2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Stackelberg Game Approach to Control the Overall Load Consumption of a Residential Neighborhood

Erhan Can Ozcan[1] and Ioannis Ch. Paschalidis[2]

Abstract— **This paper formulates a Stackelberg game between a coordination agent and participating homes to control the overall load consumption of a residential neighborhood. Each home optimizes a comfort-cost trade off to determine a load schedule of its available appliances in response to a price vector set by the coordination agent. The goal of the coordination agent is to find a price vector that will keep the overall load consumption of the neighborhood around some target value. After transforming the bilevel optimization problem into a single level optimization problem by using Karush-Kuhn-Tucker (KKT) conditions, we model how each home reacts to any change in the price vector by using the implicit function theorem. By using this information, we develop a distributed optimization framework based on gradient descent to attain a better price vector. We verify the load shaping capacity and the computational performance of the proposed optimization framework in a simulated environment establishing significant benefits over solving the centralized problem using commercial solvers.**

I. INTRODUCTION

Maintaining a balance between electricity demand and supply is extremely important to ensure the reliability of power grids, and the main challenge utility companies are facing is to meet the electricity demand during peak hours [1]. In order to overcome this issue, utility companies dispatch additional power plants with high operating costs during peak hours. The increasing production from intermittent renewable sources augments the need for such reserve capacity. Moreover, as these power plants mostly rely on fossil fuels to generate electricity, they increase carbon dioxide emissions and accelerate climate change [2]. According to the report published by the U.S. Energy Information Administration, the energy consumption in residential dwellings will increase by 22% until 2050, and the main underlying factor behind this increase is the population growth [3]. As the increase in the number of end users makes the electricity peak load problem more severe [1], the studies focusing on controlling the load consumption in residential homes and buildings is of utmost importance for the society.

Deployment of smart meters and *Home Energy Management Systems* (HEMS) to manage home energy consumption and energy costs has become popular after the developments in information and communication technologies [4]. The change in the energy usage of a consumer in response to varying electricity prices or to incentive payments is referred as *Demand Response* (DR) [5].

This paper proposes a demand response program with dynamic pricing, which has shown promising results to decrease the peak load in real-world applications [6], [7]. A Stackelberg game formulation between the energy service provider and homes is introduced in [8], where the user comfort is represented by a utility function. However, it is not possible to develop a single utility function to guarantee user preferences are always met by the resulting load plan. Since the participation of users is a key factor in the success of demand response programs, methods failing to maintain the user comfort may not realize their full potential [9]. Hence, each appliance in HEMS should be modeled with enough details so that the user specific preferences can be reflected as closely as possible. In [10], the authors propose a Stackelberg game maximizing the profit of the utility company and minimizing the electricity bill of the homes. Although different appliances such as shiftable, non-shiftable and curtailable are considered, these appliances do not have enough details to reflect the real dynamics. In [11], the authors propose a Stackelberg game where the electricity retailer wants to control the overall load consumption of the neighborhood with sparse deviations from a target value. However, the resulting optimization problem is a non-convex *Quadratically Constrained Quadratic Programming* (QCQP) problem, which can be computationally challenging to solve, especially in large neighborhoods. Therefore, employing this model in real-world applications may not be practical unless efficient computational methods are developed to solve it. In [12], a *Mixed Integer Linear Programming* (MILP) is proposed to control the load consumption of the neighborhood without considering the electricity price. In this scenario, the proposed formulation is not a Stackelberg game but a single level optimization problem, and the authors solve it by Dantzig-Wolfe decomposition. However, the application of this strategy in real life may require participating homes to sign long term contracts with fixed rates, which makes it impossible to take advantage of the low electricity price when there is ample supply in the electricity market.

In this study, we formulate a bilevel optimization problem where the coordination agent announces a price vector, and then each home determines the optimal load schedule of the appliances in response to that price vector according to its own comfort-related preferences. Since each home solves

The research was partially supported by the DOE under grants DE-EE0009696 and DEAR0001282, the NSF under grants CCF2200052, DMS-1664644, and IIS1914792, and by the ONR under grants N000141912571 and N000142112844.

[1] Erhan Can Ozcan is with Division of Systems Engineering, Boston University, Boston, MA 02215, USA cozcan@bu.edu.

