Electric Vehicles Aggregator Optimization: a Fast and Solver-Free Solution Method

Robin Vujanic, Peyman Mohajerin Esfahani, Paul Goulart and Manfred Morari

Abstract— **The increased presence of Electric Vehicles (EVs) within electricity distribution systems introduces new challenges to their reliability, since uncoordinated charging of large numbers of EV can result in overload of distribution lines or transformers. In order to manage this difficulty, entities called EV aggregators are introduced whose task is to schedule charging of the EV fleet while ensuring that network constraints are respected. In this paper we propose a solution method for the type of constrained optimization problems such aggregators must solve. Our method is simple to implement and is guaranteed to produce *good* and *feasible* solutions, while performing only lightweight centralized computations which do not require the use of additional – and often expensive – constrained optimization solvers. We show that the quality of solutions produced by our method improves as the number of EVs to be controlled is increased. In addition, the computation times remain very short even for large problem instances entailing several thousands EVs.**

I. Introduction

A major new source of demand for future electricity distribution systems is the is the increasing use of hybrid or fully electric vehicles (EVs). Recent studies project EVs sales to constitute between 18% and 45% of total car sales in the US by 2020 [3]. This transition will cause a substantial change in the aggregated electric load profiles, with serious repercussions on the distribution networks, as highlighted by several studies [8], [16].

In order to ensure the ongoing reliable operation of electricity distribution networks, control mechanisms must be put in place to manage charging of electric vehicles. To achieve this using a centralized control architecture, a distribution system operator (DSO) would have to handle a potentially massive number of EVs. To avoid this difficulty, the control structure that is expected to prevail is a hierarchical one [7], [15]. In this setting so-called *aggregators* will provide the necessary interface, acting as virtual power plants from the perspective of the DSO or, generically, towards the higher levels of the control hierarchy, while managing the load fleet under their jurisdiction. In such a control architecture, the DSO will provide the limits on the available network resources that the aggregator can use to charge its fleet. These limits must be determined so as to ensure that network equipment is not damaged. In turn, the aggregator must decide on a charging schedule for its fleet that is compatible with these constraints. In this paper we are concerned with the computations to be carried out at the aggregator level. One option to fulfil the above needs is to use computational methods based on mathematical optimization. Optimization can incorporate the diverse charging requirements of each individual EV, while ensuring that, in aggregation, the network limits are satisfied. In such a framework it is furthermore possible to define an objective function that will determine the *best* charging schedule among all possible schedules that satisfy the aforementioned local (EV level) and global (network level) constraints. We describe the type of optimization programs an aggregator has to solve. Owing to fixed charge rate requirements, integer variable will be used in the model, leading to a mixed-integer optimization problem formulation; such models have already appeared in the literature [9], [14]. We will show that, despite the non-convex nature of this formulation, we can get approximate solutions at very low computational cost.

In particular, drawing from theoretical results obtained in [17], we provide a solution method for the aforementioned aggregator optimization problem. We propose a distributed method for computing near optimal solutions without recourse to any external constrained optimization solver (e.g. linear or quadratic program solvers). The solutions recovered are *guaranteed* to be feasible with respect to the local and global constraints. We also provide explicit bounds on their suboptimality, and show that the quality of the solutions produced improves as the size of the optimization programs considered (the number of controlled EV) increases, making it particularly useful for the larger instances comprising hundreds or thousands EVs. In our simulation results we show that the computation times on a normal PC, even for these larger instances, are very fast ($\leq$ 10 sec). Furthermore, depending on the availability of computational power on the EV, it is optionally possible to carry the bulk of the required computations distributedly.

The paper is organized as follows. In Section II we define the aggregator's control task and describe a prototypical use case. We then model this control task as an optimization problem. In Section III we report our proposed solution method, while in Section IV we describe how it is to be applied to the particular case of the EV charging coordination problem. We also present simulation results, showing the performance of our method. In Section V we make some concluding remarks.

R. Vujanic, P. Mohajerin Esfahani, P. Goulart and M. Morari are with the Department of Information Technology and Electrical Engineering, Automatic Control Laboratory, Swiss Federal Institute of Technology, ETH Zurich, Switzerland. Contact: vujanicr(at)control.ee.ethz.ch

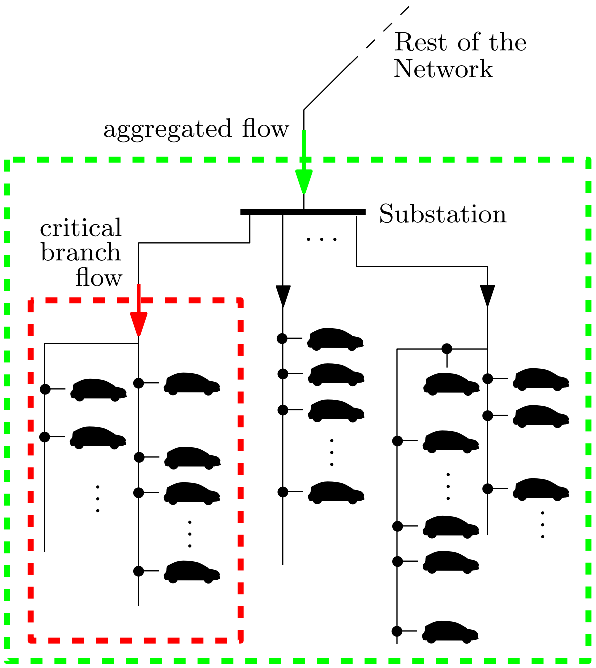


Fig. 1. Schematic representation of the distribution system.

