Integrating Aggregated Electric Vehicle Flexibilities in Unit Commitment Models using Submodular Optimization

Hélène Arvis[1][2][3]* Olivier Beaude[1]* Nicolas Gast[2]§ Stéphane Gaubert[3]§ Bruno Gaujal[2]§
[1] EDF Lab Paris-Saclay, OSIRIS department
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG,
[3] Inria, and CMAP, École polytechnique, IP Paris, CNRS,
* `{firstname.name}@edf.fr`, § `{firstname.name}@inria.fr`

Keywords : *Electric Vehicles, Optimization, Unit Commitment, Submodular Optimization*

1 INTRODUCTION

1.1 Context

As the installed capacity for renewable energy sources increases around the world, so does the need for flexibility of the electric power system to absorb their fluctuations. In particular, these flexibilities, including Electric Vehicles (EVs), must be correctly modeled in long-term optimization problems to evaluate their contribution to the system. Unit Commitment (UC) problems are a focus of study in this context and consist in dispatching electricity to balance demand and production while respecting the constraints of a large number of heteregeneous productions units, generally including nuclear power plants (subject to specific dynamic constraints), thermal plants, hydroelectric plants, and other renewable energy sources.

Typically, the UC is a very large Mixed-Integer Linear Program (MILP), modeling dispatch decisions over hundreds of time steps with thousands of power plants in a country. A common approach to solving these problems is through aggregation, i.e., replacing a group of similar units by a class representative in order to reduce problem size and compute solutions in reasonable time. The clustering of similar units has been explored by Meus et al. [1], who demonstrate the benefits of aggregation in terms of tractability, though this leads to approximate solutions.

In this paper, we consider the specific case of EV charging profiles, in particular the dispatch of power through smart charging by a centralized operator. Aggregating a large number of flexibilities boils down to computing, or approximating, the Minkowski sum of a large number of polytopes, where each summand represents the flexibility set of an individual user or group of similar users. This Minkowski sum generally has exponentially many inequalities; moreover, calculating the Minkowski sum of a family of polytopes defined by their facets is generally NP-hard [2]. Therefore, techniques must be explored to avoid an explicit computation of this Minkowski sum. The traditional, “direct”, aggregation method, which consists in summing the characteristics of individual users so as to have a "super-user" that represents the whole population, generally leads to an aggregation error.

1.2 Contribution

We study a general UC model, including both power production units and a detailed model of EV flexibilities. We leverage the generalized polymatroid structure of the EV component of the UC problem to develop an exact algorithm, combining a cutting-plane approach with submodular optimization. We show in particular that, when the constraint sets of the power plants are convex, the UC problem can be solved to optimality in a time which scales as $N \log N$, where N is the number of EV user profiles (see Corollary 2). We also present a practical cutting-plane algorithm (Algorithm 1) and demonstrate its efficiency on a realistic case study with European grid data (European Resource Adequacy Assessment project [3]) and EV data taken from a survey of French driver behavior [4]. In particular, the algorithm scales well relative to the number of iterations and computed cutting planes.

1.3 Related work

The specific case of EV aggregation has already been studied, though not always in the context of a large UC problem. Mukhi et al. [5] observed that EV flexibility sets have the structure of generalized polymatroids, a fundamental class of polyhedra studied extensively by Frank and Tardos [6]. They are also special cases of the polyhedra associated to unimodular systems, studied in the broader setting of discrete convexity by Danilov and Koshevoy [7]. Independently (before the appearance of [5]) Koshevoy pointed out to us [8] that polyhedra associated to the discrete convexity theory of the root system A_n, i.e., generalized polymatroids, can be used to describe EV constraint sets, leading us also to a polymatroid approach. However, our work addresses a more general model: instead of considering a "pure" EV charging model which can be solved by a greedy algorithm [5], we consider the full UC problem, of which EVs are only a component, and we solve it by a cutting plane approach.

