A Distributed Scheme for Fair EV Charging Under Transmission Constraints

Dzung T. Phan, Jinjun Xiong, Soumyadip Ghosh
IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA
{phandu, jinjun, ghoshs}@us.ibm.com

Abstract—We propose a distributed decision-making scheme to model the charging of a collection of Electric Vehicles (EV). Each charger individually determines its own charging schedule by iteratively transacting signals with a central authority. The EVs respond to signals from the central location, which sets its incentives to ensure safe operation of the local transmission and distribution grid. We study a model whose purpose is to flatten the load while maintaining the fairness of charging. Our model introduces three unique characteristics to this problem: capacity constraints on the distribution grid, fair rationing of energy supply available under the capacity constraint, and discrete choice of EV charger settings. We present a distributed scheme to solve the large-scale optimization model.

I. INTRODUCTION

The responsiveness and flexibility envisioned for the Smart Grid of the future provides unique advantages in integrating distributed and often intermittent energy resources at a scale that current grid technology finds hard to achieve. Among those promising technologies to realize future smart grids, electrification of the light-duty fleet is considered as the top priority for the U.S. Department of Energy as published in its recent Quadrennial Technology Review document [1].

The electrification of transportation holds out the promise of greater energy efficiency with a greener footprint, but substantial adoption will also impose significant challenges. In the long run, this has the potential of significantly altering the nature of electrical load by increasing mean consumption as well as shifting peak behaviors, thus necessitating large additional investments in generation resources and improving the transmission and distribution infrastructure. But in the interim, the distribution grid operators still have to support the increasing use of EVs in residential neighborhoods while ensuring that the grid is safely operated in the presence of existing (and newly emerging) capacity bottlenecks in the installed distribution network. This is especially the case for local transformers in residential neighborhoods, where the new load may force these to now operate near their safe operation capacity.

Promising smart grid technologies such as the loosely-coupled transactive control system studied in the Pacific Northwest Smart Grid Demonstration project [2] can help manage this challenge. The essence of such a system is to enable real-time communication and transactions between the distributed energy resources, the underlying idea being that this facilitates better decision making throughout the electricity grid by exchanging an incentive signal from a central location and its corresponding load response between nodes. This can help, for example, in creating EV charging profiles that fill in “valleys” in energy load at night so that the total load on transformers and sub-stations is flattened resulting in lower operational costs [3]. Furthermore, the energy stored in EVs may serve as an alternative to compensate for intermittency of renewable generation, regulation of voltage profiles etc. [4]. The key to the success of such a system is the proper design of a distributed decision-making scheme that lets the EVs optimize their charging schedules without violating the safe operating constraints of the distribution grid.

The literature has recently devoted much attention to the EV charging scheduling problem, that is a mixed integer nonlinear problem. Section II describes the centralized version of the charging problem. We present a heuristic to solve the centralized scheduling problem, that is a mixed integer nonlinear program. Section III proposes a heuristic to solve the centralized...
version, and Section IV presents the distributed version of this heuristic that greatly reduces the computational requirements on the central location to compute the broadcast incentive value. Numerical simulations are used to illustrate these results in Section V.

II. CENTRALIZED MODEL FORMULATION

A number N of EVs plug into the local grid to charge within a time horizon that stretches over a maximum of T intervals each of length $\Delta T$. The $n$-th vehicle plugs in at time $T^0_n$ and leaves at time $T^1_n$, and can charge at one of three charging rates $\{r^0_n, r^1_n, r^2_n\}$ in increasing order. The total power capacity of the central authority (e.g. transformer) is $C$ (in units of Watt). The base (non-EV) load faced by the authority at time $t$ is $P^0_t$, Watts. Define $(l^0_n, l^1_n)$ as binary variables to indicate if the $n$-th EV charges at slow or fast modes during the time period $t$. $E^n$ as total energy consumption (Joule) over $n$ time periods, $S^n$ as unsatisfied demand (Joule) over the non-EV load levels at each time interval. Constraint (2) enforces the capacity limitation of the central authority. Constraints (3)-(7) define the charge scheduling and unmet demand variables. Note that the latter set of constraints relate to each EV separately, and only the constraint (2) couples all the EVs together.