[2] Ioannis Ch. Paschalidis is with the Department of Electrical and Computer Eng., Division of Systems Eng., and Dept. of Biomedical Eng., Boston University, 8 St. Mary's St., Boston, MA 02215, USA yannisp@bu.edu.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

a quadratic programming problem to find its optimal load schedule, we transform the bilevel optimization problem into a single level optimization problem by using Karush-Kuhn-Tucker (KKT) optimality conditions. Solving the reformulated single level problem by using commercial solvers is the state-of-the-art strategy to solve bilevel optimization problems [13], [14]. However, we stress that this solution strategy has two major drawbacks preventing it from being used in practice. First, the problem complexity grows exponentially as the target community becomes larger, which makes attaining an effective solution in a reasonable amount of time difficult. Second, the coordination agent needs to know both the preferences and the sensitive personal data of homes to be able to formulate the aforementioned optimization problem, which can raise some data privacy concerns among users. In order to address the problems related to efficiency and data privacy, we develop a gradient based distributed optimization framework similar to the approach introduced in [13].

We compare the proposed optimization framework with a commercial solver in terms of the optimization time and the solution quality. Our experiments demonstrate that the commercial solver fails to solve the centralized problem within the desired time interval when the optimization problem includes more than a few homes. In most of the cases, it cannot even find an initial feasible solution. On the other hand, the distributed optimization framework is able to provide an effective solution in a short period of time. Furthermore, we show that better solutions can be obtained by utilizing mini-batch gradient descent optimization techniques.

Our contributions can be summarized as follows: 1) We formulate a Stackelberg game between a coordination agent and participating homes to minimize the peaks observed in the power profile of the community without severely affecting consumer comfort. 2) We develop a gradient-based distributed optimization approach to solve the proposed problem efficiently, and compare it with a commercial solver. 3) We show that stochastic unbiased estimators for the gradient exist, and using stochastic gradients can bring benefits while optimizing the formulated problem.

Notation: We denote vectors by bold lowercase letters (e.g., $\boldsymbol{p}$ and $\boldsymbol{\pi}$), and matrices by bold uppercase letters (e.g., $\boldsymbol{G}$). We use prime to denote the transpose operation. For space-saving reasons, we write $\boldsymbol{x} = (x(1) \ ... \ x(n))$ for a column vector $\boldsymbol{x}$ in $\mathbb{R}^n$.

II. Modeling Home Appliances

The electricity consumption in residential dwellings is largely attributed to space heating and cooling, water heating, electric vehicle (EV) charging and some other routine home appliances. Therefore, controlling these loads can be an effective way to deal with the peak load problem. In order to ensure the user comfort at any time, each home is equipped with an embedded *Model Predictive Controller* (MPC) that controls the activity of available appliances in accordance with the user preferences over a time horizon. The appliances considered in this study are the heating, ventilation, and air conditioning (HVAC) system, electric water heater, electric vehicle, and ON/OFF type basic appliances such as washing machine, dryer, and oven. We consider appliance models with the same properties used in [12] except minor simplifications. In this study, the comfort related constraints of the users are defined by using only linear inequalities, and the full set of constraints with detailed explanation can be found in [15].

III. Proposed Optimization Problem

Consider a power grid managed by a coordination agent serving N participating homes equipped with M appliances introduced previously. The total load consumption of the community can be controlled by formulating a Stackelberg game where the coordination agent is the leader and the homes are followers. In this game, the coordination agent announces a price vector, then homes automatically adjust the power consumption schedule of the available appliances, minimizing a cost function that accounts for their individual preferences.

Suppose that $p_{ij}(t)$ is a decision variable denoting the load consumption of home i for appliance j at time point t, and

$$\boldsymbol{p}_i = (p_{i1}(0) \ ... \ p_{i1}(K-1) \ ... \ p_{iM}(0) \ ... \ p_{iM}(K-1))$$

is a vector of decision variables denoting the load schedule of home i over the next K time intervals. Each home defines a set of comfort related linear constraints according to its preferences, and these constraints form the polyhedron $\mathcal{P}_i = \{\boldsymbol{p}_i \mid \boldsymbol{G}_i \boldsymbol{p}_i \leq \boldsymbol{h}_i\}$, $\forall i$, where $\boldsymbol{G}_i$ and $\boldsymbol{h}_i$ denote the constraint matrix and the right hand side vector for home i, respectively. Hence, the comfort of home i can be ensured as long as $\boldsymbol{p}_i \in \mathcal{P}_i$. Although determining $\boldsymbol{p}_i$ by solving an optimization problem, which minimizes the electricity bill according to the announced price vector, $\boldsymbol{\pi} = (\pi(0) \ ... \ \pi(K-1))$, over the set $\mathcal{P}_i$, is possible, there is an oversight in this strategy. Each home has different consumption habits, thus a desirable load schedule for each appliance. As any deviation from this schedule causes a social discomfort for the users, these deviations must be taken into account as well as the electricity bill while determining the load schedule of homes. Then, for a given price vector $\boldsymbol{\pi}$, the objective function of home i can be expressed as follows:

$$\begin{aligned} f_i(\boldsymbol{p}_i, \boldsymbol{\pi}) = & \sum_{j=1}^{M} \sum_{t=0}^{K-1} p_{ij}(t)\pi(t) \\ & + \sum_{j=1}^{M} \sum_{t=0}^{K-1} c_{ij} \left(p_{ij}(t) - \overline{p_{ij}}(t)\right)^2, \end{aligned} \tag{1}$$

where c_{ij} represents the importance of appliance j for home i, and $\overline{p_{ij}}(t)$ represents the desirable load of appliance j in home i at time point t. While the first term in Equation (1) calculates the electricity cost, the second term considers the cost incurred due to deviations from the desirable load curve. Accordingly, home i solves:

$$\boldsymbol{p}_i^* := \underset{\boldsymbol{p}_i \in \mathcal{P}_i}{\arg\min} \ f_i(\boldsymbol{p}_i, \boldsymbol{\pi}). \tag{2}$$

On the other hand, the coordination agent wants to keep the total load consumption of the community under control while minimizing the total social discomfort of the community by choosing an appropriate electricity price vector $\boldsymbol{\pi}$. Therefore, for a given $\boldsymbol{\pi}$, $\boldsymbol{p}^*$ denotes the optimal load consumption vector of the neighborhood, e.g., $\boldsymbol{p}^* = (\boldsymbol{p}_1^* \ldots \boldsymbol{p}_N^*)$, and the objective function of the coordination agent can be expressed as follows:

$$f(\boldsymbol{p}^*, \boldsymbol{\pi}) := \sum_{t=0}^{K-1} \left(Q(t) - \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}^*(t) \right)^2 + \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=0}^{K-1} c_{ij} \left(p_{ij}^*(t) - \overline{p_{ij}}(t) \right)^2, \quad (3)$$

where $Q(t)$ represents the targeted aggregate power consumption level at time t. While the first term in (3) minimizes the deviation from $Q(t)$ at any time interval t, the second term minimizes the total social discomfort of the neighborhood.

The coordination agent seeks a price vector minimizing its objective function. However, as the coordination agent has to act in accordance with some regulations, the electricity price cannot exceed a certain threshold. Similarly, the coordination agent cannot lower the price freely because of the cost of generating electricity. We assume that these requirements are satisfied as long as $\boldsymbol{\pi} \in \mathcal{S}$, where $\mathcal{S}$ is a bounded hypercube. The resulting bilevel optimization problem is:

$$\min_{\pi \in \mathcal{S}} \quad f(\boldsymbol{p}^*, \boldsymbol{\pi}) \quad (4)$$

$$\text{s.t.} \quad \boldsymbol{p}_i^* := \arg\min_{\boldsymbol{p}_i \in \mathcal{P}_i} \; f_i(\boldsymbol{p}_i, \boldsymbol{\pi}), \quad \forall i. \quad (5)$$

For a given price vector, the home level optimization problem is a convex quadratic program. Therefore, the bilevel problem given in Equations (4)-(5) can be reformulated as a single level problem by replacing the optimality constraint given in Equation (5) with the KKT conditions of each home:

$$\min_{\substack{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_N,\\ \boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_N,\\ \boldsymbol{\pi} \in \mathcal{S}}} \quad f(\boldsymbol{p}, \boldsymbol{\pi}) \quad (6)$$

$$\begin{aligned}
\text{s.t.} \quad & \nabla_{\boldsymbol{p}_i} f_i(\boldsymbol{p}_i, \boldsymbol{\pi}) + \boldsymbol{G}_i' \boldsymbol{\lambda}_i = 0, && \forall i, && (7)\\
& \text{Diag}(\boldsymbol{\lambda}_i) \left(\boldsymbol{G}_i \boldsymbol{p}_i - \boldsymbol{h}_i \right) = 0, && \forall i, && (8)\\
& \boldsymbol{G}_i \boldsymbol{p}_i \leq h_i, && \forall i, && (9)\\
& \boldsymbol{\lambda}_i \geq 0, && \forall i, && (10)
\end{aligned}$$

where $\boldsymbol{\lambda}_i$ is a vector of the dual variables corresponding to the constraints in the set $\mathcal{P}_i$. Since the objective function of each home is quadratic, the stationarity constraints given in (7) are linear. However, the constraints in (8) enforcing the complementary slackness condition are non-convex. Thus, the optimization problem in Equations (6)-(10) is a non-convex *Quadratically Constrained Quadratic Program* (QCQP).

