A Privacy-Preserving Distributed Greedy Framework to Desynchronize Power Consumption in a Network of Thermostatically Controlled Loads

Mojtaba Kaheni, *Member, IEEE*, Alessandro V. Papadopoulos, *Senior, IEEE*, Elio Usai, *Senior, IEEE*, Mauro Franceschelli, *Senior, IEEE*

***Abstract*—This manuscript presents a novel distributed greedy framework applicable to a network of Thermostatically Controlled Loads (TCLs) to desynchronize the network's aggregated power consumption. Compared to the existing literature, our proposed framework offers two distinct novelties. Firstly, our proposed algorithm relaxes the restrictive assumptions associated with the communication graph among TCLs. To elaborate, our algorithm only requires a connected graph to execute control, a condition less demanding than its counterpart algorithms that mandate a star architecture, K-regular graphs, or undirected connected graphs. Secondly, a significant novel feature is the relaxation of the obligation to share private information, such as each unit's local power consumption and appliance temperatures, either with a central coordinator or neighboring TCLs. The findings presented in this paper are validated through simulations conducted over a network comprising 1000 TCLs.**

***Index Terms*—Demand Response, Distributed Optimization, Greedy Control, Multi-agent Systems, Thermostatically Controlled Loads.**

I. INTRODUCTION

Power networks and microgrids are experiencing a continuous increase in the installation of Renewable Energy Sources (RESs). These sources offer significant advantages in reducing environmental pollution and mitigating climate change. However, their efficiency in contributing to power system inertia is limited. Consequently, the vulnerability of power networks and microgrids to frequency variations rises as the penetration of RESs increases. Among various strategies to address this challenge, Demand-side Response (DSR) appears particularly promising [1], [2]. DSR has the potential to offer cost-effective solutions without causing substantial changes to the Quality of Service (QoS).

A significant proportion of residential energy consumption is attributed to Thermostatically Controlled Loads (TCLs), including electric water heaters, refrigerators, air conditioners, and more. Consequently, any residential DSR initiative should strategically consider this substantial potential to achieve its objectives. Studies conducted on TCL-based DSR programs in regions such as Germany [3], Great Britain [4], Sardegna [5], and others underscore the fundamental role TCLs play in the success of DSR.

TCL-DSR has been a prominent research topic for the past few decades. The first scheme to harness the potential of TCLs in DSR was direct load control [6], [7] where utilities have the ability to remotely manage customers' TCLs. With recent advancements in multi-agent systems and distributed control, centralized control frameworks have given way to distributed control methods. These methods make decisions in a distributed manner, relying on local measurements and/or estimations of QoS.

The terms *distributed control* and *multi-agent* often denote *distributed decision-making* in the literature of TCL-DSR. In this context, each appliance interacts with a central unit. This central unit collects data and updates local setpoints [8]–[11]. However, the requirement of a central aggregator renders the entire system susceptible to a single-point denial-of-service (DoS) attack. If the central aggregator becomes non-operational, the whole network will cease functioning. Moreover, customers' privacy might be compromised if agents transmit their real-time local and private power consumption information to a central unit. To address these limitations, it becomes imperative to eliminate the need for an aggregator and instead consider the adoption of *fully distributed* algorithms. Such approaches, as seen in [12]–[15], involve agents solely exchanging information with their neighboring TCLs.

According to *ENTSO-E* [16], strategies like those outlined in [8], [17], [18] that involve adjusting the TCLs' setpoints are no longer considered legit. The remaining feasible means of control include altering the typical hysteresis cycle of the appliances' thermostats while keeping their setpoints unchanged. This objective can be achieved either by directly modifying the hysteresis boundaries as proposed in [11] or by temporarily disconnecting and reconnecting the TCLs at appropriate intervals, as demonstrated in strategies such as [9], [14]. However, it's worth noting that modifying the hysteresis

This work was supported in part by the Project "Network 4 Energy Sustainable Transition–NEST" funded under the National Recovery and Resilience Plan (NRRP), (Mission 4, Component 2, Investment 1.3–Call for tender No. 1561 of 11.10.2022), Ministero dell'Università e della Ricerca (MUR), by the Fondazione di Sardegna with grant "Formal Methods and Technologies for the Future of Energy Systems", n. F72F20000350007, by the Swedish Research Council (VR) with grant "Pervasive Self-Optimizing Computing Infrastructures (PSI)" n. 2020-05094, by the Knowledge Foundation (KKS) with grants "Safe and Secure Adaptive Collaborative Systems (SACSys)", n. 20190021, and "Mälardalen University Automation Research Center (MARC)", n. 20240011, by the Swedish Agency for Innovation Systems (Vinnova) with grant "GREENER: Intelligent energy management in connected construction sites" n. 2019-05877.