2 THE UC PROBLEM AND EV FLEXIBILITIES

2.1 Modeling the Unit Commitment problem

We consider a Unit Commitment problem on a discrete time horizon $\mathcal{T} = \{1, \dots, T\}$ with time steps of duration τ. Consider a set $\mathcal{M}$ of production units. Each element $m \in \mathcal{M}$ represents a single unit, which may be a nuclear or thermal (gas, coal, etc.) power plant, as well as a hydroelectric plant. The production of unit m is represented by a vector $\boldsymbol{z}^m = (z_t^m)_{t\in\mathcal{T}} \in \mathbb{R}^{\mathcal{T}} \in \mathcal{K}^m$, where z_t^m represents the production at time t and $\mathcal{K}^m$ is the unit's *constraint set.* The unitary production cost is denoted by c_t^m.

Variable (uncontrollable) renewable energy sources (solar, wind) are integrated via a residual demand $\boldsymbol{D} \in \mathbb{R}^{\mathcal{T}}$, obtained by subtracting the renewable production from the power consumption, at every instant t. Finally, we consider an aggregate of EV flexibilities with constraint set $\mathcal{P}$, which aggregates the behavior of a large heterogeneous population of users. We will elaborate on this set in the next section. The UC problem is of the form

$$\min_{\boldsymbol{z},\boldsymbol{p}} \quad \sum_{m\in\mathcal{M},t\in\mathcal{T}} c_t^m z_t^m \tag{1a}$$

$$\text{s.t.} \quad \sum_m z_t^m = D_t + p_t, \quad \forall t \in \mathcal{T} \tag{1b}$$

$$\boldsymbol{z}^m \in \mathcal{K}^m, \qquad \forall m \in \mathcal{M} \tag{1c}$$

$$\boldsymbol{p} \in \mathcal{P} \ , \tag{1d}$$

and aims to minimize the total production cost while keeping the supply-demand balance at every time step (1b) and respecting the constraints of each unit. The variable $\boldsymbol{p} \in \mathbb{R}^{\mathcal{T}}$ represents the charging power (positive or negative, in the case of Vehicle-to-Grid (V2G)) of the fleet of EVs.

2.2 Modeling EV flexibilities

We consider EV constraints over the whole time horizon. For an individual, this requires the knowledge of charging availability slots, maximal charging power, and energy need at the end of each slot. Consider an EV $n \in \mathcal{N}$ and its charging profile $\boldsymbol{p}^n$. At each time $t \in \mathcal{T}$, we have

$$\underline{p}_t^n \leqslant p_t^n \leqslant \overline{p}_t^n \ , \tag{2}$$

where $\overline{p}_t^n$ is equal to the maximal charging power available, typically based on charger limits, and $\underline{p}_t^n$ is equal to the maximal discharging power. If the EV is not plugged in at t, $\overline{p}_t^n = \underline{p}_t^n = 0$.

At each instant, the EV also has a state-of-charge (SoC) SoC_t^n, bounded by the battery capacity $\overline{\mathrm{SoC}}_t^n$ and the minimal energy required at time t, $\underline{\mathrm{SoC}}_t^n$, giving that

$$\underline{\mathrm{SoC}}_t^n \leqslant \mathrm{SoC}_t^n \leqslant \overline{\mathrm{SoC}}_t^n \ . \tag{3}$$

The minimal energy required at time t is typically the energy required by the user to complete his or her next trip. Define $\gamma_t^n \geqslant 0$ the energy consumed by driving during time step t.

Since we naturally have $\mathrm{SoC}_t^n = \mathrm{SoC}_{t-1}^n + \boldsymbol{p}_t^n - \gamma_t^n$, (3) becomes:

$$\underline{\mathrm{SoC}}_t^n - \mathrm{SoC}_0^n + \sum_{t'=1}^{t} \gamma_{t'}^n \leqslant \sum_{t'=1}^{t} p_{t'}^n \leqslant \overline{\mathrm{SoC}}_t^n - \mathrm{SoC}_0^n + \sum_{t'=1}^{t} \gamma_{t'}^n, \ \forall t \in \mathcal{T} \ . \tag{4}$$

In this case, the individual EV constraint set is defined by

$$\mathcal{P}^n = \left\{ \boldsymbol{p}^n \in \mathbb{R}^{\mathcal{T}} \middle| \boldsymbol{p}^n \text{ respects (2) and (4)} \right\} . \tag{5}$$

