Course of Control of Autonomous Multi-agent Systems

A survey about decentralized control of PEV recharge

Dumitru Demerji
2073949

Simone Orelli
1749732

Antonio Rapuano
2044902

DIAG Department
University of Rome "La Sapienza"
[surname].[id number]@studenti.uniroma1.it

May 2024

Abstract

This document presents a comprehensive review of decentralized solution methods for Mixed-Integer Linear Programming (MILP) problems. These problems are essential in various fields due to their ability to model complex decision-making scenarios, characterized by linear relationships involving both continuous and integer variables. The review begins with a general problem formulation and an explanation of the centralized solution approach. It then explores the evolution of decentralized solutions, highlighting their advantages and disadvantages. A case study on the decentralized control of the charging process for a large fleet of plug-in electric vehicles (PEVs) in a smart grid environment is provided to illustrate practical applications. The case study uses the solution method proposed by Manieri et al. (2023), demonstrating the practical implications and challenges of decentralized solution methods in real-world settings.

Contents

1 Introduction

Mixed-integer linear programming (MILP) problems are a class of optimization problems, pivotal in various fields due to their ability to model complex decision-making scenarios, characterized by linear relationships involving both continuous and integer variables.

These typically involve maximizing or minimizing a linear objective function subject to a set of linear constraints. The presence of integer variables makes the problem particularly challenging to solve, often requiring sophisticated algorithms like branch-and-bound, branch-and-cut, and cutting plane methods.

This document aims to provide a comprehensive review of the solution methods for MILP problems in a decentralized setting, meaning that the optimization problem is solved collaboratively by multiple agents, each controlling a subset of the decision variables.

In order to provide a clear understanding of the problem, we first present it in a general form and explain how it can be solved in a centralized way. Then, we go through the evolution of the decentralized solutions proposed by the literature and discuss the pros and cons of each method.

Alongside the theoretical aspects, we carry forward a case study regarding the decentralized control of the charging process of a large fleet of plug-in electric vehicles (PEVs) in a smart grid environment using the approach proposed by Manieri et al.[1] in 2023. This study will help to understand the practical implications of the decentralized solution methods and the challenges faced in real-world applications.

1.1 Problem formulation

A general form of MILP problem is the following:

$$\min_{v_1,\ldots,v_m} \sum_{p=1}^{m} c_p^T v_p$$

$$\text{subject to: } \sum_{p=1}^{m} A_p v_p \leq b \qquad (\mathcal{P})$$

$$v_p \in V_p, \quad p = 1, \ldots, m.$$

Here, $b \in \mathbb{R}^{\beta}$ is called the *resource array*, and $V_p = \{v_p \in \mathbb{R}^{n_{c,p}} \times \mathbb{Z}^{n_{d,p}} \ s.t. \ D_p v_p \leq d_p\}$ are the *subsystems*. We further assume that the total number of subsystems is finite and greater than the length of the resource array.

Problem $\mathcal{P}$ is capable of modeling any scenario for which a large number of subproblems, each with its own domain V_p possibly including integer variables, are coupled through globally-valid constraints of the form $\sum_{p=1}^{m} A_p v_p \leq b$. The latter are introduced in order to limit the available resources to be shared among the subsystems[2].

1.2 Centralized solution

The centralized solution to problem $\mathcal{P}$ consists of solving the problem as a whole, i.e., compute within a central entity the optimal solution with respect to the cost function by considering all the decision variables $v_1, \ldots, v_m$ simultaneously, with each of these entailing a set of local constraints to satisfy besides the global ones.

Since a closed form solution to this problem is generally not available, the most common way to solve it is by relaxing the integer constraints and solving the resulting linear programming (LP) problem, which means temporarily ignoring the integer constraints and allowing all variables to be continuous. This is known as the *linear relaxation* of the MILP problem. Then, the solution can be refined by using branch-and-bound algorithms, cutting plane methods, and heuristics to find the best integer-feasible solution.

An issue with this approach is that it requires a central entity to have access to all the decision variables, which might not be feasible in practice due to privacy concerns, communication constraints, or computational limitations. Moreover, centralized solutions often struggle with scalability as the size of the problem increases, requiring significant memory and processing power that may exceed the capabilities of a single central processor. For these reasons, decentralized solutions have been proposed as an alternative to the centralized approach.