II. Problem Formulation and Mathematical Modelling

In this section we first describe our control problem from an aggregator viewpoint. We then propose to tackle this control task using mathematical optimization, proposing an appropriate model to this end.

A. Scenario and Control Task Description

We take the perspective of a load aggregator responsible for the electric vechicles (EVs) connected to a particular section of a radial distribution network, as depicted in Figure 1. Most of the EV charging occurs overnight, but left uncontrolled it typically happens as soon as the EVs are connected to the network. Since EVs are relatively large loads this may cause load spikes which, in the worst case, can cause a partial blackout of the distribution system [8].

Our control task is to determine a charging schedule for each individual EV, such that the aggregated EV load is more evenly distributed over the entire night period. Additionally, the flows through a particular branch of the network under control must be kept within some limits. This may be for instance necessary to prevent an excessive voltage drop within the branch.

We assume that EV charging stations draw power at fixed rates, which is usually the case in practice [9]. Charging can be interrupted and resumed, but in order to avoid excessive switching, once charging starts it must continue for at least 20 minutes. This is a reasonable way of charging Lithium-Ion batteries, which are the most common in EVs, because they do not present memory effects and thus charging interruption does not cause any appreciable degradation [13]. Non-interruptible charging is not discussed in this paper as it is uncommon in practice. However, those applications for which this is necessary (e.g., Nickel-Cadmium batteries) can be readily incorporated in our proposed framework with an appropriate design of the local constraints.

We therefore split the overnight period into intervals of 20 minutes each, and assume that the aggregator has the authority to flag, for each individual EV, the available charging time slots.

The information to be communicated between each EV and the aggregator depends on the mode in which our proposed method is implemented. We elaborate on this in Section IV. For the moment, let us assume that when the EV is connected, it transmits all its charging requirements to the aggregator, i.e., the initial state of charge (SOC), a desired minimum final SOC and the time when the EV is planned to be disconnected. The initial SOC is directly measured by the EV, while the other two parameters can be set to some default values, so that the EV user does not need to manually enter this information every time.

For the sake of simplicity, we also assume that all the EVs are connected at the time when the charging schedule is established. This assumption can be easily relaxed by buffering newly connected EVs, and recomputing every 20 minutes a charging schedule with the new population information.

B. Optimization Problem Model

We begin by encoding the control task described in the previous subsection as an optimization problem. Generally, such a model will comprise a large number of subsystems, with a global objective and in the presence of some coupling constraints. Owing to the hierarchical structure discussed in Section I, the aggregator is responsible only for a limited number of network constraints. Thus, the number of coupling constraints is typically significantly smaller than the number of the subsystems under control.

The optimization problem model we propose in this paper is as follows:

- **Decision Variables.** The binary variables $u_i[k] \in \{0, 1\}$ are the decision variables representing the flag that the aggregator assigns to time-slot k for the electric vehicle i. When $u_i[k] = 1$, the EV is allowed to charge during the k-th time slot and, vice versa, when $u_i[k] = 0$ charging cannot occur.
- **Objective Function.** We model the requirement of evenly distributing load throughout the night as a reference tracking objective, in which the reference signal P^{ref} is chosen so as to achieve the desired "valley fill":

$$\underset{u_i}{\text{minimize}} \quad \sum_{k=0}^{N-1} \left| \sum_{i \in I} P_i u_i[k] - P^{\text{ref}}[k] \right|, \tag{1}$$

where P_i is the power consumption of the i-th EV when charging. Alternatively, one can write (1) as

$$\underset{u_i,r}{\text{minimize}} \quad \sum_{k=0}^{N-1} r[k], \tag{2}$$

together with the additional constraint

$$-r[k] \leq \sum_{i\in I} P_i u_i[k] - P^{\text{ref}}[k] \leq r[k] \quad k \in \mathbb{N}_{[0,N-1]} \tag{3}$$

- **Coupling Constraints.** In addition to (3), coupling constraints arise from the limits on the flows through the critical branch. We model these as hard constraints on the consumption of the electric vehicles $i \in I_{\text{cr}} \subseteq I$ belonging to that critical branch,

$$\sum_{i\in I_{\text{cr}}} P_i u_i[k] \leq P^{\max}[k] \qquad k \in \mathbb{N}_{[0,N-1]}. \tag{4}$$

- **Subsystems Model.** The subsystems in this application are the EVs batteries that are to be charged. We denote by $e_i[k]$ the charge level of the i-th battery. The initial state of charge is E_i^{init}, and we require that the state of charge reaches E_i^{ref} by $N_i \leq N$, the time when the vehicle is planned to be unplugged. The charging conversion efficiency is $\zeta_i < 1$, and the battery's maximum capacity is $E_i^{\max}$. Loss of charge when not in use is neglected, as it is typically a small quantity [13]. Subsystems are then modeled as follows:

$$\begin{aligned} & e_i[0] = E_i^{\text{init}} \\ & e_i[k+1] = e_i[k] + (P_i \Delta T \zeta_i) u_i[k] && k \in \mathbb{N}_{[0,N-1]} \\ & e_i[N_i] \geq E_i^{\text{ref}} \\ & e_i[k] \leq E_i^{\max} && k \in \mathbb{N}_{[0,N]} \\ & u_i \in \{0,1\}^N . \end{aligned}$$