III. HEURISTIC FOR CENTRALIZED FORMULATION

MINLPs are in general very difficult to solve to optimality even for medium-sized instances. To the best of our knowledge, no existing MINLP solvers are able to globally solve this problem for practical instances in a reasonable time. Therefore, we propose first a heuristic based on the homotopy method to solve the centralized MINLP, and then describe a distributed version of this approach that is suitable for large-scale problems. The MINLP (1)-(7) can be compactly cast as follows

$$\begin{align*}
\min_{x,y} \quad & \varphi(x, y) = x^T P x + q^T x + y^T D y \\
\text{s.t.} \quad & A x + B y \leq b \\
& x \in \{0, 1\}^n, y \geq 0.
\end{align*}$$

where $P$ is an indefinite matrix, and $D$ is a strictly positive diagonal matrix. Here the variables $x$ are the charging variables $f^n_t, l^n_t$ and the continuous variables $y$ represent the unmet demands $S^n$. A standard relaxation of such MINLPs [8] is the nonconvex nonlinear program:

$$\begin{align*}
\min_{x,y,\alpha} \quad & \psi(x, y, \alpha) = \varphi(x, y) + \alpha \sum_i x_i (1 - x_i) \\
\text{s.t.} \quad & A x + B y \leq b \\
& 0 \leq x \leq 1, y \geq 0.
\end{align*}$$

For sufficiently large $\alpha > 0$, the penalty $\alpha \sum_i x_i (1 - x_i)$ ensures that these variables $x_i$ taken values in $\{0, 1\}$. Unfortunately, the (aNLP) often has a large number of local minimizers as $\alpha$ gets large, because the penalty drives the objective function to become concave or at least nonconvex. Any local search method fares poorly under these conditions, often returning local solutions of this relaxation that are of low quality to the MINLP. To deal with this nonconvexity of (aNLP), we make use of the homotopy method. The idea is that we start from a well-behaved convex problem, and gradually deform this “easy” problem into the original non-convex one. The solution from the previous problem is a starting point for the next deformation, and eventually the procedure ends with a solution of the original problem. The spirit of our approach is closely related to the method in [9]. Note that if a small enough $\alpha$ is used in (aNLP) such that $\alpha \leq \lambda_{\min}(P)$, where $\lambda_{\min}(P)$ denotes the smallest eigenvalue of matrix $P$, then $\psi(x, y, \alpha)$ is a strict convex function. Thus, increasing $\alpha$ iteratively from a small value reproduces the homotopy effect. The detailed description of our homotopy algorithm is as follows:

**HOMOTOPY METHOD FOR MINLP**

1. Initialize $x^0, y^0, \delta_1 > 0, \delta_2 > 1$ and $\alpha^0 < \lambda_{\min}(P)$.

2. For $k = 0, 1, \ldots$

   (a) Solve the program (aNLP) starting from $(x^k, y^k)$ to get $(x^{k+1}, y^{k+1})$.

   (b) Check a stopping criteria.

   (c) If $\alpha^k < 0$ then set $\alpha^{k+1} = \alpha^k + \delta_1$, else set $\alpha^{k+1} = \alpha^k \delta_2$.

**IV. DECOMPOSITION ALGORITHM FOR (aNLP)**

Constraints (3)-(7) are separable with respect to each EV, so we re-write the formulation (aNLP) in Section IV-A by making the constraint (2) separable with respect to each EV.
and obtain a natural distributed algorithm where each EV solves its own optimization subproblem.

A. Decomposition of Constraint (2)

Expanding (αNLP) with the original variables, we have

\[
\min_{\ell^t, l^T, S^N} \psi(f, l, S, \alpha^k) = \varphi(f^{\ell^t}, l^{T^t}, S^N) + \alpha^k \sum_{n=1}^{N} \sum_{t=1}^{T} (f^{\ell^t}_n (1 - f^{\ell^t}_n) + l^{T^t}_n (1 - l^{T^t}_n))
\]

s.t. \((2), (3), (4), (5), (7)\).

