\| \left[\nabla \Gamma[x(t), p(t), \mathbf{M}^k] \right]^{-1} \left(\dots \right. \right.$$

$$\left. \left. \left[\Gamma[x(t), p(t), \mathbf{M}^k] \right]^{-1} \left(-\frac{F[t, x(t), p(t)](\lambda)}{\varepsilon} \right) \right) \right\|$$

$$= \sup_{\lambda} \left\| \left[\nabla^2 D[x(t), \mathbf{M}^k] + \nabla^2 E[x(t), \mathbf{M}^k] \right]^{-1} \left(\dots \right. \right.$$

$$\left. \left. \left[\Gamma[x(t), p(t), \mathbf{M}^k] \right]^{-1} \left(-\frac{F[t, x(t), p(t)](\lambda)}{\varepsilon} \right) \right) \right\|$$

Since $\nabla^2 D[x(t), \mathbf{M}^k](\lambda)$ and $\nabla^2 E[x(t), \mathbf{M}^k](\lambda)$ are positive symmetric definite matrices, their singular values are the eigenvalues and $s_n(\nabla^2 D[x(t), \mathbf{M}^k](\lambda) + \nabla^2 E[x(t), \mathbf{M}^k](\lambda)) \geq s_n(\nabla^2 D[x(t), \mathbf{M}^k](\lambda))$, where $(s_n(\cdot))_n$ is the sequence of eigenvalues in decreasing order. For any Ky-Fan norm [28] $\|\cdot\|_{(k)} = \sum_{i=1}^k s_i(\cdot)$, one has

$$\left\| \left[\nabla^2 D[x(t), \mathbf{M}^k] + \nabla^2 E[x(t), \mathbf{M}^k] \right]^{-1}(\lambda) \right\|_{(k)} = \dots$$

$$\sum_{i=n_x-k+1}^{n_x} \frac{1}{s_i \left(\nabla^2 D[x(t), \mathbf{M}^k](\lambda) + \nabla^2 E[x(t), \mathbf{M}^k](\lambda) \right)}$$

$$\leq \sum_{i=n_x-k+1}^{n_x} \frac{1}{s_i \left(\nabla^2 D[x(t), \mathbf{M}^k](\lambda) \right)}$$

$$= \left\| \left[\nabla^2 D[x(t), \mathbf{M}^k] \right]^{-1}(\lambda) \right\|_{(k)}$$

$$\leq \sup_{\lambda \in (0,1)} \frac{1}{\gamma_{\lambda}''(\lambda)} = \frac{1}{\gamma_{\lambda}''(0.5)} = \frac{1}{8}$$

From Theorem 4 one has

$$\|r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda_1) - r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\lambda_2)\|_2 \leq \frac{1}{8} \|\lambda_1 - \lambda_2\|_2$$

Mapping $r[t, x(t), p(t), \mathbf{M}^k, \varepsilon](\cdot)$ is $\frac{1}{8}$ -Lipschitz, hence it is a contraction. Using Banach fixed point Theorem one has that for all $t, x(t), p(t), \varepsilon, \mathbf{M}^k$, there exists a unique solution λ^* to TPBVP (47)-(51).

REFERENCES

- [1] "Initial contact," <https://www.volkswagen-newsroom.com/en/press-releases/initial-contact-the-mobile-charging-robot-presenting-a-vision-6736>.
- [2] T. Liu, H. Yu, H. Guo, Y. Qin, and Y. Zou, "Online energy management for multimode plug-in hybrid electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4352–4361, 2019.
- [3] D. Zhou, A. Al-Durra, I. Matraji, A. Ravey, and F. Gao, "Online energy management strategy of fuel cell hybrid electric vehicles: A fractional-order extremum seeking method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6787–6799, 2018.
- [4] R. Xiong, J. Cao, and Q. Yu, "Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle," *Applied Energy*, vol. 211, pp. 538–548, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261917316707>
- [5] T. Morstyn, C. Crozier, M. Deakin, and M. D. McCulloch, "Conic optimization for electric vehicle station smart charging with battery voltage constraints," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 478–487, 2020.
- [6] T. Rui, C. Hu, G. Li, J. Tao, and W. Shen, "A distributed charging strategy based on day ahead price model for pv-powered electric vehicle charging station," *Applied Soft Computing*, vol. 76, pp. 638–648, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494618307270>
- [7] L. Zhang and Y. Li, "Optimal management for parking-lot electric vehicle charging by two-stage approximate dynamic programming," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1722–1730, 2017.
- [8] P. You, Z. Yang, M. Chow, and Y. Sun, "Optimal cooperative charging strategy for a smart charging station of electric vehicles," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2946–2956, 2016.
- [9] C. Sabillon, J. Franco, M. J. Rider, and R. Romero, "A new methodology for the optimal charging coordination of electric vehicles considering vehicle-to-grid technology," *IEEE Transactions on Sustainable Energy*, vol. 7, pp. 1–12, 04 2016.
- [10] Y. Zheng, Y. Song, D. Hill, and K. Meng, "Online distributed mpc-based optimal scheduling for ev charging stations in distribution systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 02 2019.
- [11] O. Salah and A. Oulamara, "Simultaneous electric vehicles scheduling and optimal charging in the business context: case study," in *5th IET Hybrid and Electric Vehicles Conference (HEVC 2014)*, 2014, pp. 1–6.
- [12] A. Bryson and Y.-C. Ho, *Applied optimal control*. Hemisphere Publishing, New-York, 1975.
- [13] H. Sussman, "A maximum principle for hybrid optimal control problem," in *38th IEEE Conference on Decision and Control*, 1999.
- [14] M. Buss, M. Glocker, M. Hardt, O. von Stryk, R. Bulirsch, and G. Schmidt, "Nonlinear hybrid dynamical systems: Modeling, optimal control, and applications," in *Modelling, Analysis, and Design of Hybrid Systems*, S. Engell, G. Frehse, and E. Schnieder, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 311–335.
- [15] S. Sager, "Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control," *Journal of Process Control*, vol. 19, pp. 1238–1247, 2009.
- [16] S. Sager, H. Bock, and M. Diehl, "The integer approximation error in mixed-integer optimal control," *Mathematical Programming*, vol. 133, pp. 1–23, 06 2010.
- [17] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems*. Springer, 2013.
- [18] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization, algorithms and complexity*, 2nd ed. Dover edition, 1998, ch. 8.
- [19] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009, ch. 6.
- [20] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Review*, vol. 37, no. 2, pp. 181–218, 1995.
- [21] P. Malisani, F. Chaplais, and N. Petit, "An interior penalty method for optimal control problems with state and input constraints of non linear systems," *Optimal Control Applications and Methods*, vol. 37, no. 1, 2014.
- [22] U. Ascher, R. Mattheij, and R. Russel, *Numerical solution of boundary value problems for ordinary differential equations*. Prentice Hall, 1988.
- [23] L. Shampine and J. Kierzenka, "Solving boundary value problems for ordinary differential equations in matlab with bvp4c," Matlab Tutorial.
- [24] M. Morari and J. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4, pp. 667–682, 1999.
- [25] R. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, pp. 119–147, 1991.
- [26] V. Zorich, *Mathematical Analysis, part I*, 2nd ed. Springer, 2015, ch. 8, p. 499.
- [27] K. Fan and A. Hoffman, "Some metric inequalities in the space of matrices," *Proceedings of American Mathematics Society*, vol. 6, pp. 111–116, 1955.
- [28] R. Bhatia, *Matrix Analysis*. Springer-Verlag, New York, 1997.