The optimization problem that is to be solved at the aggregator level amounts then to

$$\left\{ \begin{array}{ll} \underset{r,e_i,u_i}{\text{minimize}} & \text{reference tracking error (2)} \\ \text{subject to} & \text{tracking error definition (3)} \\ & \text{critical branch flow constraint (4)} \\ & (e_i, u_i) \in X_i, \end{array} \right. \tag{6}$$

in which

$$X_i = \left\{ \begin{bmatrix} e_i \\ u_i \end{bmatrix} \in (\mathbb{R}\times\mathbb{Z})^N \;\middle|\; \begin{array}{l} e_i[k] = E_i^{\text{init}} + \sum_{t=0}^{k} B_i u_i[t] \\ e_i \leq E_i^{\max} \\ e_i[N_i] \geq E_i^{\text{ref}} \\ 0 \leq u_i \leq 1 \end{array} \right\} \tag{7}$$

where $B_i \doteq P_i \Delta T \zeta_i$.

III. Proposed Solution Method

The optimization problem model (6) expresses the typical structure we are interested in as aggregators. Generically, we want to solve problems of the form

$$\left\{ \begin{array}{lll} \underset{x}{\text{minimize}} & \sum_{i\in I} c_i^\top x_i & \\ \text{subject to} & \sum_{i\in I} H_i x_i \leq b & \\ & x_i \in X_i & \forall i \in I, \end{array} \right. \tag{$\mathcal{P}$}$$

in which a number of *subsystems*, described by the model X_i,

$$X_i = \left\{ x \in \mathbb{R}^{r_i} \times \mathbb{Z}^{z_i} \;\middle|\; A_i x \preceq d_i \right\},$$

is coupled through the global constraints defined by the matrix $H \doteq [H_1, \ldots, H_{|I|}]$ and the resource vector b. In the EV application these are for instance branch flow constraints (4), and the resource is the power line capacity. The subsystems are the EV batteries, and the local model X_i is the charging model (7).

Note that the local models X_i we are interested in may entail integer variables. In the EV aggregator application, these arise due to fixed rate charging. In other smart grid applications, integer variables can be used to model on/off devices such as thermostastically controlled loads [7].

In this Section we summarize the solution method for problems structured as $\mathcal{P}$ which is proposed in [17]. In Section IV we will apply this method to the EV aggregator problem (6).

The method is based on the lagrangian duality framework for mixed integer optimization problems. Roughly speaking, while duality is able to produce optimal solutions for convex programs, in the discrete case duality is weaker and is generally unable to provide even feasible solutions to the primal problem. However, it is known that the duality gap of programs structured as $\mathcal{P}$ decreases, in relative terms, as their size increases, i.e., when the number of subsystems X_i considered grows with respect to the number of coupling constraints [2], [5], [4]. This means that, as they become larger, these programs *resemble* more and more convex problems. This allows one to derive *good* and *feasible* solutions from their dual. We indicate how in the next subsection.

A. Duality for the Problem Structure of P

Duality for the specific problem structure $\mathcal{P}$ has been studied previously [5], [4]. In this framework we *dualize* the coupling constraints by relaxing them in the objective, leading to the *dual function*

$$d(\lambda) \quad \doteq \quad \min_{x\in X} \left(\sum_{i\in I} c_i^\top x_i + \lambda^\top (\sum_{i\in I} H_i x_i - b) \right). \tag{8}$$

It is known that, for any $\lambda \geq 0$, the dual function $d(\lambda)$ provides a lower bound to the optimal objective of $\mathcal{P}$. It is thus natural to ask for the best (i.e., greatest) lower bound duality can provide, and hence formulate the following *dual problem*

$$\left\{ \begin{array}{ll} \sup\limits_{\lambda} & -\lambda^\top b + \sum_{i\in I} \min\limits_{x_i\in X_i} \left(c_i^\top x_i + \lambda^\top H_i x_i \right) \\ \text{s.t.} & \lambda \succeq 0. \end{array} \right. \tag{$\mathcal{D}$}$$

The minimization in $\mathcal{D}$ is referred to as the *inner problem*. Note that, in contrast to the original coupled system $\mathcal{P}$, the inner problem is separated in $|I|$ decoupled, lower dimensional optimization problems. Therefore it is generally much easier to solve. For a given λ, we denote by $\mathcal{X}(\lambda)$ the set of minimizers to the inner problem, and when we say that we recover a primal solution from a dual one, we mean that we perform an arbitrary selection of an element from $\mathcal{X}(\lambda)$, and we denote it by $x(\lambda) \in \mathcal{X}(\lambda)$. These solutions are the central object of this paper. For generic mixed integer programs, even at an optimal dual solution $\lambda^\star$, a solution $x(\lambda^\star) \in \mathcal{X}(\lambda^\star)$ is typically suboptimal or even infeasible, see [6, Example 4.4]. However, inner solutions of programs structured as $\mathcal{P}$ do acquire some useful properties that can be exploited to devise a solution method.