We will decompose this subproblem into \(N\) problems associated with each EV, which can be done by handling the sole coupling constraints (2). To do this, we use a penalty technique called the Powell-Hestenes-Rockafellar augmented Lagrangian method [10]. Let us define

\[
\ell(f, l, S, \mu, \rho; \alpha^k) = \psi(f, l, S, \alpha^k) + \sum_{t=1}^{T} \left[ (C - B_t - \sum_{n=1}^{N} (R^n f^n + r^n l^n) + \frac{\mu t}{\rho}) \right]^2,
\]

where \(\alpha^k = \max[0, \alpha]\). When applying the augmented Lagrangian method, we need to solve the following optimization problem at each iteration

\[
\min_{f, l, S} \ell(f, l, S, \mu, \rho; \alpha^k) \quad \text{(LSP)}
\]

s.t. \((3), (4), (5), (7), (8)\).

Note that \(\ell(f, l, S, \mu, \rho; \alpha^k)\) is continuously differentiable, but its Hessian is discontinuous. Hence only first-order methods can be used to solve (LSP). In the objective function, the decision variables \(f\) and \(l\) are coupled, not separated for each EV; however they are decoupled in the feasible set. We are going to approximate \(\ell(\cdot)\) by a sequence of functions, which are separable in \(f\) and \(l\). We observe that \(\ell(f, l, S, \mu, \rho; \alpha^k)\) can be expressed as follows

\[
\ell(f, l, S, \mu, \rho; \alpha^k) = h(f, l) + \sum_{n=1}^{N} \beta_n(S^n)^2,
\]

where \(h(f, l)\) is an indefinite quadratic function with respect to \((f, l)\). The program (LSP) falls into the general optimization model

\[
\min_{x \in \Omega} \Phi(x) = F(x) + G(x) \quad \text{(CF)}
\]

where \(\Omega\) is a convex set, \(F\) is continuously differentiable, and \(G\) is closed and convex but possibly nonsmooth on \(\Omega\).

B. First-order Method for Minimizing Composite Functions

First-order gradient-based methods for the problem (CF) have been extensively studied in the literature. An algorithm developed in [11] generalizes the gradient projection method. The differentiable term \(F\) is approximated around a point \(x\) by a linearized function and a proximal term

\[
Q_\gamma(x, z) = F(x) + \langle \nabla F(x), z - x \rangle + \frac{\gamma}{2} ||z - x||^2 + G(z),
\]

where \(\gamma\) is a step-size parameter. A sequence of such problems are solved with step-sizes \(\gamma\). Given any \(x^k\), the next iterate is uniquely defined by

\[
x^{k+1} = \arg \min_{x \in \Omega} Q_{\gamma, \epsilon}(x, z)
\]

\[
= \arg \min_{x \in \Omega} \frac{\gamma}{2} ||z - (x^k - \frac{1}{\gamma^2} \nabla F(x^k))||^2 + G(z).
\]

Both global convergence and the convergence rate of gradient-based methods greatly depend on the selection of the step-size \(\gamma^k\). First-order methods have been widely noticed to be very slow to converge, exhibiting a particularly zigzag behavior for ill-conditioned problems. A common recommendation in such cases is to include second-order information of the objective in the algorithm as much as possible. Barzilai and Borwein [12] suggest to choose step-sizes \(\gamma^k\) that mimic the second-order Newton step-size property that the Hessian \(\nabla^2 F(x^k)\) is approximated well by \(\gamma^k I\), where \(I\) is the identity matrix. This can be done approximately by imposing a quasi-Newton property on \(\gamma^k I\):

\[
\gamma^k_{BB} = \arg \min_{\gamma \in [0, \infty)} \| s^{k-1} - r^{k-1} \|_2^2 = \frac{(s^{k-1})^T r^{k-1}}{(r^{k-1})^T r^{k-1}}, \quad (9)
\]

where \(s^{k-1} = x^k - x^{k-1}\) and \(r^{k-1} = \nabla F(x^k) - \nabla F(x^{k-1})\). That is, we require that \(\gamma s^{k-1} = r^{k-1}\) in the least-squares sense, which mimics the secant equation for the Hessian \(\nabla^2 F(x^k) s^{k-1} \approx r^{k-1}\).

**Algorithm 1: Solving (CF)**

Given \(\eta > 1\), \(\sigma \in (0, 1), [\gamma_{\min}, \gamma_{\max}] \subset (0, \infty)\), and initial point \(x^1\).

Set \(k = 1\).

(a) Let \(\gamma_0 = \min(\gamma_{\max}, \max(\gamma_{\min}, \gamma_{BB}))\).