IV. GRADIENT BASED OPTIMIZATION FRAMEWORK

Although the commercial solvers can handle non-convex QCQP problems by transforming them into *Mixed Integer Programs* (MIP), solving MIP problems can be computationally intractable when the problem size grows. Therefore, we propose an efficient gradient-based optimization technique to solve the formulated problem. Since the coordination agent minimizes its objective function by choosing a price vector, the gradient that we need to compute is as follows:

$$\nabla_{\boldsymbol{\pi}} f(\boldsymbol{p}^*, \boldsymbol{\pi}) = f_{\boldsymbol{\pi}}(\boldsymbol{p}^*, \boldsymbol{\pi}) + f_{\boldsymbol{p}}(\boldsymbol{p}^*, \boldsymbol{\pi}) \nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*, \quad (11)$$

where $f_{\boldsymbol{\pi}}$ and $f_{\boldsymbol{p}}$ represent the partial derivatives, and $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$ is a Jacobian matrix showing how the neighborhood alters its optimal load schedule in response to any change in the price vector. Since the objective function of the coordination agent does not involve the price vector, $f_{\boldsymbol{\pi}}$ is equal to 0. Thus, the gradient calculation can be reduced to

$$\nabla_{\boldsymbol{\pi}} f(\boldsymbol{p}^*, \boldsymbol{\pi}) = f_{\boldsymbol{p}}(\boldsymbol{p}^*, \boldsymbol{\pi})' \nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*. \quad (12)$$

While it is straightforward to calculate $f_{\boldsymbol{p}}$, estimating $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$ requires a bit more work. For a given $\boldsymbol{\pi}$, the optimization problem solved by home i, $f_i(\boldsymbol{p}_i, \boldsymbol{\pi})$, is a quadratic program with linear constraints. In [16], how the optimal solution responds to any change in the problem parameters (e.g., cost function, coefficients in constraint matrix) is shown by taking matrix differentials of the KKT conditions. Moreover, [17] provides a detailed derivation of this approach by using the implicit function theorem. Finally, [13] develops a gradient descent-based approach to optimize the leader's strategy in a Stackelberg game. Similar to [13], each home solves a convex optimization problem with a twice differentiable objective function in our case. Therefore, we describe how to obtain $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$ by recalling the strategy illustrated in [13].

Since the inequality constraints in Equation (9) are affine, the weak form of Slater's condition holds. Thus, around $\boldsymbol{p}_i^*$, the KKT conditions we have to maintain reduce to the stationarity and the complementary slackness conditions. Then, by taking the differentials of the remaining KKT conditions of home i, we obtain the following set of equations in matrix form:

$$\begin{bmatrix} \nabla^2_{\boldsymbol{p}_i \boldsymbol{p}_i} f_i & \boldsymbol{G}_i' \\ \text{Diag}(\boldsymbol{\lambda}_i^*) \boldsymbol{G}_i & \text{Diag}(\boldsymbol{G}_i \boldsymbol{p}_i^* - \boldsymbol{h}_i) \end{bmatrix} \begin{bmatrix} d\boldsymbol{p}_i^* \\ d\boldsymbol{\lambda_i}^* \end{bmatrix} = \begin{bmatrix} -\nabla^2_{\boldsymbol{\pi} \boldsymbol{p}_i} f_i d\boldsymbol{\pi} - d\boldsymbol{G}_i' \boldsymbol{\lambda}_i^* \\ -\text{Diag}(\boldsymbol{\lambda}_i^*)(d\boldsymbol{G}_i \boldsymbol{p}_i^* - d\boldsymbol{h}_i) \end{bmatrix}. \quad (13)$$

If we set $d\boldsymbol{\pi}$ and dG_i to the identity and the zero matrices of appropriate dimensions, respectively, and solve the system of equations above for $d\boldsymbol{p}_i^*$, we obtain the Jacobian matrix $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^*$. However, as we want to obtain $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$, we need to write the equations above for each home, and solve the following set of equations:

$$\begin{bmatrix} \boldsymbol{C} & \boldsymbol{G}' \\ \text{Diag}(\boldsymbol{\lambda}^*) \boldsymbol{G} & \text{Diag}(\boldsymbol{G} \boldsymbol{p} - \boldsymbol{h}) \end{bmatrix} \begin{bmatrix} d\boldsymbol{p}^* \\ d\boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{\pi}} \boldsymbol{v} \\ 0 \end{bmatrix}, \quad (14)$$

where $\boldsymbol{C} = \text{Diag}(\nabla^2_{\boldsymbol{p}_1\boldsymbol{p}_1} f_1,\ \ldots,\ \nabla^2_{\boldsymbol{p}_N\boldsymbol{p}_N} f_N)$, $\boldsymbol{G} = \text{Diag}(\boldsymbol{G}_1,\ \ldots,\ \boldsymbol{G}_N)$, and $\boldsymbol{v} = \left[(\nabla_{\boldsymbol{p}_1} f_1)' \ \ldots\ (\nabla_{\boldsymbol{p}_N} f_N)'\right]'$. Unlike [13], since the optimal action of each follower does not depend on the other followers' actions in our problem, the large matrix on the left-hand side of Equation (14) consists of diagonal blocks which do not interact with other followers' data. Hence, we can calculate $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$ in a distributed way.