To get this result, we first introduce the following *convexified* version of $\mathcal{P}$:

$$\left\{\begin{array}{lll} \underset{x}{\text{minimize}} & \sum\limits_{i\in I} c_i^\top x_i & \\ \text{subject to} & \sum\limits_{i\in I} H_i x_i \le b & \\ & x_i \in \operatorname{conv}(X_i) \qquad \forall i \in I. & \end{array}\right. \qquad (\mathcal{P}_{\mathrm{LP}})$$

Note that $\mathcal{P}_{\mathrm{LP}}$ is a linear program, but it *does not* coincide with the standard relaxation in which the integrality conditions in (7) are just dropped. It is fact generally tighter [10]. Notice further that we usually don't have access to this LP because we generally don't have an explicit description of the convex hulls $\operatorname{conv}(X_i)$.

One of the most important results in duality for mixed integer programs is that the optimal objective of the convexified problem $\mathcal{P}_{\mathrm{LP}}$ coincides with the optimal dual objective, i.e., $J^\star_{\mathcal{P}_{\mathrm{LP}}} = J^\star_{\mathcal{D}}$ [6], [10]. In [17] we show that the relation between $\mathcal{P}_{\mathrm{LP}}$ and $\mathcal{D}$ goes beyond their objectives, and also concerns their optimizers. We obtain this under the following assumption.

Assumption 1: The optimization problem $\mathcal{P}_{\mathrm{LP}}$ and $\mathcal{D}$ have, respectively, unique solutions $x^\star_{\mathrm{LP}}$ and $\lambda^\star$.

This assumption concerns linear programs (see [17, A.2] for the explicit LP version of $\mathcal{D}$). Uniqueness of solutions in this setting is discussed in [11]. The degenerate circumstance in which this assumption fails can always be avoided by introducing negigible perturbations to the cost c and the resource vector b [11].

In the next Theorem we assert that the solutions $x^\star_{\mathrm{LP}}$ and $x(\lambda^\star)$ differ in at most $\operatorname{rank}(H) \le m$ subproblem components, for any selection $x(\lambda^\star) \in \mathcal{X}(\lambda^\star)$, where $H \doteq [H_1, \dots, H_{|I|}]$.

Theorem 1: Under Assumption 1, for all $x(\lambda^\star) \in \mathcal{X}(\lambda^\star)$ there exists $I_1 \subseteq I$, with $|I_1| \ge |I| - \operatorname{rank}(H)$, such that $x_i(\lambda^\star) = (x^\star_{\mathrm{LP}})_i$ for all $i \in I_1$.

Proof: See [17], Appendix A.2. The key theoretical result used is the Shapley–Folkman–Starr theorem [4, Prop. 5.7.1]. ■

This result is interesting from a practical point of view. Note in fact that inner solutions $x(\lambda^\star)$ always satisfy the local technical constraints (including integrality), i.e., $x_i(\lambda^\star) \in X_i$ by construction. However, they usually violate the coupling constraints [6, Example 4.4]. In contrast, an optimizer to $\mathcal{P}_{\mathrm{LP}}$ always satisfies the coupling constraints by definition, but may violate the local integrality conditions in X_i, due to the convexification operation. The fact that these two optimizers are "close", as specified in Theorem 1, implies that we only need a relatively limited amount of compensation to make $x(\lambda^\star)$ feasible with respect to the coupling constraints. In the next subsection we propose a method to achieve this using a constraint tightening approach.

B. A Distributed Method based on Resource Contraction

Consider the following contracted version of $\mathcal{P}$:

$$\left\{\begin{array}{lll} \text{minimize} & \sum\limits_{i\in I} c_i^\top x_i & \\ \text{subject to} & \sum\limits_{i\in I} H_i x_i \le \bar{b} & \\ & x_i \in X_i \qquad \forall i \in I, & \end{array}\right. \qquad (\overline{\mathcal{P}})$$

in which the resource vector b has been contracted to $\bar{b} \doteq b - \rho$, and where the k-th entry of the contraction vector $\rho \in \mathbb{R}^m$ is given by

$$\rho^k = \operatorname{rank}(H) \cdot \max_{i\in I} \left(\max_{x_i \in X_i} H_i^k x_i - \min_{x_i \in X_i} H_i^k x_i \right), \tag{9}$$

where H_i^k is the k-th row of H_i. In the following Theorem we establish that any solution recovered from the dual of the contracted $\overline{\mathcal{P}}$ is feasible for the original problem $\mathcal{P}$. We also provide a performance bound for the solutions recovered. In order to obtain an explicit bound, however, we need to make the following assumption.

Assumption 2: Problem $\overline{\mathcal{P}}_{\mathrm{LP}}$ has a Slater point with increasing slack, i.e., there exists $\eta > 0$ and $\hat{x}_i \in \operatorname{conv}(X_i)$ for all $i \in I$ such that

$$\sum_{i\in I} H_i \hat{x}_i \le \bar{b} - \eta |I| \mathbf{1}. \tag{10}$$

Theorem 2: If Assumption 1 holds for the programs $\overline{\mathcal{P}}_{\mathrm{LP}}$ and $\overline{\mathcal{D}}$ associated to the contracted primal problem $\overline{\mathcal{P}}$, then any solution $x(\bar{\lambda}^\star) \in \mathcal{X}(\bar{\lambda}^\star)$ is feasible for the original, uncontracted, $\mathcal{P}$. Additionally, if Assumption 2 holds, then the recovered solution satisfies the following performance bound

$$\begin{aligned} &J_{\mathcal{P}}(x(\bar{\lambda}^\star)) - J^\star_{\mathcal{P}} \le \\ &\quad (\operatorname{rank}(H) + \tfrac{\|\rho\|_\infty}{\eta}) \cdot \max_{i\in I} \left(\max_{x_i \in X_i} c_i x_i - \min_{x_i \in X_i} c_i x_i \right). \end{aligned} \tag{11}$$