(b) Set \(\gamma = \eta/\gamma_0\) where \(j \geq 0\) is the smallest integer such that

\[
\Phi(x^{k+1}) \leq \phi_k^B - \sigma \|x^{k+1} - x^k\|^2
\]

where \(x^{k+1} = \min_{x \in \Omega} Q_{\gamma}(x, z)\).

(c) Set \(k = k + 1\) and go to (a).

To ensure the global convergence of the algorithm, some form of globalized monotonicity in the iterates is imposed. The general scheme maintains a reference function value \(\phi_k^B\), throughout. Every step \(k\) starts with the BB-stepsize but possibly scales \(\gamma_{BB}\) up to ensure that every iteration uses a step-size \(\gamma_k\) so that the objective function value reduces by a certain amount related to the reference function value \(\phi_k^B\). The first-order algorithm for solving (CF) using a reference function value \(\phi_k^B\) in the backtracking linesearch step is described in Algorithm 1.

A well-known reference function value is proposed by Grippo, Lampariello, and Lucidi (GLL) [13], which remembers the largest function value seen in the last \(M\) steps:

\[
\phi_k^{R, \text{GLL}} = \max \{ \Phi(x^{k-j}) : 0 \leq j < \min(k, M) \}.
\]

Wright et. al [14] use the GLL globalization technique in their SpaRSA algorithm. In [15], we study the convergence
rate of the SpaRSA and give an improved version of the algorithm based on a cyclic version of the BB iteration and an adaptive line search. In this paper, our approach to solving for problem (CF) is to generalize the strategies given in [15] to the constrained case. The adaptive reference function \( \Phi^R \) is given as follows:

(i) \( \Phi^R_k = \Phi(x^k) \),
(ii) \( \Phi(x^k) \leq \Phi^R_k \leq \max\{\Phi^R_{k-1}, \Phi^R_{GLL} \} \) for each \( k > 1 \),
(iii) \( \Phi^R_k \leq \Phi^R_{GLL} \) infinitely often.

Getting back to the decomposable form of the charger scheduling formulation (LSP), Algorithm 1 comes into play. For a given \((f^k, l^k)\), we approximate the \( h(f, l) \) term by the linear term of its Taylor approximation and a proximal regularization, where \( \gamma^k > 0 \) is a suitable parameter. Our approach generates a sequence \( \{f^k, l^k, S^k\}_{k=1,2,\ldots} \) by solving the following separable subproblems

\[
\min_{f^k, l^k, S^k} h(f^k, l^k) + \nabla h(f^k, l^k) \left[ f - f^k; 1 - l^k \right] + \frac{1}{2} \left\| f - f^k; 1 - l^k \right\|^2 + \sum_{n=1}^{N} \beta_n (S^n)^2
\]

s.t. (3), (4), (5), (7), and (8).

Equivalently, it amounts to solving for \( N \) separate subproblems associated with the \( n \)th EV:

\[
\min_{f^n, l^n, S^n} h_{n^\star, l^n^\star}(f^n, l^n) \left[ f^n - f^n_{\star}; 1 - l^n_{\star} \right] + \frac{1}{2} \left\| f^n - f^n_{\star}; 1 - l^n_{\star} \right\|^2 + \beta_n (S^n)^2
\]

s.t. \( \sum_{t=1}^{T} (R^n f^n_t + r^n l^n_t) \Delta T + S^n \geq E^n, \quad f^n_t + l^n_t \leq 1, \quad (T^n_t \leq t \leq T^n_{T}) \), \( f^n_t + l^n_t = 0, \quad (t < T^n_t \text{ or } T^n_T < t) \), \( 0 \leq f^n_t, l^n_t, \quad (t = 1, \ldots, T), \quad S^n \geq 0 \).

The main computational cost in applying Algorithm 1 is borne by this subproblem (10). In Subsection IV-C, we propose an efficient algorithm to solve the subproblem in linearithmic time.