2 Evolution of decentralized solutions

The decentralized approach aims to distribute the computational burden among multiple agents and reduce the communication overhead. The idea is to divide the whole problem into m sub-problems that can be solved by m separate agents. Unfortunately, the presence of the global constraints is a clear hindrance. A plausible solution is to introduce an array λ, having same dimension as b, of non-negative Lagrange multipliers to dualize the global constraints. This allows us to formulate the following dual problem of $\mathcal{P}$:

$$\max_{\lambda \geq 0} \left\{ -\lambda^T b + \sum_{p=1}^{m} \min_{v_p \in V_p} \left\{ (c_p^T + \lambda^T A_p) v_p \right\} \right\}. \quad (\mathcal{D})$$

The latter is composed of m subproblems, called *inner problems*, defined as:

$$\min_{v_p \in V_p} \left\{ (c_p^T + \lambda^T A_p) v_p \right\}, \quad p = 1, ..., m. \qquad (\mathcal{I})$$

The optimal solution λ^* of $\mathcal{D}$ can be used to solve the inner problem for each agent $p = 1, 2, \ldots, m$, and it can be found through a decentralized version of the sub-gradient iterative algorithm, based on the following update law:

$$\lambda(k+1) = \left[\lambda(k) + \alpha(k) \left(\sum_{p=1}^{m} A_p v_p(\lambda(k)) - b \right) \right]_+ , \qquad (2.1)$$

where the operator $[\cdot]_+$ is equivalent to $\max\{0, \cdot\}$ and ensures the non-negativity of λ, $v_p(\lambda(k))$ is the solution of $\mathcal{I}$ with $\lambda = \lambda(k)$, and the step-size $\alpha(k)$ is chosen according to the Robbins-Monro conditions[3]. However, this approach only penalizes without fully preventing the violation of the global constraints of $\mathcal{P}$. Therefore the corresponding primal solution $v(\lambda^*) = \begin{bmatrix} v_1(\lambda^*) & v_2(\lambda^*) & \ldots & v_m(\lambda^*) \end{bmatrix}$ may not be feasible (nor is it possible to recover feasibility through the standard procedures[4]).

2.1 Shared resources restriction

Vujanic et al. (2014)[2] further explores the properties of the solutions of $\mathcal{I}$. One of the main results therein introduced is that the larger the number of agents m, the lower the duality gap between $\mathcal{P}$ and $\mathcal{D}$:

$$\lim_{m \to \infty} \frac{J_{\mathcal{D}}^*}{J_{\mathcal{P}}^*} = 1.$$

A particular case occurs when considering the linear program corresponding to the relaxation of $\mathcal{P}$:

$$\min_{v_1, \ldots, v_m} \sum_{p=1}^{m} c_p^T v_p$$
$$\text{subject to: } \sum_{p=1}^{m} A_p v_p \leq b \qquad (\mathcal{P}_{LP})$$
$$v_p \in \text{conv}(V_p), \quad p = 1, \ldots, m,$$

with $\text{conv}(V_p)$ being the convex hull of V_p, and where $J_{\mathcal{P}_{LP}}^* = J_{\mathcal{D}}^*$. Anyway, assuming that the solutions of $\mathcal{D}$ and $\mathcal{P}_{LP}$, respectively λ^* and x_{LP}^*, are unique, they differ in at most β subsystems. Since x_{LP}^* is feasible with respect to the global constraints and provides a good objective value, it is expected that $x(\lambda^*)$ is "almost" feasible with a satisfactory cost. Note that x_{LP}^*

cannot be computed directly because there is no explicit description of $\text{conv}(V_p)$.

Therefore, the proposed method consists in contracting the resource array b through a virtual price $\rho \in \mathbb{R}^\beta$. We get the following modified version of $\mathcal{P}$:

$$\min_{v_1, \ldots, v_m} \sum_{p=1}^{m} c_p^T v_p$$
$$\text{subject to: } \sum_{p=1}^{m} A_p v_p \leq b - \rho \qquad (\bar{\mathcal{P}})$$
$$v_p \in V_p, \quad p = 1, \ldots, m,$$

where ρ is computed as:

$$\rho = \beta \cdot \max_p \left\{ \max_{v_p \in V_p} A_p v_p - \min_{v_p \in V_p} A_p v_p \right\}; \qquad (2.2)$$

here, the max operator has to be applied component-wise.

Accordingly, the problems $\bar{\mathcal{P}}_{LP}$ and $\bar{\mathcal{D}}$ are introduced, defined similarly to $\mathcal{P}_{LP}$ and $\mathcal{D}$ with the resource vector b replaced by $\bar{b} \doteq b - \rho$. The fundamental result is that, if $\bar{\mathcal{P}}_{LP}$ and $\bar{\mathcal{D}}$ have unique solutions, then any selection $x(\bar{\lambda}^*)$ is feasible for $\mathcal{P}$, where $\bar{\lambda}^*$ is the optimal solution of $\bar{\mathcal{D}}$. This means that for the problem to remain feasible after the contraction, the available resources must be sufficient.