Proof: See [17], Appendix A.3 and A.4. ■

Observe that the bound given in (11) indicates that, if the sets X_i are uniformly bounded and $J^\star_{\mathcal{P}}$ grows linearly with increasing $|I|$, then

$$\frac{J(x(\bar{\lambda}^\star)) - J^\star_{\mathcal{P}}}{J^\star_{\mathcal{P}}} \to 0 \quad \text{as} \quad |I| \to \infty. \tag{12}$$

This shows one of the main attractive aspects of our proposed method: the quality of the solutions produced improves with increasing problem's size. This behaviour can be expected also without Assumption 2, but the analysis is more complicated, see [17, Theorem 3.9].

Note further that the constraint tightening approach always results in increased optimal dual multipliers, i.e., $\bar{\lambda}^\star \geq \lambda^\star$. When dual multipliers are seen as "shadow prices", we can interpret the contraction method as a way to establish an appropriate price-uplift, which ensures that the population participating in the control converges to an equilibrium which satisfies the global constraints despite the presence of local non-convexities.

IV. Application to EVs Aggregator

We now describe how to apply the method presented in Section III to the aggregator's optimization problem of Section II. We also highlight several important practicalities.

A. Implementation and Practical Remarks

To make the description easier to follow, we have divided it in a number of separate steps.

Step 1 (resource contraction). The number of coupling constraints in (6) is $3N$. However, one may observe that in view of the box constraints (3), $\mathrm{rank}(H)$ clearly drops to $2N$. Using an argument based on simplex tableaus, it is furthermore possible to show that this number does not need to be increased if other critical branches are added to the problem model, ensuring that the necessary contraction does not grow excessively; we omit the technical details here, and refer the interested readers to [17, App. A.2].

Moreover, note that for any solution $x(\lambda)$ the feasibility of the coupling constraints (3) is guaranteed by increasing the value r, ensuring that these constraint need not to be contracted. Constraints on the flows through the critical branch are true hard constraints whose feasibility is the main objective of the tightening procedure (9) proposed in this paper. This leads to

$$\sum_{i \in I_{\mathrm{cr}}} P_i u_i[k] \leq \bar{P}^{\max}[k] \doteq P^{\max}[k] - \rho, \tag{13}$$

where

$$\rho = 2N \cdot \max_{i \in I} P_i. \tag{14}$$

Remark 1: The interpretation of this result is that in the uncontracted solution $x(\lambda^\star)$ at most $2N$ EVs can be responsible for hard constraint violations which, in the worst case, occur all at the same time step.

Step 2 (perturbation). In our simulations, reported in the next subsection, we have observed that adding small perturbations to the cost vector, as in

$$\underset{u_i, r}{\text{minimize}} \quad \sum_{k=0}^{N-1} \left(r[k] + \sum_{i \in I} \delta_i[k](P_i u_i[k]) \right) \tag{15}$$

with small δ_i, significantly enhances dual convergence; we recommend to always implement this. Intuitively speaking, the reason why this helps is that the master problem intends to achieve coordination by issuing the "prices" λ, but each individual EV "sees" the same price without perturbation term. When the diversity of the subsystems X_i is limited, i.e., the EVs have similar dynamics, their reaction to price profiles is also similar. Consequently, as the price profile is adjusted, they all respond similarly, leading to an oscillatory behaviour which is solved only if the dual solution $\lambda^\star$ is found with very high precision. The perturbation term in (15) artificially changes the price received by each individual EV, neutralizing oscillations earlier and hence leading to substantially faster convergence rates when solving the dual.

Step 3 (setup and solve dual problem). Dualizing the coupling constraints in (6) leads to the dual function

$$\begin{aligned} d(\lambda, \mu) \quad = \quad & \sum_{i \in I} \min_{x_i \in X_i} P_i u_i^\top (\delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}_{\mathrm{cr}}(i)) \\ & + \min_r r^\top (\mathbf{1} - \lambda_1 - \lambda_2) \\ & - (\lambda_1 - \lambda_2)^\top P^{\mathrm{ref}} - \lambda_3^\top \bar{P}^{\max}, \end{aligned}$$

in which λ_1 and λ_2 are the dual multipliers associated with (3), while λ_3 is related to the dualization of (4), and

$$\mathbb{I}_{\mathrm{cr}}(i) = \begin{cases} 1 & \text{if } i \in I_{\mathrm{cr}} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Since the variable r is unbounded, the dual function remains bounded only for $\lambda \in \Lambda$, where

$$\Lambda = \left\{ (\lambda_1, \lambda_2, \lambda_3) \in \mathbb{R}^{3N} \,|\, \lambda_1 + \lambda_2 = \mathbf{1}, \lambda \geq 0 \right\}. \tag{17}$$