C. Solving Each EV’s Decomposition Subproblem

The subproblem (10) associated with each EV is an instance of a knapsack-type problem of the following form:

\[
\min_{x, y} \in \mathbb{R}^n \times \mathbb{R}^n \left\| x - x_0 \right\|^2 + \left\| y - y_0 \right\|^2 + \beta \left\| z - z_0 \right\|^2
\]

s.t. \( d \leq a^\top x + b^\top y + c^\top z, \quad (KS) \)

\( x_i + y_i \leq 1, \quad i = 1, \ldots, n \),
\( x, y, z \geq 0 \),

where parameters \( a, b, c > 0, x_0 \in \mathbb{R}^n, y_0 \in \mathbb{R}^n, z_0 \in \mathbb{R}^n, \beta > 0 \) and \( d \) is a scalar. Without loss of generality, we assume \( a_i > b_i \) for all \( i \). The problem of minimizing a positive diagonal quadratic over the intersection of a singly linear constraint and a box has been of interest to the literature [16], [17].

For a given \( \lambda \geq 0 \), we define a new optimization problem by incorporating the complicating constraint in (KS) into the objective function

\[
\min_{x, y, z \geq 0} \left\| x - x_0 \right\|^2 + \left\| y - y_0 \right\|^2 + \beta \left\| z - z_0 \right\|^2 + \frac{2\lambda}{d - a^\top x - b^\top y - c^\top z}
\]

(11)

s.t. \( x_i + y_i \leq 1, \quad i = 1, \ldots, n \).

The following theorem characterizes the relationship between the optimal solutions of (KS) and (11); furthermore it motivates us an approach to solve for (KS).

Theorem 1: \( x, y, z \) is an optimal solution of (KS) if and only if there exists \( \bar{\lambda} \geq 0 \) such that \( x, y, \bar{z} \) is the solution of (11) associated with \( \bar{\lambda} \) and

(i) either \( \bar{\lambda} = 0 \) and \( d - a^\top \bar{x} - b^\top \bar{y} - c^\top \bar{z} \leq 0 \)
(ii) or \( \bar{\lambda} > 0 \) and \( d - a^\top \bar{x} - b^\top \bar{y} - c^\top \bar{z} = 0 \).

Proof: Notice that from conditions (i) and (ii) we can imply that

\[
d - a^\top \bar{x} - b^\top \bar{y} - c^\top \bar{z} \leq 0
\]

\[
\lambda (d - a^\top \bar{x} - b^\top \bar{y} - c^\top \bar{z}) = 0, \quad \bar{\lambda} \geq 0 .
\]

The remainder of the proof is just a straightforward application of the KKT conditions for the convex optimization problems (KS) and (11).

\[\text{ALGORITHM 2: Compute } (x(\lambda), y(\lambda)) \text{ for } \lambda \geq 0\]

\[\text{input: } \lambda\]

\[\text{output: } x(\lambda), y(\lambda)\]

\[\text{for } i = 1, \ldots, n \text{ do}
\]

\[x_i(\lambda) \leftarrow x_{0i} + \lambda a_i, \quad y_i(\lambda) \leftarrow y_{0i} + \lambda b_i\]

\[\text{if } x_i(\lambda) \leq 0 \text{ then }
\]

\[x_i(\lambda) \leftarrow 0, \quad y_i(\lambda) \leftarrow \max(0, \min(y_{0i}, 1))\]

\[\text{else if } y_i(\lambda) \leq 0 \text{ then }
\]

\[x_i(\lambda) \leftarrow \max(0, \min(x_{0i}, 1)), \quad y_i(\lambda) \leftarrow 0\]

\[\text{else if } x_i(\lambda) + y_i(\lambda) \leq 1 \text{ then }
\]

\[x_i(\lambda) \leftarrow x_i(\lambda), \quad y_i(\lambda) \leftarrow y_i(\lambda)\]

\[\text{else}
\]

\[x_i(\lambda) \leftarrow \max(0, \min(\frac{1 + x_i(\lambda) - y_i(\lambda)}{2}, 1)), \quad y_i(\lambda) \leftarrow 1 - x_i(\lambda)\]

\]
strict convexity, therefore, the function \( g(\lambda) \) is well-defined for any \( \lambda \geq 0 \). More precisely, we have
\[
z_i(\lambda) = \max \left\{ \frac{\beta z_{0i} + \lambda c_i}{\beta}, 0 \right\} \quad \text{for } i = 1, \ldots, m.
\]

In Algorithm 2, we describe how to evaluate \( x(\lambda), y(\lambda) \).