2.3 Aggregating EV constraints

The power profile set $\mathcal{P}$ of constraint (1d) arising from the flexible EVs is in fact the Minkowski sum of the individual power profile sets of the EVs: $\mathcal{P} = \sum_{n \in \mathcal{N}} \mathcal{P}^n$. In general, this set has a very rich face description whose complexity can grow quickly with N. This makes the UC (1) hard to solve. To cope with the complexity of $\mathcal{P}$, a classical method is to aggregate the constraint sets by using what we call a *naive* aggregation. The naive aggregation set $\hat{\mathcal{P}}$ is defined by summing the individual constraints:

$$\hat{\mathcal{P}} = \left\{ p \in \mathbb{R}^{\mathcal{T}} \middle| \begin{array}{l} \sum_{n \in \mathcal{N}} \underline{p}_t^n \leqslant p_t \leqslant \sum_{n \in \mathcal{N}} \overline{p}_t^n, \qquad \forall t \in \mathcal{T} \\ \sum_{n \in \mathcal{N}} \underline{s}_t^n \leqslant \sum_{t'=1}^{t} p_{t'} \leqslant \sum_{n \in \mathcal{N}} \overline{s}_t^n, \forall t \in \mathcal{T} \end{array} \right\}, \tag{6}$$

where $\underline{s}_t^n$, resp. $\overline{s}_t^n$, is the lower, resp. upper, bound of (4).

It is immediate that $\hat{\mathcal{P}}$ is an outer approximation of $\sum_{n \in \mathcal{N}} \mathcal{P}^n$. The inverse is not true, as taking the Minkowski sum creates "mixed" facets, not obtained by a naive summation; in fact, the naive polytope has at most $4T$ facets, corresponding to the $4T$ constraints.

3 GENERALIZED POLYMATROIDS

Recall that a set function $f : 2^{\mathcal{T}} \to \mathbb{R} \cup \{+\infty\}$ is **submodular** if

$$f(A) + f(B) \geqslant f(A \cap B) + f(A \cup B), \quad \forall A, B \subseteq \mathcal{T} \ , \tag{7}$$

and g is **supermodular** if $-g$ is submodular. Furthermore, considering a submodular function f and a supermodular function g such that $g(\emptyset) = f(\emptyset) = 0$, the pair (g, f) is said **paramodular** if

$$f(A) - g(B) \geqslant f(A \setminus B) - g(B \setminus A), \quad \forall A, B \subseteq \mathcal{T} \ . \tag{8}$$

From these functions, we can derive the notion of generalized polymatroids [6].

Definition 1. Consider a paramodular pair of functions (g, f). The polyhedron

$$Q(g, f) := \{x \in \mathbb{R}^{\mathcal{T}} \mid g(A) \leqslant x(A) \leqslant f(A), \ \forall A \subseteq \mathcal{T}\} \tag{9}$$

is a **generalized polymatroid**, or a **g-polymatroid**.

According to [6, Prop. 2.5], the border pair of a given g-polymatroid $Q \neq \emptyset$ is unique. In particular, we have $f(A) = \max_{x \in Q} x(A)$ and $g(A) = \min_{x \in Q} x(A)$ for every subset $A \subseteq \mathcal{T}$.

Regarding the modeling of EVs, it is shown in [8, 5] that EV flexibilities can be seen as generalized polymatroids, i.e., for each EV $n \in \mathcal{N}$, the set $\mathcal{P}^n$ is a g-polymatroid. As pointed out to us by Koshevoy [8], this follows from the laminar character of the family of constraints. This observation is essential as one of the main properties of g-polymatroids, established in [9, Thm. 14.2.15], also following from [7, Sec. 7], is that they are stable by Minkowski sum, i.e., given $(g^i, f^i),\ i \in [k]$, then $\sum_i Q(g^i, f^i) = Q(\sum_i g^i, \sum_i f^i)$. In the context of EVs, this gives a new description of exact aggregation.

To tackle the larger UC problem, we rely on base polyhedra.