In conclusion, it is established that the primal solutions obtained from the dual of $\bar{\mathcal{P}}$ are feasible for $\mathcal{P}$, with:

$$\lim_{m \to \infty} \frac{J_{\bar{\mathcal{P}}}^*}{J_{\mathcal{P}}^*} = 1, \qquad (2.3)$$

holding under the assumption that V_p are uniformly bounded and $J_{\mathcal{P}}^*$ grows linearly in terms of m.

2.1.1 Application example

Consider the following scenario: a fleet of PEVs are connected to a charging station, and the charging stations are linked to a central controller, which is responsible for managing the charging process. The controller has to decide the charging rate of each vehicle, taking into account their charging requirements and the power limits of the grid connection. The goal is to minimize the total charging cost satisfying the aggregated power constraint (i.e., the maximum value for the total power withdrawn from the grid), while ensuring that the charging process is completed according to the preferences of each driver, expressed in terms of final state of charge and charging time.

Typically, the problem is addressed with Model Predictive Control (MPC) techniques, which are well suited for the management of complex systems with multiple constraints.

Vujanic et al. (2014)[2] validates the presented work by adapting the decentralized procedure to the problem at hand. This is justified by the fact that, since the fleet is large, a centralized approach is not suitable due to the heavy computational burden. Moreover, the latter would also require the communication of a considerable amount of private information about the PEVs, which, on the contrary, is not required by the decentralized approach.

The results reported in the paper show that the proposed solution is, in general, considerably better than the centralized one. In particular, the higher the number of PEVs, the more appreciable the difference between the computation times of the two approaches becomes. A further notable result is the experimental demonstration of 2.3. However, the tentative solutions, as expected, do not immediately satisfy the global constraint, but there is a need to wait for a fair number of iterations.

2.2 Improved computation of the tightening array

Mainly relying on the work of Vujanic et al., Falsone et al. (2019)[4] proposes a decentralized iterative procedure allowing to compute in finite time a less conservative and well performing solution for $\mathcal{P}$.

As a matter of fact, in the previous part, the larger the value of ρ, the more the feasibility of the solution is hampered because of the reduction of the maximum available resources after the restriction. In order to recover feasibility of part of the solutions, the amount of resource tigthening (through ρ) is iteratively updated based on the candidate solutions explored up to that moment. This guarantees that the value of ρ is always lower or equal to the tightening array proposed in 2.2, which represents the worst-case scenario. Consequences include the possibility of applying the procedure to a larger class of problems with lower computational effort due to the finite-time convergence of the sub-gradient algorithm.

The main contribution provided in this work is summed up in the following law:

$$\rho(k) = \beta \cdot \max_p \left\{ \max_{\tau \leq k} A_p v_p(\tau) - \min_{\tau \leq k} A_p v_p(\tau) \right\},$$

which constitutes a monotonically non-decreasing sequence whose computation at instant k is based on the solutions $v_p(\tau)$ for $\tau \leq k$ of the inner problem $\mathcal{I}$, with $\lambda = \lambda(k)$, which depends on the restricted resource array $\bar{b}$ through the last available value of ρ, i.e., $\rho'(k-1)$.

Vujanic et al.'s approach would need, in principle, to solve β MILPs for an infinite number of iterations, un-

like Falsone et al.'s one, that only requires to solve β MILPs for a finite number of times.

2.2.1 Application example

In the final pages of Falsone et al. (2019)[4], the authors refer directly to the PEV recharging problem presented in Vujanic et al. (2014)[2] to make a comparison between the performance of the two decentralized approaches.

The first aspect where improvements are observed is the reduction in conservativity: the infinity-norm of the resource restriction array ρ of Falsone et al. is equal to 50% of that Vujanic et al.'s, regardless of the number of vehicles considered. Consequently, the authors report having simulated scenarios where, given stricter global constraints on the maximum aggregate power, a feasible solution was found only by using Falsone et al.'s procedure.

Moreover, in both solutions the optimality gap keeps decreasing with the number of PEVs according to 2.3, but the convergence is faster in Falsone et al.'s approach.

2.3 Semi-continouos decision variables and reference tracking

In Liberati et al. (2023)[5], the authors worked towards two goals, both introduced for the purpose of applying the previously introduced framework to a realistic PEV charging setting. The first one is to model the decision variables v_p possibly as semi-continuous variables (i.e., they are allowed to be either null or continuous); the second is to combine the coupling constraint with the goal of tracking a global reference.