**Theorem 2:** The following hold.

(i) There exists a constant \( C > 0 \) such that \( g(\lambda) \leq 0 \) for any \( \lambda \geq C \).

(ii) \( g(\lambda) \) is a monotonically piecewise linear decreasing function when \( \lambda \geq 0 \).

(iii) The optimal solution of (KS) can be found by a binary search algorithm with \( O(\log(n)) \) iterations requiring order \( O(n \log(n)) \) operations for a given \( m \).

**Proof:** (i) Assume \( i_0 = \arg \max_i \{ c_i : i = 1, \ldots, m \} \).

For any \( \lambda \geq C = \max \{0, \frac{\beta d}{c_i^0} - \frac{\beta z_{0i}^0}{c_i^0}, 0\} \), we have
\[
g(\lambda) \leq d - c^T z(\lambda) \leq d - c_{i_0} z_{i_0}(\lambda)
\]
\[
\leq d - c_{i_0} \max \left\{ \frac{\beta z_{0i_0} + \lambda c_{i_0}}{\beta}, 0 \right\}
\]
\[
= d - c_{i_0} \frac{\beta z_{0i_0} + \lambda c_{i_0}}{\beta} \leq 0.
\]

(ii) The result easily follows because of the above formulas for \( (x(\lambda), y(\lambda), z(\lambda)) \) and the fact \( a, b, c > 0 \).

(iii) For any \( i \), from the formulas for \( (x(\lambda), y(\lambda), z(\lambda)) \), we can conclude that the piecewise linear function \( a_i x_i(\lambda) + b_i y_i(\lambda) \) has at most seven kinks: \( \max \{0, -\frac{z_{0i}}{a_i}\}, \max \{0, -\frac{1-x_{0i}}{a_i}\}, \max \{0, -\frac{1-x_{0i}+z_{0i}}{a_i}\}, \max \{0, -\frac{1-x_{0i}-z_{0i}}{a_i}\}, \max \{0, -\frac{1-x_{0i}+z_{0i}}{b_i}\}, \max \{0, -\frac{1-x_{0i}+z_{0i}}{b_i}\}; \) and \( c_i z_i(\lambda) \) has the kink at \( -\frac{a_i}{b_i} \); if \( z_{i_0} < 0 \). Hence \( g(\lambda) \) has up to \( 7n + m \) kinks, at which \( g \) is nondifferentiable, and is linear between two adjacent kinks. Note that the median of \( n \) elements can be computed in \( O(n) \) time. Suggested by the techniques in [16], we can claim the unique solution of \( g(\lambda) \) can be found in \( O(n \log(n)) \) time.

Here we exploit the fact that the right derivative of \( g \) can be computed at the time we compute the function value \( g \) at little additional computational cost. In our implementation, we adapt the Newton-Secant scheme described in [18] to solve for \( g(\lambda) = 0 \). We replace the derivative in [18] by the right derivative in the Newton step.

**V. Numerical Experiments**

We performed several numerical experiments to test two basic aspects: the computational time required by our approach as opposed to standard solvers in finding a good solution to the original EV charger scheduling MINLP, and the solution quality of the homotopy heuristic. All tests were carried out on a 64-bit Windows 7 ThinkPad W520 with Intel i7-2720QM 2.2 GHz CPU and 8GB RAM. The solver used was CPLEX version 12.2 with default parameters.

**A. Knapsack-Type Subproblems: Comparison with CPLEX**

We first tested the benefit provided by our algorithm for the quadratic-cost knapsack-type problems described in Section IV-C. Randomly generated problem instances were tested on our approach as well as CPLEX. Our algorithm is terminated when \( |g(\lambda)| \leq 10^{-7} \) and the code was written in C. Values of vector \( a \) was chosen randomly from values of \( \{2, 2.5\} \), \( b \) are from \( \{0.8, 1.3\} \). Let \( \text{rand}(l, u) \) denote a scalar whose value is chosen randomly in the interval \((l, u)\). We took \( c = 1, d = \text{rand}(0.7, 1.3) \sum a_i, \beta = 500 + \text{rand}(-100, 100) \) and \( m = 1 \). Table I reports the average CPU time taken over 100 runs by each method to solve problems with increasing number of time-periods \( T \). Of course, in realistic settings \( T \) will be within 100