Definition 2. Consider $\tilde{\mathcal{T}} = \mathcal{T} \cup \{T+1\}$ and a submodular function f such that $f(\tilde{\mathcal{T}})$ is finite. The **base polyhedron** with border function f is the set

$$\mathcal{B}(f) := \left\{ x \in \mathbb{R}^{\tilde{\mathcal{T}}} \,\middle|\, \begin{array}{l} x(\tilde{\mathcal{T}}) = f(\tilde{\mathcal{T}}) \\ x(A) \leqslant f(A),\ \forall A \subseteq \tilde{\mathcal{T}} \end{array} \right\} . \tag{10}$$

We can easily show that a base polyhedron is a g-polymatroid by taking $g(A) := f(\tilde{\mathcal{T}}) - f(\tilde{\mathcal{T}} \setminus A)$ [9]. More to the point, every g-polymatroid $Q(g, f)$ is the projection of a 0-base polyhedron (such that $f(\tilde{\mathcal{T}}) = 0$) [9, Thm. 14.2.5]. Therefore, optimizing over a g-polymatroid reduces to optimizing over the corresponding base polyhedron [9]. These polyhedra can easily be described as, given a paramodular pair (g, f), the submodular function defining the base polyhedron is given by

$$f^*(A) = \begin{cases} f(A) & \text{if } A \subseteq \mathcal{T} \\ -g(\mathcal{T} \setminus A) & \text{if } T+1 \in A \end{cases} . \tag{11}$$

4 SOLVING THE UC PROBLEM WITH EVs

4.1 Polynomial-time solvability of the convex UC problem

We will now place ourselves in the *convex* UC problem, in which we suppose that each constraint set $\mathcal{K}^m$ is given by a collection of linear inequalities, i.e., $\mathcal{K}^m = \{z^m \in \mathbb{R}^{\mathcal{T}} \mid F^m z^m \leqslant b^m\} \subset \mathbb{R}^{\mathcal{T}}$.

Our approach relies on the analysis of linear programs defined implicitly by separation oracles, which goes back to the work of Gröetschel, Lovász and Schrijver [10]. To obtain a tighter explicit bound, we use a recent improvement of their approach [11].

A key notion is that of a *polyhedral separation oracle* in the sense of [11] which, for a polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^d | G\boldsymbol{x} \leqslant \boldsymbol{h}\}$ and a vector $x \in \mathbb{R}^d$, asserts that $x \in P$ or returns a violated inequality $G_i\boldsymbol{x} > h_i$.

We work in the Turing model of computation, so we assume that all of the problem inputs are rational vectors. Without loss of generality, after multiplying by common denominators, we will even assume that the entries are integers. Then, we denote by B the maximal absolute value of all the integer coefficients appearing in the problem input, i.e., of the entries of the matrix F^m, of the vector $\boldsymbol{b}^m$, and of the bounds $\underline{s}_t^n$, $\overline{s}_t^n$, $\underline{p}_t^n$ and $\overline{p}_t^n$. The *dimension* (total number of scalar variables) is $d = (|\mathcal{M}| + 1)T$. The following result, along with its corollary below, is proved in our companion paper [12].

Theorem 1. *The convex UC problem integrating a collection of N flexible EVs can be solved in a number of polyhedral separation oracle queries bounded by $O(d^3 \log(dBN)))$, with d the dimension and B the maximal absolute value of integer coefficients of the problem input.*

In our context, a separation oracle can be implemented via Submodular Function Minimization (SFM). In fact, given a vector $\boldsymbol{x} = ((\boldsymbol{z}^m)_{m \in \mathcal{M}}, \boldsymbol{p})$, checking the validity of the inequalities $F^m \boldsymbol{z}^m \leqslant \boldsymbol{b}^m$ or of the global demand constraint is immediate ($\sum_m r^m + T$ scalar constraints that can be checked one by one, with r^m the number of inequalities for unit m). The hard part of the separation oracle consists in deciding whether the vector $\boldsymbol{p}$ belongs to the Minkowski sum $\sum_n \mathcal{P}^n$, which is decribed by 2^T g-polymatroid type inequalities. By what precedes, the membership problem for $\sum_{n \in \mathcal{N}} \mathcal{P}^n$ is equivalent to the membership problem from the base polyhedron $\mathcal{B}(f^*)$. By considering $\boldsymbol{x}^* = (x_1, x_2, \ldots, x_T, -\sum_{t \in \mathcal{T}} x_t) \in \mathbb{R}^{\mathcal{T}+1}$, retrieving a minimal subset A by SFM for $f^* - \boldsymbol{p}^*$ gives either an optimality certificate ($(f^* - \boldsymbol{p}^*)(A) \geqslant 0$) or a violated inequality ($\boldsymbol{p}^*(A) > f^*(A)$).