The first objective is achieved with the introduction of auxiliary boolean variables $\delta_p \in \{0, 1\}$ that only influence the local domains V_p.

On the other hand, in order to achieve the second goal, in principle one has to minimize (component-wise) the distance between a linear functional of the decision variables $\mathcal{F}(v_1, \ldots, v_m)$ and the chosen reference $f^\star$. In light of this, the renewed cost function would be:

$$\min_{v_1, \ldots, v_m} \left\{ \mathbb{1}^T |\mathcal{F}(v_p) - f^\star| + \sum_{p=1}^m c_p^T v_p \right\},$$

where $\mathbb{1}$ is a vector of ones of appropriate dimension. The problem is that the last expression is nonlinear, but it can be linearized by introducing an auxiliary decision variable t that satisfies the following inequality constraints:

$$-t \leq \mathcal{F}(v_1, \ldots, v_m) - f^\star \leq t.$$

The final formulation of the primal problem is:

$$\min_{v_1,\ldots,v_m,t} \left\{ \mathbb{1}^T t + \sum_{p=1}^{m} c_p^T v_p \right\}$$

$$\text{subject to: } \sum_{p=1}^{m} A_p v_p \leq b, \tag{$\mathcal{P}$}$$

$$-t \leq \mathcal{F}(v_1,\ldots,v_m) - f^\star \leq t,$$

$$v_p \in V_p, \quad p = 1,\ldots,m.$$

2.3.1 Application example

Within the PEV charging problem (already mentioned in both Vujanic et al. (2014)[2] and Falsone et al. (2019)[4]), the innovations introduced by Liberati et al. (2023)[5] translate into modulating the charging and discharging power of each vehicle as a semi-continuous variable, modeling the realistic case for which the power can be either zero or take on values within an allowed range, and adding a desired aggregated power profile to be tracked globally.

The results show that the proposed approach is able to track the reference profile with a good approximation, while still respecting the constraints on the maximum aggregated power.

3 A novel approach

Although the underlying framework is almost unaltered, Manieri et al. (2023)[1] proposes a different procedure for computing, still in an iterative fashion, the resource restriction array ρ, allowing it both to increase and to decrease, differently from the previously seen studies.

3.1 Proposed procedure

Once $\rho \geq 0$ is fixed, let λ_ρ^* be the optimal solution of the restricted[1] version of $\mathcal{D}$, and let $v(\lambda_\rho^*)$ be the corresponding solution of $\mathcal{I}$ with $\lambda = \lambda_\rho^*$. Moreover, let v_ρ^{LP} be the solution of the restricted[1] version of $\mathcal{P}_{LP}$. Manipulating the global constraint through addition and subtraction of the same quantity and rearrangement, we get:

$$\sum_{p=1}^{m} A_p v_p(\lambda_\rho^*) = \sum_{p=1}^{m} A_p v_{p,\rho}^{LP} + \sum_{p=1}^{m} A_p(v_p(\lambda_\rho^*) - v_{p,\rho}^{LP}).$$

Since v_ρ^{LP} satisfies:

$$\sum_{p=1}^{m} A_p v_{p,\rho}^{LP} \leq b - \rho, \tag{3.1}$$

it is possible to derive the following upper bound:

$$\sum_{p=1}^{m} A_p v_p(\lambda_\rho^*) \leq b - \rho + \sum_{p=1}^{m} A_p(v_p(\lambda_\rho^*) - v_{p,\rho}^{LP}).$$

The coupling constraint of $\mathcal{P}$ is satisfied for sure if we choose:

$$b - \rho + \sum_{p=1}^{m} A_p(v_p(\lambda_\rho^*) - v_{p,\rho}^{LP}) \leq b,$$

resulting in the following sufficient condition on ρ:

$$\rho \geq \sum_{p=1}^{m} A_p(v_p(\lambda_\rho^*) - v_{p,\rho}^{LP}).$$

From this, the authors derive a less conservative update law for ρ:

$$\rho(k) = \left[\sum_{p=1}^{m} A_p(v_p(\lambda_{\rho(k-1)}^*) - v_{p,\rho(k-1)}^{LP}) \right]_+, \tag{3.2}$$

which is always positive in order to prevent any virtual broadening of the shared resources.

The authors also propose an algorithm (*Algorithm 2*[1]) that, given ρ as input, is able to return an estimate of v_ρ^{LP} and the value of $v(\lambda_\rho^*)$.