Given an initialization point $\lambda^{[1]}$, the corresponding dual problem, as defined in $\mathcal{D}$, can be solved by deploying the projected subgradient method [1], i.e.,

$$\lambda^{[n+1]} = \mathrm{P}_\Lambda \left(\lambda^{[n]} + s^{[n]} \cdot \gamma^{[n]} \right), \tag{18}$$

in which $\gamma^{[n]} = \sum_{i \in I} H_i x(\lambda^{[n]}) - b$ is a valid subgradient, and $s^{[n]}$ is the chosen stepsize rule. Several rules exist which will guarantee the convergence $\lambda^{[n]} \to \lambda^\star \in \Lambda^\star$, see [1, Thm. 2]. Let us remark that the iterative update rule (18) involves a projection operation onto Λ at each step, whose analytical solution is available. That is, letting $\lambda^+ \doteq \max(0, \lambda)$, where the maximum is component-wise, then

$$P_\Lambda(\lambda) = \begin{bmatrix} \frac{\lambda_1^+}{\lambda_1^+ + \lambda_2^+} \\ \frac{\lambda_2^+}{\lambda_1^+ + \lambda_2^+} \\ \lambda_3^+ \end{bmatrix}. \tag{19}$$

Step 4 (inner problem). At each iteration of the subgradient method, we need to compute a solution to the inner problem $x(\lambda^{[n]})$ in order to determine a valid subgradient $\gamma^{[n]}$. The inner problem consists of $|I|$ decoupled local optimization programs of the form

$$\min_{(u_i, e_i) \in X_i} P_i u_i^\top (\delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}_{\mathrm{cr}}(i)). \tag{20}$$

	Tracking Error θ (%)			Solve time† (sec)		
# PEVs	Min	Avg	Max	Min	Avg	Max
50	3.37	5.03	6.26	0.05	0.05	0.06
100	2.49	3.36	4.40	0.10	0.10	0.11
500	1.85	2.03	2.28	0.52	0.54	0.55
1000	0.29	0.56	0.72	1.05	1.08	1.09
2000	0.24	0.39	0.51	2.12	2.18	2.29
5000	0.21	0.27	0.30	4.34	4.42	4.50
10000	0.16	0.20	0.24	7.93	8.13	8.62

†sum of the time spent solving the dual iterations.

TABLE I

COMPUTATIONAL RESULTS.

The method proposed in this paper is most useful when these minimizations are simple. This is the case for the EV aggregator application. In fact, for any given dual multiplier $\lambda \in \Lambda$, the solution to (20) – the optimal local charging policy $u_i^{\star}(\lambda)$ – is **greedy**. As we illustrate in Appendix VI, due to this greedy subproblem structure, the required local computation amounts to a single vector sort, which is computationally extremely lightweight.

Note that these computations do not need to be executed by the aggregator itself. If computational power is available on the EVs to calculate solutions to (20), then the computational burden can be distributed. The advantage is that when the optimization problem is solved in this fashion, a higher degree of privacy results, as the EVs do not need to provide any local data to the aggregator (i.e., any of the parameters describing X_i), but only their planned consumption for a given price profile. The drawback is that such an iterative process will require more communication rounds to converge to a solution, instead of transmitting information only once.

B. Simulation Results

We have implemented this method on a set of instances of problem (6), which are generated according to the parameters in Table II. These reflect reasonable values for an aggregator managing a section of the distribution system containing $50'000$ households, and with a population of controlled EVs which we increase from $|I| = 50$ to $|I| = 10'000$. For each population size, we have generated and solved 10 instances. Computations are done on a Laptop with a 2.7GHz CPU and 4GB of RAM using MATLAB.

The numeric performance determined from these experiments is reported in Table I. Solve times are stable and well within 10 seconds even for the largest instances. If needed, these can be further reduced by exploiting parallelism, since the bulk of the computations consists in determining inner solutions, or by rewriting the code using a faster language (e.g., C). Tracking performance is measured by

$$\theta = \frac{\Delta T \| r(\lambda^{[\mathrm{end}]}) - r^{\star} \|_1}{\Delta T \| P^{\mathrm{ref}} \|} \cdot 100, \tag{21}$$

where $r^{\star}$ is the optimal solution, which we computed using Gurobi [12], and $r(\lambda^{[\mathrm{end}]})$ is the tracking error of the solution recovered from the last dual iterate $\lambda^{[\mathrm{end}]}$. This performance metric represents the achieved energy of the tracking error, relative to the optimal attainable one, divided by the total amount of energy drawn by the population.

Parameter	Value	Units
$\lvert I \rvert$	$50 - 10000$	PEVs
$\lvert I_{\mathrm{cr}} \rvert$	$1/3 \cdot \lvert I \rvert$	PEVs
P_i	$[3 - 5]$	kW
$E_i^{\max}$	$[8 - 16]$	kWh
E_i^{init}	$[0.2 - 0.5] \cdot E_i^{\max}$	kWh
E_i^{ref}	$[0.2 - 0.5] \cdot E_i^{\max}$	kWh
ζ_i	$[0.015 - 0.075]$	–
ΔT	20	min
N	24	–
δ_i	$[0 - 0.03]$	–
$P^{\max}$	$4 \cdot \frac{\sum_{i \in I_{\mathrm{cr}}} E_i^{\mathrm{ref}} - E_i^{\mathrm{init}}}{N \cdot \Delta T}$	kW

TABLE II

PARAMETERS USED IN THE SIMULATIONS. VALUES IN THE BRACKETS ARE SAMPLED FROM A UNIFORM DISTRIBUTION.