Algorithm 1 The framework of a distributed algorithm with a given maximum number of iterations $k^{\max}$, and threshold on objective improvement rate ϵ.

```
Input: Batch size $B$, learning rate $\alpha$, online optimization algorithm $\mathbb{A}$, bounded hypercube $\mathcal{S}$.
Initialize
Select a random price vector $\boldsymbol{\pi}_1 \in \mathcal{S}$, and set $\boldsymbol{\pi} = \boldsymbol{\pi}_1$.
Set $z_0 = \infty$.
for $k = 1\ \ldots\ k^{\max}$ do
    Set $\mathbb{B} = \emptyset$, and randomly select $B$ homes.
    Store the indices of these homes in $\mathbb{B}$.
    Set $\boldsymbol{g}_k = 0$.
    for each $i \in \mathbb{B}$ do
        Solve Equation (2) to obtain $\boldsymbol{p}_i^*$.
        Solve Equation (13) for $d\boldsymbol{p}_i^*$ to obtain $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^*$.
        $\boldsymbol{g}_k = \boldsymbol{g}_k + f_{\boldsymbol{p}_i}(\boldsymbol{p}^*, \boldsymbol{\pi})' \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^*$.
    end for
    Send $\boldsymbol{g}_k$, $\alpha$, and $\boldsymbol{\pi}$ to Algorithm $\mathbb{A}$.
    Receive $\boldsymbol{\pi}_{k+1}$ from Algorithm $\mathbb{A}$.
    $\boldsymbol{\pi}$ = Projection of $\boldsymbol{\pi}_{k+1}$ onto $\mathcal{S}$.
    $z_k = f(\boldsymbol{p}^*, \boldsymbol{\pi})$.
    if $|z_k - z_{k-1}|/z_{k-1} \leq \epsilon$ then
        break
    end if
end for
return $\boldsymbol{\pi}$.
```

In reality, we do not even need to construct the Jacobian matrix $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}^*$ since the gradient calculation in Equation (12) is equivalent to:

$$\nabla_{\boldsymbol{\pi}} f(\boldsymbol{p}^*, \boldsymbol{\pi}) = \left[f_{\boldsymbol{p}_1}(\boldsymbol{p}^*, \boldsymbol{\pi})' \ \ldots\ f_{\boldsymbol{p}_N}(\boldsymbol{p}^*, \boldsymbol{\pi})' \right] \begin{bmatrix} \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_1^* \\ \vdots \\ \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_N^* \end{bmatrix} \tag{15}$$

$$= \sum_{i=1}^{N} f_{\boldsymbol{p}_i}(\boldsymbol{p}^*, \boldsymbol{\pi})' \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^*, \tag{16}$$

where we can obtain $\nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^*$ for each home i by solving Equation (13) as it is already explained. The overall optimization problem we formulate is non-convex, and we know that the gradient based approaches can get stuck at local minima even if the step length is chosen appropriately. On the other hand, the Stochastic Gradient Descent (SGD)-based approaches may avoid these local minimums by utilizing the randomness during the global minimum search. Thus, they are extensively used to optimize non-convex loss functions, especially in studies involving artificial neural networks. Fortunately, we can employ SGD-based optimization techniques in our problem since it is possible to obtain unbiased estimators for $\nabla_{\boldsymbol{\pi}} f(\boldsymbol{p}^*, \boldsymbol{\pi})$. We can make this observation quickly after manipulating Equation (16) in the following way:

$$\nabla_{\boldsymbol{\pi}} f(\boldsymbol{p}^*, \boldsymbol{\pi}) = N \sum_{i=1}^{N} \frac{1}{N} f_{\boldsymbol{p}_i}(\boldsymbol{p}^*, \boldsymbol{\pi})' \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^* \tag{17}$$

$$= N\ \mathbb{E}_{i \sim z} \left[f_{\boldsymbol{p}_i}(\boldsymbol{p}^*, \boldsymbol{\pi})' \nabla_{\boldsymbol{\pi}} \boldsymbol{p}_i^* \right], \tag{18}$$

where z denotes a Discrete Uniform$(1, N)$, and $\mathbb{E}$ is the expectation operator.