3.2, together with 3.1, implies that $\rho(k)$ satisfies:

$$\rho(k) \geq \sum_{p=1}^{m} A_p(v_p(\lambda_{\rho(k-1)}^*) - v_{p,\rho(k-1)}^{LP})$$

$$\geq \rho(k) + \sum_{p=1}^{m} A_p v_p(\lambda_{\rho(k-1)}^*) - b).$$

Let $w(h) = \sum_{p=1}^{m} A_p v_p(\lambda_{\rho(k-1)}^* - b)$ be the violation vector. From the previous inequality, it follows that:

$$\rho(k) \geq \rho(0) + \sum_{h=0}^{k-1} w(h).$$

Note that, because of the boundedness of the local constraint sets V_p, the sequence of $\rho(k)$ is bounded as well. This implies that each component of $w(h)$ must become negative sooner or later. Therefore, a feasible solution is certainly found when $\beta = 1$, while for $\beta > 1$ convergence is not guaranteed. Moreover, typically multiple mixed-integer solutions are admitted: this expectedly causes oscillations across iterations in the pursuit of a feasible solution.

---

[1] *Restricted* implies the use of the restricted resource array $\bar{b} = b - \rho$.

3.2 Application example

We implement the solution proposed by the Manieri et al. in a dedicated algorithm, which is tested in the aforementioned use case of the PEV recharge problem. The challenge involves determining the best charging schedule for a fleet of m vehicles that need to be charged overnight to achieve a specified battery state of charge by the following morning. This plan is organized over a period broken down into N time slots of duration T, corresponding to the prediction horizon of a MPC controller.

The upcoming discussion is mainly based on the scenario described in Liberati et al. (2023)[5].

3.2.1 Problem definition

The goal is to compute the optimal charging/discharging schedule for a set of m PEVs connected to the power grid.

Let $P = \begin{bmatrix} P_1^T & \dots & P_m^T \end{bmatrix}^T$, with $P_p \in \mathbb{R}^N$ being the difference between the charging and discharging power schedules of PEV $p = 1, \dots, m$ over the prediction horizon of duration N, given by:

$$P_p \doteq P_p^{ch} - P_p^{dis}, \quad \forall p. \tag{3.3}$$

At each time slot $t = 1, \dots, N$, each PEV can be exclusively charged or discharged within eligibility intervals:

$$\delta_p^{ch} P_p^{ch,min} \leq P_p^{ch} \leq \delta_p^{ch} P_p^{ch,max}, \quad \forall p, \tag{3.4}$$

$$\delta_p^{dis} P_p^{dis,min} \leq P_p^{dis} \leq \delta_p^{dis} P_p^{dis,max}, \quad \forall p, \tag{3.5}$$

$$\delta_p^{dis} + \delta_p^{ch} \leq 1, \quad \forall p. \tag{3.6}$$

Here, 3.6 guarantees the mutual exclusiveness of the charging/discharging processes, since $\delta_p^{ch}, \delta_p^{dis} \in \{0,1\}$ are boolean variables that determine the charging and discharging status of PEV p at time t. Meanwhile, 3.4 and 3.5 ensure that the power injected into or drawn from the battery is within the limits.

The aggregated power profile of the fleet is simply the sum of the individual PEV power profiles, and it has to comply with the maximum power limit constraint P^{max} imposed by the distribution system operator:

$$P^{agg} \doteq \sum_{p=1}^{m} P_p, \tag{3.7}$$

$$P^{agg} \leq P^{max}. \tag{3.8}$$

The state of charge of PEV p, namely x_p, evolves in time according to the following equation:

$$\begin{cases} x_p(t+1) = x_p(t) + T(\eta_p^{ch} P_p^{ch} - \frac{1}{\eta_p^{dis}} P_p^{dis}) \\ x_p(t_0) = (x_0)_p \end{cases} \quad \forall p, \tag{3.9}$$

where t_0 is the initial time slot, $(x_0)_p$ is the initial state of charge, and η_p^{ch} and η_p^{dis} are the charging and discharging efficiencies.

Some of the local constraints arise from the preferences of the PEV owners, who have the faculty to express the desired duration and final level of charge of the process; some other, instead, depend on the physical limitations of the battery, such as the minimum and maximum state of charge x_p^{min} and x_p^{max}:

$$x_p(F_p) = x_p^{ref}, \quad \forall p, \tag{3.10}$$

$$x_p^{min} \leq x_p \leq x_p^{max}, \quad \forall p. \tag{3.11}$$

Finally, the objective function to minimize is the following:

$$V \doteq \sum_{p=1}^{m} \xi_p^T P_p, \tag{3.12}$$

where small perturbation array ξ_p is introduced, weighting the power to satisfy Assumption 2.4 of Vujanic et al. (2014)[2].