<table>
<thead>
<tr>
<th>( T )</th>
<th>( 1e+3 )</th>
<th>( 1e+4 )</th>
<th>( 1e+5 )</th>
<th>( 1e+6 )</th>
<th>( 1e+7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>0.0002</td>
<td>0.0016</td>
<td>0.0151</td>
<td>0.1728</td>
<td>1.8230</td>
</tr>
<tr>
<td>proposed</td>
<td>0.0233</td>
<td>0.1017</td>
<td>0.9273</td>
<td>18.7656</td>
<td>failed</td>
</tr>
</tbody>
</table>

(e.g. 96 intervals of 15 min each). Table II reports the total CPU time taken to solve knapsacks (KS) for 500 EVs over 10 iterations of Alg. 1 to solve (LSP). Each test shows a remarkable improvement in the computation time achieved by our method, savings of over 3 orders of magnitude.

**B. Homotopy Method and CPLEX**

The homotopy heuristic to solve a sequence of programs (\( \alpha \text{NLP} \)) to get a good approximation of the centralized MINLP was written in Matlab 7.12.

**Examples on small instances:** We use the small test instance described in Table III to show the effectiveness of our heuristic in approximating the true optimal solution in shifting EV load to fill the demand valley, and preventing the blow-up of transformer while maintaining the fairness of charging. CPLEX was able to find the optimal solution for problems of these sizes, thus we can compare our heuristic solutions to the exact solutions. Table IV compares the results for two cases of transformer capacity constraint settings: sufficient (\( \geq 100\% \)) and limited (\( \leq 85\% \)).

**Examples on large instances:** Finally, we test the performance of our proposed algorithm (including homotopy, decomposition and knapsack steps) for problems with practical sizes. We solve for the problem of a day-ahead charging schedule of \( T = 24 \) time periods with interval length \( \Delta T = \)

**TABLE III**

<table>
<thead>
<tr>
<th>( T )</th>
<th>( E^T V^1 )</th>
<th>( E^T V^2 )</th>
<th>( E^T V^3 )</th>
<th>( E^T V^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T^\alpha )</td>
<td>2.5</td>
<td>3</td>
<td>2.5</td>
<td>2</td>
</tr>
<tr>
<td>( r^\alpha )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( B_\alpha )</td>
<td>( {1, 1.2, 2, 6, 5, 6, 8} )</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1 hour. In practice, this is a very large-scale MINLP problem having a large number of binary variables; for instance when \( N = 200 \), the number of binary variables is 9600. CPLEX was not able to solve any of these test cases. The base demand is the average residential load profile in the area of South California Edition from 20:00 on 02/13/2011 to 19:00 on 02/14/2011 as reported in [6]. Each vehicle has a fast charging rate of 3.3 kW, and a slow rate of 1.44 kW [7]. The total energy demand for each EV is randomly chosen from 10 to 20 kWh. Fig. 1 reports both the aggregated demand curves and the CPU time taken to solve problems of sizes ranging from 100 to 600 EVs. The objective of flattening the overall load is being well met, and the total time taken to solve shows only a mild super-linear growth. The final Figure 2 provides a box-plot of the fraction of demand satisfied for a randomly generated instance of a problem with 550 EVs. The fairness parameter \( \beta \) was set to \( 5000 \frac{R}{E} \), that is, inversely proportional to the quickest time each EV takes to fully charge. The plot shows that the identified solution does a good job of fairly allocating the available capacity (a fraction of the total demand) to all the EVs, with the middle 50% of the population accommodated in reasonably narrow boxes. Also, no EV is overly penalized, though there are some positive outliers.

### VI. CONCLUSIONS

This article studies the charging of a collection of electric vehicles fed by a central authority. We propose a distributed decision-making scheme where each charger individually determines its own charging schedule iteratively by transacting signals with the authority. Our model introduces three unique characteristics: capacity constraints on the distribution grid, fair rationing of energy supply available under the capacity constraint, and discrete choice of EV charger settings. The centralized version of the problem is an MINLP, and standard solvers find problems of even toy sizes too hard. We propose a multi-step approach that combines homotopic approximations with a decomposition into knapsack-like quadratic programs solved for each EV. The ultimate computational burden is thus placed on the individual EVs. We provide a linearithmic time method to solve these EV knapsack-type programs.

### REFERENCES