The pseudocode showing the steps of the distributed optimization algorithm is given in Algorithm 1. Since each home independently performs the inner loop steps, we have a distributed optimization algorithm. Therefore, the designed algorithm solves the issues related to efficiency and data privacy.

V. Experiments

In this study, we propose not only a Stackelberg game that can keep the load consumption of the neighborhood around some target level without causing too much social discomfort but also a gradient based distributed optimization framework that can find effective solutions in a reasonable amount of time. In order to test these hypotheses, we conduct various experiments on a simulation environment. First, the details of our experimental setup are explained in Section V-A. Next, the optimization performance of the proposed optimization framework with different parameters is observed, and the optimization performance is compared with a commercial solver in Section V-B. Finally, the capability of the proposed optimization problem to control the load consumption of the community is shown in Section V-C.

A. Simulation Setup

A simulation environment that can create a residential neighborhood, in which each home is equipped with the appliances mentioned in Section II, is designed in Python, and we utilize the Gurobi 9.5.1 package to solve both the centralized non-convex QCQP problem given in Equations (6)-(10) and the convex optimization problem given in (2).

The properties of the appliances are identical across the neighborhood. However, both the preferences and the consumption habits of households can vary for each appliance, thus their desirable load schedules. Moreover, the cost of deviation from the desirable load curve is different for each home and appliance. We follow the same experimental setup used in [12] while generating our simulation environment. The probability distributions that we use to sample the preferences and penalty coefficients, c_{ij}s, of households as well as the details of the appliance properties are provided in [15]. Additionally, the source code is publicly available[1].

In our simulations, we assume that the length of the planning horizon is 24 hours, and each time interval lasts

[1] Code available at `https://github.com/erhancanozcan/energy-qcqp-grad`.

15 minutes ($K = 96$). Since the desirable load consumption schedule of each home changes dynamically at each time interval, the optimization algorithm must provide a reasonably good price vector $\boldsymbol{\pi}$ in less than 15 minutes. We assume that the electricity price always has to be in a certain interval, and the price range we consider in our experiments is as follows:

$$\mathcal{S} = \{\boldsymbol{\pi} \mid 0.1 \leq \pi(t) \leq 1.0, \quad \forall t\}.$$

Finally, $Q(t)$, the target load level the coordination agent wants to achieve at time interval t, is set as the time average of total desirable power consumption. In other words, $Q(t)$ takes the form

$$Q(t) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=0}^{K-1} \overline{p_{ij}}(t)}{K}, \ \forall t.$$

B. Computational Performance of Distributed Optimization

The proposed algorithm has three parameters, and the performance of the proposed algorithm changes with respect to these parameters. The first parameter is the batch size B, and we have two levels for this parameter in our experiments. It is either equal to 25 or equal to the total number of homes in the community (e.g., full batch). Secondly, we consider the Adam optimizer and the scaled SGD (e.g., $\alpha \propto \frac{1}{\sqrt{k}}$, where k is the iteration number) as two options for the online learning algorithm $\mathbb{A}$. Lastly, the possible α values we consider for each online learning algorithm vary. While two α values we consider for the scaled SGD are $1e^{-5}$ and $1e^{-6}$, the possible learning rate options for the Adam optimizer are $1e^{-1}$ and $1e^{0}$. In addition to these three parameters, we set $k^{\max}$ to 50, and ϵ to $1e^{-3}$ in our experiments.

In order to observe how the performance of the proposed algorithm changes with respect to different parameter settings, we randomly initialize communities with varying number of homes (e.g., 50, 100, 250). To decrease the effect of random initialization, we use five different seeds in our experiments. Table 1 shows the number of times in which each algorithm yields the lowest objective value among all options across five trials.

TABLE I
LOWEST OBJECTIVE ACROSS FIVE TRIALS

Algorithm 1 w/parameters	50 homes	100 homes	250 homes
Adam_B25_α1en1	2	2	3
Adam_B25_α1en0	0	0	0
Adam_fullbatch_α1en1	0	0	0
Adam_fullbatch_α1en0	0	0	0
scaledSGD_B25_α1en5	1	1	1
scaledSGD_B25_α1en6	0	1	1
scaledSGD_fullbatch_α1en5	0	0	0
scaledSGD_fullbatch_α1en6	2	1	0

According to Table I, a mini batch optimization technique helps our algorithm to attain better objective values. On the other hand, both the Adam optimizer and the scaled SGD may yield the lowest objective value depending on the problem at hand. In order to decide which setup yields the best performance in Algorithm 1, we compare the competing setups based on their average ranks over attained objective value. Figure 1 provides the average ranks of each competing setting with critical differences where lower rank corresponds to a better performance.