Corollary 2. *The convex UC problem with N flexible EVs can be solved in $N \log N \operatorname{poly}(\cdot)$ arithmetic operations, where $\operatorname{poly}(\cdot)$ denotes a polynomial in $T, d, \log B$ and $r = \sum_m r^m$ is the total number of constraints satisfied by production units.*

4.2 The practical cutting-plane algorithm

The cutting-plane procedure described by Algorithm 1 iteratively solves the UC problem (1) while enriching the constraint set at each iteration. We build a decreasing sequence of polytopes $\mathcal{P}^{(0)} \supset \mathcal{P}^{(1)} \supset \cdots \supset \mathcal{P}^{(\infty)}$ such that $\mathcal{P}^{(0)} = \hat{\mathcal{P}}$ the naive aggregation polytope (6), $\mathcal{P}^{(\infty)} := \cap_{k \in \mathbb{N}} \mathcal{P}^{(k)} = \sum_{n \in \mathcal{N}} \mathcal{P}^n$ the exact polytope, and each intermediate polytope has a richer constraint dictionary than the previous one.

Additionally, consider, for $k \in \mathbb{N}$, the enriched iterative UC problem

$$\begin{aligned}
\min_{\boldsymbol{z},\boldsymbol{p}} \quad & \sum_{m \in \mathcal{M}, t \in \mathcal{T}} c_t^m z_t^m && (UC_k\text{-a})\\
\text{s.t.} \quad & \text{(1b), (1c)} && (UC_k\text{-b})\\
& \boldsymbol{p} \in \mathcal{P}^{(k)} \;. && (UC_k\text{-c})
\end{aligned}$$

After solving problem (UC_k), we can apply the separation oracle to the EV solution $\boldsymbol{p}_k$: if the solution is an exact aggregation, the algorithm stops. Otherwise, the separation oracle returns a violated constraint which can be added to build the enriched constraint set $\mathcal{P}^{(k+1)}$. In practice, running the SFM algorithm allows for far more constraints to emerge as it must evaluate f^* repeatedly, and we use the whole collection of cuts obtained in this way, all at once, to enrich the constraint set, passing from $\mathcal{P}^{(k)}$ to $\mathcal{P}^{(k+1)}$. We show in [12] that Algorithm 1 terminates and returns an optimal solution.

Algorithm 1: A cutting-plane resolution of UC

```
$\mathcal{P}^{(0)} \leftarrow \hat{\mathcal{P}}$;
$k \leftarrow 0$;
for $k \geqslant 0$ do
    Solve $(UC_k)$ and retrieve optimal $\boldsymbol{p}_k$ ;
    Get $A \in \arg\min(f^* - \boldsymbol{p}_k^*)$ by SFM ;
    Get $\mathcal{C}_k$ polyhedron defined by facets calculated during SFM ;
    if $(f^* - \boldsymbol{p}_k^*)(A) \geqslant 0$ then
        $\boldsymbol{p}_k$ belongs to $\sum_{n \in \mathcal{N}} \mathcal{P}^n$: STOP ;
    else
        $\mathcal{P}^{(k+1)} = \mathcal{P}^{(k)} \cap \mathcal{C}_k$
    end
end
```

4.3 Numerical results of the cutting-plane algorithm

The performance of Algorithm 1 is demonstrated by a benchmark on a realistic instance of the European grid and French EV fleet. The European grid data is extracted from the European Resource Adequacy Assessment (ERAA) dataset [3], while the EV user profiles are selected at random from profiles in a French survey on driver behavior [4]. We consider that the total controllable population of 5.1 million EVs is evenly distributed into N different user profiles.

Using this data, we tackled a seven-country UC problem centered around the French market node. The linear programs were solved by *CPLEX* 22.1.2., with a Python 3.12 interface. The separation oracle was implemented in Julia 1.9.4, using the SFM package [13]. Simulations were run on the week, starting on Monday, of January 13, 2030 with the climatic conditions of 2003 and hourly time steps, over horizons $T \in \{24, 48, 96, 168\}$ and considering different numbers of EV user profiles $N \in \{2, 10, 50, 100\}$.