Figure 2 represents the typical performance plots for the primal solutions recovered through our contraction scheme, for an instance with $|I| = 5'000$ EVs. Figure 2(a) depicts the reference tracking performance at some intermediate and the final iteration. The reference signal is tracked well. In Figure 2(b) we can observe the flow limits through the critical branch (dotted line), and their tightened counterparts (dashed). At the final iteration, the solution recovered satisfies the line capacity constraints. Figure 2(c) shows the resulting "valley fill" achieved through the EVs control.

In Figure 3 we report the typical primal and dual convergence behaviour. Figure 3(a) depicts the constraint violation of the primal solutions recovered, with and without the perturbation terms discussed in Section IV. Convergence with the small additive perturbations (cf. Table II) is substantially enhanced.

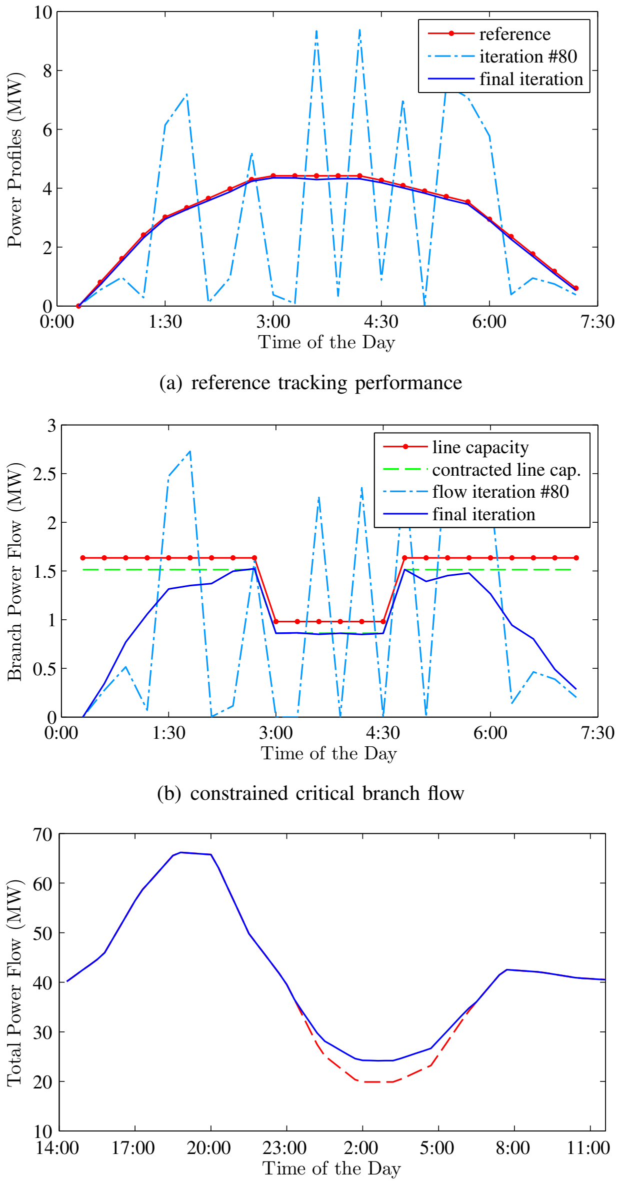
V. CONCLUSION

In this paper we have formulated the control task of an aggregator of Electric Vehicles as an optimization problem. Borrowing from recent theoretical results, we have provided a solution method that is simple to implement, has fast computation times and is guaranteed to provide feasible solutions attaining a certain minimum performance bound. Furthermore, the bulk of the computations required to obtain a solution can be distributed.

This paper is focused on the control of aggregated EVs. However, the proposed method is flexible enough to incorporate other on/off devices that are important for demand side management purposes, such thermostatically controlled loads.

ACKNOWLEDGEMENTS

The authors are thankful for the fruitful discussions with Sebastien Mariethoz regarding the application studied in this



(a) reference tracking performance

(b) constrained critical branch flow

(c) valley filling

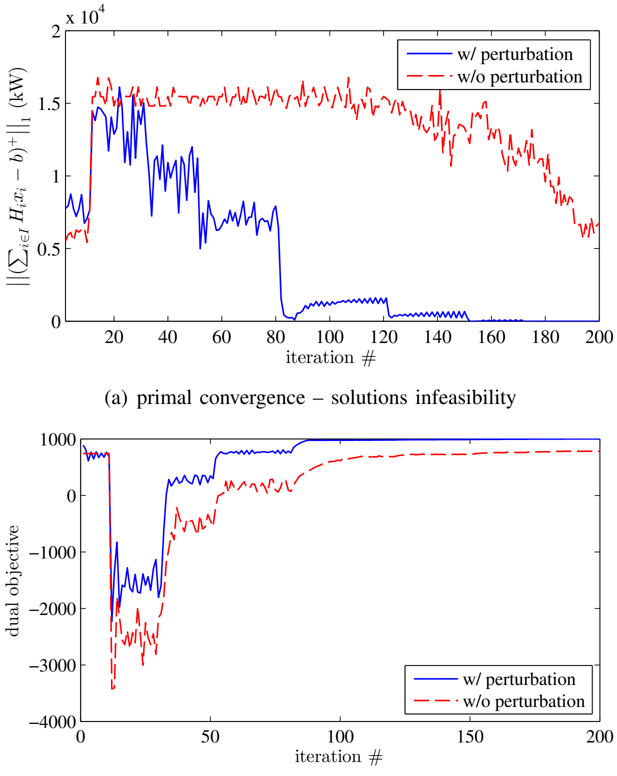
Fig. 2. Performance of the recovered primal solution.

paper.