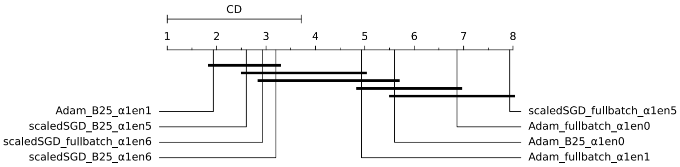

Fig. 1. The average ranks over attained objective value for all competing methods on 15 problems. Lower rank corresponds to a better performance. Bold horizontal lines shows the methods that are not different in a 5% significance level according to the Nemenyi test.

According to Figure 1, the Adam optimizer with mini batch seems as a good alternative to attain lower objective values across a range of problems although there is no statistically significant difference with the setups using the scaled SGD as the optimization algorithm. We pick the top three options in the average rank plot as our candidate setups, and continue our analysis with those.

It is possible to solve the centralized version of the formulated problem by using the non-convex QCQP solver available in Gurobi, and this strategy is a natural competitor of our algorithm. However, as the centralized version is an MIQP problem, it is hard to solve this problem optimally. Unfortunately, Gurobi fails to solve this problem, and it cannot even find a feasible point within the desired time limit in many trials. The optimization iterations of our algorithm start using a random price vector, and a corresponding optimal load schedule plan of the community. By using this information, an initial feasible solution can be easily found for the MIQP problem, and we provide this information to Gurobi in order to have a fair comparison between our algorithm and Gurobi. Even in this case, we observe either a slight improvement in the objective or no improvement at all when Gurobi is employed. Suppose z^{G} denotes the objective value calculated by Gurobi at the end of 900 seconds for a given problem, and the best objective value that the n-th method can find within this time period is denoted by z^{method_n}. Then, we define the improvement ratio of the n-th method as follows:

$$\text{Ratio (method}_n) = \frac{z^{\mathrm{G}} - z^{\mathrm{method}_n}}{z^{\mathrm{method}_n}}, \ \forall \ n.$$

Table II provides the mean improvement ratio for each algorithm.

TABLE II
AVERAGE IMPROVEMENT RATIO COMPARED TO GUROBI

Algorithm 1 w/parameters	50 homes	100 homes	250 homes
Adam_B25_α1en1	35.2	35.9	34.6
scaledSGD_B25_α1en5	32.7	33.1	34.4
scaledSGD_fullbatch_α1en6	34.9	35.6	30.4

According to Table II, on average, the objective values provided by our algorithm are at least 30 times better than the objective values provided by Gurobi. Moreover, the run time of our algorithm is significantly shorter than Gurobi. Table III summarizes the average run time of our algorithm under different parameter settings. In all cases, our algorithm can provide the solution within the time limit.

TABLE III
AVERAGE RUN TIME IN SECONDS

Algorithm 1 w/parameters	50 homes	100 homes	250 homes
Adam_B25_α1en1	130.9	165.3	225.2
scaledSGD_B25_α1en5	95.8	116.9	222.4
scaledSGD_fullbatch_α1en6	65.7	41.1	73.3

C. The Load Shaping Capacity

In this section, we show the capability of the proposed formulation to shape the overall load consumption of the community after optimizing the price vector via the proposed algorithm. Initially, the desirable load consumption of home i at time t is equal to $\sum_{j=1}^{M} \overline{p_{ij}}(t)$, and the desirable load consumption of the whole community is equal to $\sum_{i=1}^{N} \sum_{j=1}^{M} \overline{p_{ij}}(t)$. On the other hand, $\sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}^{*}(t)$ denotes the optimal load consumption of the community at time t with respect to the optimized price vector $\boldsymbol{\pi}^{*}$. Figure 2 shows how the overall desirable load consumption of the community changes when the coordination agent announces an optimized price vector.

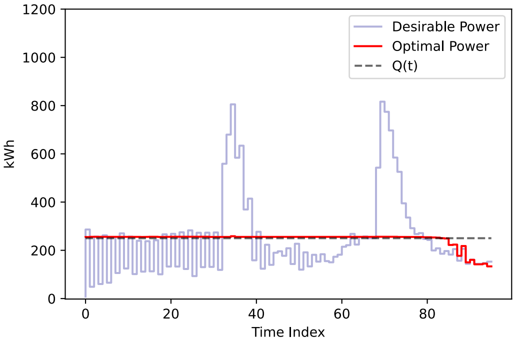

Fig. 2. The desirable and the optimal power consumption profiles of the community consisting of 100 homes (Adam optimizer with mini batch).