3.2.2 Solution algorithms

In this subsection, we implement the algorithms proposed in Manieri et al. (2023)[1] by adapting them to the problem at hand.

Given a value of the tightening array ρ, Algorithm 2 is able to compute the corresponding solution couple $[P_\rho^{LP}, P(\lambda_\rho^*)]$. P_ρ^{LP} is the solution of the linear program corresponding to the restricted[1] version of the convexified problem $\mathcal{P}_{LP}$, while λ_ρ^* is the solution to the restricted[1] version of the dual problem $\mathcal{D}$. In particular, the Lagrange multiplier λ is updated through 2.1 with the solution of $\mathcal{I}$ found using the last available value of λ until convergence. Once λ is at convergence, the code constructs the sequence $\hat{P}(h)$, whose accumulation points are shown to be solutions of the restricted[1] version of $\mathcal{P}_{LP}$[2]. When $\hat{P}(h)$ is also at convergence, its most recent value is selected as the estimate of P_ρ^{LP}. On the other hand, once lambda has converged, each solution found in steps 4-6 is a valid $P(\lambda_\rho^*)$; among these, only the cheapest one is selected (steps 14-16).

Algorithm 1 allows to compute the optimal power schedule P^* for the PEVs. After initializing the resource restriction array ρ at zero, iteratively Algorithm 2 is employed in order to find $[P_{\rho(k)}^{LP}, P(\lambda_{\rho(k)}^*)]$,

---

[2]One can prove that $\lim h \to \infty \hat{P}(h) = P_\rho^{LP}$.

and if the latter is feasible and it is better than any previous one in terms of cost, it is stored, together with its cost, as the current optimal solution. Finally, the tightening array ρ is updated according to 3.2, and the process is repeated until a stopping condition is met.

Algorithm 1: Computation of P^*.

Output: P^*

1   $\rho(0) = 0$;

2   $V^* = \infty$;

3   **for** $k = 0, 1, 2, \ldots$ **do**

4     $[P_{\rho(k)}^{LP}, P(\lambda_{\rho(k)}^*)] = $ **Algorithm 2**$(\rho(k))$;

5     $P^{agg}(k) = \sum_{p=1}^{m} P_p(\lambda_{\rho(k)}^*)$;

6     $V(k) = \sum_{p=1}^{m} \xi_p^T P_p(\lambda_{\rho(k)}^*)$;

7   **if** $P^{agg}(k) \leq P^{max} \wedge V(k) \leq V^*$ **then**

8       $P^* = P(\lambda_{\rho(k)}^*)$;

9       $V^* = V(k)$;

10     $\rho(k+1) = \left[\sum_{p=1}^{m} P_p(\lambda_{\rho(k)}^*) - P_{p,\rho(k)}^{LP} \right]_+$;

11   **return** P^*;

Algorithm 2: Computation of $[P_\rho^{LP}, P(\lambda_\rho^*)]$.

Input: ρ

Output: $P_{\rho(k)}^{LP}$, $P(\lambda_{\rho(k)}^*)$

1   $\lambda(0) = 0$;

2   $h = 1$;

3   **for** $l = 1, 2, \ldots$ **do**

4   **for** $p = 1, \ldots, m$ **do**

5       $P_p(l) = \arg\min_{P_p} \left\{ \left(\xi_p^T + \lambda^T(l) \right) P_p \right\}$

6   s.t. 3.3-3.6, 3.9-3.11;

7     $\gamma(l) = \sum_{p=1}^{m} P_p(l) - P^{max} + \rho$;

8     $\lambda(l+1) = [\lambda(l) + \alpha(l)\gamma(l)]_+$;

9   **if** λ *is at convergence* **then**

10   **for** $p = 1, \ldots, m$ **do**

11         $\hat{P}_p(h) = \frac{\sum_{\tau=l-h}^{l} \alpha(\tau) P_p(\tau)}{\sum_{\tau=l-h}^{l} \alpha(\tau)}$;

12   **if** $\hat{P}(h)$ *is at convergence* **then**

13         $P_\rho^{LP} = \hat{P}(h)$;

14   **for** $p = 1, \ldots, m$ **do**

15           $P_p(\lambda_\rho^*) = \arg\min_{P_p} \xi_p^T P_p$

16   s.t. $P_p = P_p(\tau) : \tau \geq l - h$;

17   **return** $P_\rho^{LP}, P(\lambda_\rho^*)$;

18       $h++$;

3.2.3   Numerical results

First of all, it is necessary to recall the results of the numerical simulations carried out in Manieri et al. (2023)[1]. The authors, in fact, report having observed not only that the optimality gap between the solutions of $\mathcal{P}$ and $\bar{\mathcal{P}}$ tends to zero as the number of agents m increases according to 2.3 just as aforementioned for the cases of Vujanic et al. (2014)[2] and Falsone et al. (2019)[4], but that in this case it even sits some orders of magnitude below. This outstanding improvement clearly derives from a decrease in the level of conservativity of the approach, which however occurs at the cost of increased difficulty in identifying an admissible solution (as already mentioned, in fact, the guarantee of finding it persists only when $\beta = 1$).