These simulations show Algorithm 1 to terminate in few iterations, as illustrated by Figure 1, most instances with fewer than 5. The number of cuts calculated by the separation oracle remained far smaller than the theoretical 2^{T+1} bound, as shown by Figure 2. The main point of improvement for this algorithm is the runtime, as illustrated by Figure 3. The multithreading implementation of the evaluation oracle is already faster than a sequential calculation; exploring other parallelization implementations could further improve the performance.

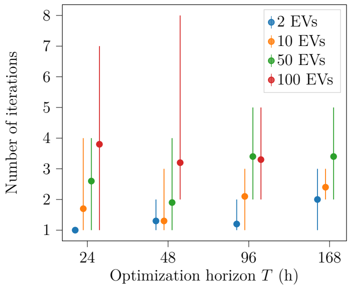

FIG. 1: Number of iterations performed by Algorithm 1. The error bar represents extreme values obtained, as well as the average value. Despite an increase with problem size, the iteration number stays low.

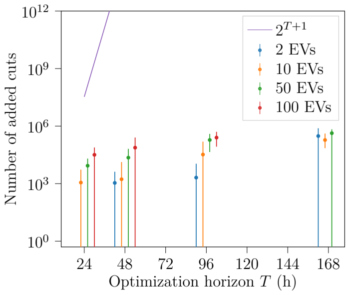

FIG. 2: Number of cuts added to UC. There is no point for $N = 2$ at $T = 24$ as every simulation converged in 1 iteration. All values are much smaller than the theoretical 2^{T+1} bound.

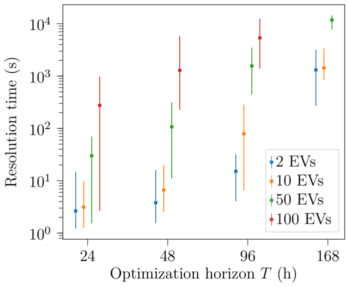

FIG. 3: Resolution time as a function of T and N. As expected, it increases with T and N.

References

[1] J. Meus, K. Poncelet, and E. Delarue, "Applicability of a Clustered Unit Commitment Model in Power System Modeling," *IEEE Transactions on Power Systems*, vol. 33, pp. 2195–2204, Mar. 2018.

[2] H. R. Tiwary, "On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes," *Discrete & Computational Geometry*, vol. 40, pp. 469–479, Oct. 2008.

[3] ENTSO-E, "ERAA - European Resource Adequacy Assessment ." `https://www.entsoe.eu/eraa/`.

[4] Ministère De L'Environnement (SDES), "Enquête Mobilité des Personnes - 2019," 2021.

[5] K. Mukhi, G. Loho, and A. Abate, "Exact Characterization of Aggregate Flexibility via Generalized Polymatroids." arXiv:2503.23458, Mar. 2025.

[6] A. Frank and É. Tardos, "Generalized polymatroids and submodular flows," *Mathematical Programming*, vol. 42, pp. 489–563, Apr. 1988.

[7] V. I. Danilov and G. A. Koshevoy, "Discrete convexity and unimodularity—I," *Advances in Mathematics*, vol. 189, pp. 301–324, Dec. 2004.

[8] G. Koshevoy, "Personnal communication on g-polymatroids," Feb. 2025.

[9] A. Frank, *Connections in combinatorial optimization*. No. 38 in Oxford lecture series in mathematics and its applications, Oxford: Oxford Univ. Press, 1. publ ed., 2011.

[10] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. No. v.2 in Algorithms and Combinatorics Ser, Springer, 1988.

[11] D. Dadush, L. A. Végh, and G. Zambelli, "On finding exact solutions of linear programs in the oracle model," in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2700–2722, SIAM, 2022.

[12] H. Arvis, O. Beaude, N. Gast, S. Gaubert, and B. Gaujal, "Integrating Aggregated Electric Vehicle Flexibilities in Unit Commitment Models using Submodular Optimization." preprint arXiv:2511.11191, Nov. 2025.

[13] E. Lock, "SubmodularMinimization.jl," 2025. `https://github.com/edwinlock/SubmodularMinimization.jl`.