According to Figure 2, the observed peaks in the desirable load consumption profile of community are prevented by setting the electricity price vector appropriately. In Figure 2, the horizontal dashed line denotes the targeted aggregate power consumption level set by the coordination agent, and we observe that the optimal load consumption curve of the community mostly matches this target. Therefore, our approach can be helpful to maintain the balance between electricity demand and supply as well as mitigating the peak load problem.

VI. CONCLUSIONS

In this paper, we present a Stackelberg game to control the overall load consumption of a residential neighborhood. As the control decisions are taken by considering the individual preferences of the households, we ensure that the user comfort is maintained at any time. However, refining the appliance models to reflect the user comfort better is one of our future goals. Although it can be challenging to solve the formulated problem, the proposed distributed optimization framework can obtain effective solutions in a significantly shorter time period. Moreover, our approach simplifies electricity generation planning by balancing load consumption throughout the planning horizon and reducing peak demand. As future work, we want to investigate how we can use our algorithm to help Gurobi to start its optimization from a better branch, as part of its branch and bound search.

REFERENCES

[1] M. Uddin, M. F. Romlie, M. F. Abdullah, S. Abd Halim, T. C. Kwang, *et al.*, “A review on peak load shaving strategies,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3323–3332, 2018.

[2] S. Gyamfi, S. Krumdieck, and T. Urmee, “Residential peak electricity demand response—highlights of some behavioural issues,” *Renewable and Sustainable Energy Reviews*, vol. 25, pp. 71–77, 2013.

[3] U.S. EIA. Annual energy outlook 2022. U.S. EIA. [Online]. Available: https://www.eia.gov/outlooks/aeo/narrative/consumption/sub-topic-03.php

[4] A. Pratt, D. Krishnamurthy, M. Ruth, H. Wu, M. Lunacek, and P. Vaynshenk, “Transactive home energy management systems: The impact of their proliferation on the electric grid,” *IEEE Electrification Magazine*, vol. 4, no. 4, pp. 8–14, 2016.

[5] P. Siano, “Demand response and smart grids—a survey,” *Renewable and sustainable energy reviews*, vol. 30, pp. 461–478, 2014.

[6] L. Wen, K. Zhou, W. Feng, and S. Yang, “Demand side management in smart grid: A dynamic-price-based demand response model,” *IEEE Transactions on Engineering Management*, 2022.

[7] B. Foster, T. Bialecki, D. Burns, D. Kathan, M. P. Lee, and S. Peirovi. (2019) Assessment of demand response and advanced metering. Federal Energy Regulatory Commission. [Online]. Available: https://www.ferc.gov/sites/default/files/2020-04/DR-AM-Report2019{_}2.pdf

[8] M. Yu and S. H. Hong, “A real-time demand-response algorithm for smart grids: A stackelberg game approach,” *IEEE Transactions on smart grid*, vol. 7, no. 2, pp. 879–888, 2015.

[9] X. Jin, K. Baker, D. Christensen, and S. Isley, “Foresee: A user-centric home energy management system for energy efficiency and demand response,” *Applied Energy*, vol. 205, pp. 1583–1595, 2017.

[10] F.-L. Meng and X.-J. Zeng, “A stackelberg game-theoretic approach to optimal real-time pricing for the smart grid,” *Soft Computing*, vol. 17, pp. 2365–2380, 2013.

[11] M. Zeifman, H. Fontenot, K. Roth, D. Leclerc, S. Taylor, G. Zavaliagkos, *et al.*, “Almost diy: A nonintrusive way to integrate homes with the electric grid,” in *2022 ACEEE Summer Study on Energy Efficiency in Buildings*, 2022.

[12] E. C. Ozcan and I. Paschalidis, “A distributed optimization framework to regulate the electricity consumption of a residential neighborhood,” *arXiv preprint:2306.09954*, 2023.

[13] K. Wang, L. Xu, A. Perrault, M. K. Reiter, and M. Tambe, “Coordinating followers to reach better equilibria: End-to-end gradient descent for stackelberg games,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 5, 2022, pp. 5219–5227.

[14] N. Basilico, S. Coniglio, N. Gatti, and A. Marchesi, “Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games,” *EURO Journal on Computational Optimization*, vol. 8, no. 1, pp. 3–31, 2020.

[15] E. C. Ozcan and I. Paschalidis, “A stackelberg game approach to control the overall load consumption of a residential neighborhood,” *arXiv preprint:2306.10935*, 2023.

[16] B. Amos and J. Z. Kolter, “Optnet: Differentiable optimization as a layer in neural networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 136–145.

[17] S. Barratt, “On the differentiability of the solution to convex optimization problems,” *arXiv preprint arXiv:1804.05098*, 2018.