Our simulations were performed adopting the parameters in Table 3.1 using *MATLAB R2024a*. The *Optimization Toolbox* and the *Global Optimization Toolbox* provided the utilities to solve the MILP problems (in particular, the *intlinprog* solver), while *Parallel Computing Toolbox* was exploited to take avantage of the modularity of the decentralized problem in order to speed up the computation employing multiple CPU cores, each effectively corrisponding to a cluster of PEVs.

Symbol	Value
m	500
$N = F_p$	24
T [min]	20
P^{max} [kW]	$\{400, 200, 600\}$
P_p^{max} [kW]	3.3
x_p^{max} [kWh]	$[8, 16]$
x_p^{min} [kWh]	1
$(x_0)_p$ [kWh]	$[0.2, 0.5] \cdot x_p^{max}$
$x_p(F_p)$ [kWh]	$[0.55, 0.8] \cdot x_p^{max}$
η_p^{ch}	$[0.925, 0.985]$
ξ_p	$[0, 0.3]$

Table 3.1: Simulation parameters.

The simulations were carried out setting the step-size α in Algorithm 2 to $\frac{0.01}{n+1}$. Moreover, since as just mentioned the procedure does not guarantee the feasibility of the solution, we introduced a tolerance of 5% on the aggregated power constraint violation, meaning that all solutions with a violation below this threshold were considered feasible.

Iteratively (and up to iteration 25, where we stopped), the percentage of violation of the coupling constraint descended under the set threshold, as shown in Figure 3.1, due to the evolution of the resource restriction array ρ (whose norm is plotted in Fig. 3.2).
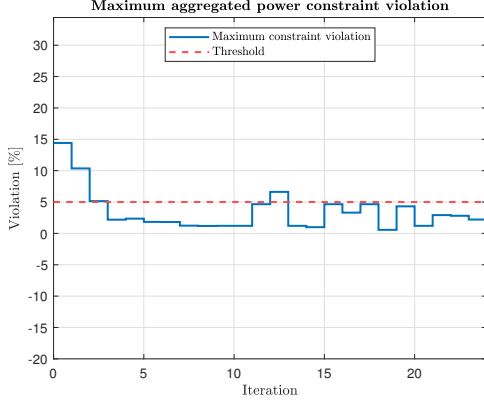
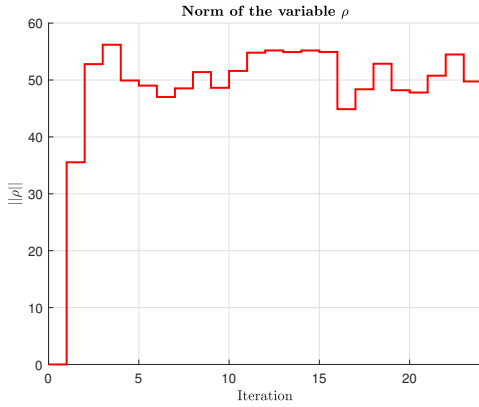Figure 3.1: Maximum percentage aggregated power constraint violation per iteration.

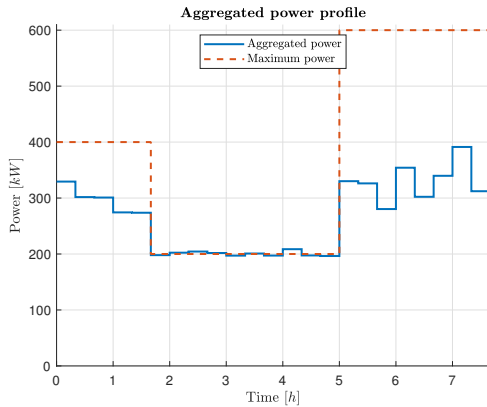Figure 3.2: Norm of the resource restriction array per iteration.

Figure 3.3: Maximum and actual aggregated power profiles.