VI. Appendix

The local problem to be solved is

$$\begin{array}{ll} \min & P_i \sum_{k=0}^{N-1} u_i[k]\psi_i[k] \\ \text{subject to} & e_i[k] = E_i^{\text{init}} + \sum_{t=0}^{k} (P_i \Delta T \zeta_i) u_i[k] \\ & e_i \leq E_i^{\max} \\ & e_i[N_i] \geq E_i^{\text{ref}} \\ & u_i \in \{0, 1\}\,, \end{array} \tag{22}$$



(a) primal convergence – solutions infeasibility

(b) dual convergence

Fig. 3. Primal and dual convergence.

in which we have defined the "price profile" seen by EV i as $\psi_i \doteq \delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}(I_{\text{cr}})$. We furthermore define the minimum and maximum number of charging steps as

$$C_i^{\min} \doteq \left\lceil \frac{E_i^{\text{ref}} - E_i^{\text{init}}}{P_i \Delta T \zeta_i} \right\rceil, \quad C_i^{\max} \doteq \left\lfloor \frac{E_i^{\max} - E_i^{\text{init}}}{P_i \Delta T \zeta_i} \right\rfloor. \tag{23}$$

We see that if $\psi_i \geq 0$, then the optimal control strategy is to activate charging for the least possible number of steps $C_i^{\min}$, and to do so during those times when ψ_i is lowest. For a price profile which contains possibly negative values, we can extend this as follows. We sort the entries of ψ_i in ascending order, storing it in $\psi_i^{\uparrow}$, and introduce

$$C_i^0 \doteq \max\left\{ k \,\middle|\, \psi_i^{\uparrow}[k] < 0 \right\},$$
$$C_i = \max\left\{ C_i^{\min}, \min\left\{ C_i^0(\psi_i^{\uparrow}), C_i^{\max} \right\} \right\}.$$

Then,

$$u_i^{\star}[k] = \left\{ \begin{array}{ll} 1 & \psi_i[k] \leq \psi_i^{\uparrow}[C_i] \\ 0 & \text{otherwise.} \end{array} \right. \tag{24}$$

The most expensive computation of this procedure is the vector sort used to determine $\psi_i^{\uparrow}$.

References

[1] Kurt M. Anstreicher and Laurence A. Wolsey. Two "well-known" properties of subgradient optimization. *Math. Program.*, 120(1):213–220, April 2009.

[2] J. P. Aubin and I. Ekeland. Estimates of the duality gap in nonconvex optimization. *Mathematics of Operations Research*, 1(3):225–245, January 1976.

[3] Thomas A Becker, Ikhlaq Sidhu, and Burghardt Tenderich. Electric vehicles in the united states: a new model with forecasts to 2030. *Center for Entrepreneurship and Technology, University of California, Berkeley*, (2009.1), 2009.

[4] Dimitri P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 1st edition, June 2009.

[5] Dimitri P. Bertsekas, G. Lauer, N. Sandell, and T. Posbergh. Optimal short-term scheduling of large-scale power systems. *IEEE Transactions on Automatic Control*, 28(1):1–11, January 1983.

[6] Dimitris Bertsimas and Robert Weismantel. *Optimization Over Integers*. Dynamic Ideas, June 2005.

[7] Duncan Callaway and Ian Hiskens. Achieving controllability of electric loads. *Proceedings of the IEEE*, 99(1):184–199, 2011.

[8] K. Clement-Nyns, E. Haesen, and J. Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, February 2010.

[9] A. Di Giorgio, F. Liberati, and S. Canale. Optimal electric vehicles to grid power control for active demand services in distribution grids. In *2012 20th Mediterranean Conference on Control Automation (MED)*, pages 1309–1315, 2012.

[10] A. M. Geoffrion. *Lagrangean relaxation for integer programming*, volume 2 of *Mathematical Programming Studies*. Springer Berlin Heidelberg, 1974.

[11] O.L. Mangasarian. Uniqueness of solution in linear programming. *Linear Algebra and its Applications*, 25:151–162, June 1979.

[12] Gurobi Optimization. Constrained optimization software, http://www.gurobi.com/.

[13] M.J. Riezenman. The search for better batteries. *IEEE Spectrum*, 32(5), May 1995.

[14] N. Ruiz, I. Cobelo, and J. Oyarzabal. A direct load control model for virtual power plant management. *IEEE Transactions on Power Systems*, 24(2):959–966, May 2009.

[15] Ramteen Sioshansi. Modeling the impacts of electricity tariffs on plug-in hybrid electric vehicle charging, costs, and emissions. *Operations Research*, 60(3):506–516, January 2012.

[16] J. Taylor, A. Maitra, M. Alexander, D. Brooks, and M. Duvall. Evaluations of plug-in electric vehicle distribution system impacts. In *2010 IEEE Power and Energy Society General Meeting*, pages 1–6, July 2010.

[17] R. Vujanic, P. Mohajerin Esfahani, P. Goulart, S. Mariethoz, and M. Morari. Vanishing duality gap in large scale mixed-integer optimization: a solution method with power system applications. `http://control.ee.ethz.ch/index.cgi?page=publications&action=details&id=4587`.