The lowest-cost solution found among those examined is depicted in Fig. 3.3, together with the maximum aggregated power profile to be satisfied. In Fig. 3.4, the state of charge of a random batch of 10 PEVs is shown, highlighting the convergence to the desired final level of charge.
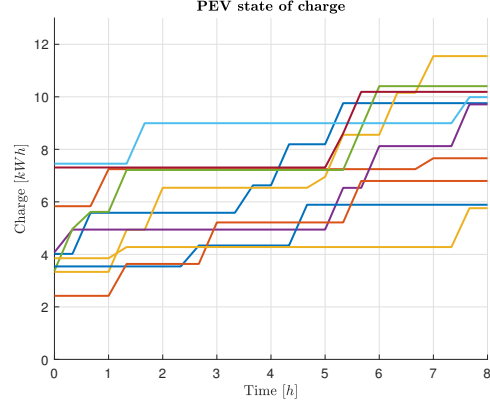

Figure 3.4: State of charge of 10 random PEVs.

Being computed in a decentralized way, as one would expect the solution at issue was found taking a much shorter amount of time than the MPC sampling time, namely about 19.4 seconds[3] ($<< 20$ minutes). However, this duration is considerably longer than that of the other procedures (for comparison, see Liberati et al (2023)[5], where, despite the presence of a greater number of constraints due to the reference tracking objective, the computation time is approximately 1 second).

Finally, we further tested the procedure by varying the quantity of PEVs of which the fleet is made up (of course, the maximum aggregated power constraint varied accordingly). The results obtained show that the maximum violation of the coupling constraint is greater the smaller the number of agents involved, and they are directly linked to what both the authors and the theory suggest in terms of optimality gap (related to, once again, 2.3).

m	Violation
20	40.97 %
50	14.66 %
100	10.26 %
200	8.65 %
500	3.46 %

Table 3.2: Mean global constraint violation on 25 runs for different numbers of PEVs.

---

[3]Mind that this duration is estimated and not directly measured, as that the number of cores of the computer CPU is less than m.

4 Conclusions

In this paper, we presented the state of the art of the decentralized approach to solve large-scale mixed-integer linear programming (MILP) problems, with a specific focus on the application to the control of plug-in electric vehicles (PEVs) charging in a smart grid environment.

Through the studies provided in Vujanic et al. (2014)[2], we have seen how the problem of PEV charging can be formulated as a MILP problem, and how a multi-agent decentralized approach can be used to find a solution satisfying global coupling constraints through the use of a resource restriction array. Vujanic et al.'s method has been further improved in Falsone et al. (2019)[4], which provides a less-conservative and better-performing iterative method to compute the restriction array, and therefore to find a solution. Furthermore, we have seen the strategy of Liberati et al. (2023)[5] to exploit the structure of a MILP problem and suggest a realistic modeling of the PEV recharge problem (introducing global reference tracking and semi-continuous variables).

Finally, the main focus of this paper has been on the work of Manieri et al. (2023)[1]. Here, the authors propose a novel minimally-conservative method to find a solution for the MILP problem, once again based on the idea of resource restriction, with the additional perk of allowing the value of the restriction array both to increase and to decrease. Although the procedure in issue promises great performance in theory, we have found that gaining an advantage from an application point of view is somewhat complicated and requires appropriate trade-offs. Certainly, in order to obtain even better results in this direction, there is a need for further investigation, especially in the case where the coupling constraints have large size, or when the number of vehicles is scarce.

References

[1] Lucrezia Manieri, Alessandro Falsone, and Maria Prandini. "A novel decentralized approach to large-scale multi-agent MILPs". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress, pp. 5919–5924. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2023.10.616`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896323009874`.

[2] Robin Vujanic et al. "A Decomposition Method for Large Scale MILPs, with Performance Guarantees and a Power System Application". In: *Automatica* 67 (Nov. 2014). DOI: `10.1016/j.automatica.2016.01.006`.

[3] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. DOI: `10.1214/aoms/1177729586`. URL: `https://doi.org/10.1214/aoms/1177729586`.

[4] Alessandro Falsone, Kostas Margellos, and Maria Prandini. "A decentralized approach to multi-agent MILPs: Finite-time feasibility and performance guarantees". In: *Automatica* 103 (2019), pp. 141–150. ISSN: 0005-1098. DOI: `https://doi.org/10.1016/j.automatica.2019.01.009`. URL: `https://www.sciencedirect.com/science/article/pii/S0005109819300093`.

[5] Francesco Liberati, Simone Barcellona, and Alessandro Di Giorgio. "Decentralized PEV Control Based on a Subgradient Method for Mixed-Integer Programming Problems". In: *2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I-CPS Europe)*. 2023, pp. 1–6. DOI: `10.1109/EEEIC/ICPSEurope57605.2023.10194623`.