M. Kaheni, and A. V. Papadopoulos are with IDT, Mälardalen University, 721 23 Västerås, Sweden. Emails: {mojtaba.kaheni,alessandro.papadopoulos}@mdu.se

E. Usai and M. Franceschelli are with DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: {mauro.franceschelli, elio.usai}@unica.it

Mauro Franceschelli is the corresponding author.

bounds of commercially available and pre-installed TCLs is not feasible. Therefore, although this method aligns with ENTSO-E regulations, it does not appear practical.

Identifying the dynamics of TCLs using an LTI model is a common practice in the literature. However, LTI models can be inaccurate for practical implementation [19] due to disturbances like water withdrawal and seasonal changes. To the best of our knowledge, only a few recent papers have explored model-free algorithms, such as those in [9], [15]. In [9], a priority-based optimization method is used to desynchronize power demand by controlling TCLs. The algorithm leverages data on the TCL temperatures and duty cycles, dispensing with the explicit use of the temperature model. Subsequently, each TCL autonomously takes actions based on its score and the optimization results. In [15], two model-free and distributed algorithms are proposed for tracking the desired load. The first algorithm, applicable in K-regular graphs, involves each agent collecting real-time power consumption data from its neighbors to make control decisions. The requirement of a K-regular graph is subsequently relaxed using a dynamic consensus protocol, which estimates the mean network power consumption. In this paper, we also introduce a model-free scheme. The primary distinctions between this paper and the approaches presented in [9] and [15] are outlined below:

- The algorithm in [9] necessitates an aggregator within its framework. In contrast, the algorithm proposed in this article is fully distributed, wherein each agent communicates solely with its neighboring agents.
- Both algorithms introduced in [15] require the exchange of real-time power consumption data among neighboring agents, potentially raising privacy concerns. In contrast, our proposed algorithm involves the transfer of only the Lagrangian multipliers among agents, thus maintaining the agents' privacy.
- Both [9] and [15] assume the availability of temperature measurements from thermostatically controlled parameters for control purposes. This assumption implies a requirement for smart appliances. As a result, these algorithms are not suitable for off-the-shelf TCLs. In contrast, our proposed method eliminates this requirement and is compatible with off-the-shelf appliances.

A. Statement of contributions

In this article, we present a privacy-preserving distributed greedy algorithm designed to desynchronize power consumption within a network of TCLs. Our model-free approach offers a straightforward yet highly efficient and practical solution. The main contributions of this article are as follows:

- Our proposed algorithm is applicable to off-the-shelf TCL appliances.
- We present a novel formulation of the desynchronization problem for a network of TCLs based on an objective function only indirectly related to decision-making and show how the Lagrange multipliers corresponding to our optimization problem can be used to make control decisions.
- The proposed algorithm ensures privacy, i.e., consumption and temperature data is not sent outside the devices.
- The approach is fully distributed and based only on local direct interactions between peer and anonymous appliances.

B. Structure of the paper

Section II reviews some fundamental concepts in multi-agent systems. The problem statement of this study is presented in Section III. Section IV details the proposed privacy-preserving, greedy, and fully distributed TCL-DSR algorithm. The effectiveness of our proposed scheme is showcased through simulation results in Section V. Concluding remarks are provided in Section VI.

II. Preliminaries

Consider a network composed of n TCLs that can interact. In the remainder of the manuscript, we refer to each TCL as an agent or node. Let $V = \{1, 2, \dots, n\}$ represent the set of agents in the network and $E \subseteq V \times V$ be the set of communication links (or edges) among the agents, i.e., if agent i sends information to j, then $(i,j) \in E$. Such a network is thus modeled as a graph $G = (V, E)$, including the sets of nodes V and edges E. We denote the set of in-neighbors of agent i as $N_i^{in} = \{j \in V : (j,i) \in E\}$. Similarly, the set of out-neighbors of agent i is denoted as $N_i^{out} = \{j \in V : (i,j) \in E\}$. A graph G is called *K-regular* if $\forall i \in V$, $|N_i^{out}| = |N_i^{in}| = K$, i.e. each agent has K in-neighbors and K out-neighbors. A graph G is defined *undirected* if the communication links are bidirectional, i.e., if $(i,j) \in E$ implies that $(j,i) \in E$, and is defined *directed* otherwise. A path $\mathcal{P}_{i,j}$ between nodes i and j is a sequence of consecutive edges, starting from node i and ending in node j, i.e., it is composed of the edges $\{(i, v_1), (v_1, v_2), \dots, (v_m, j)\} \subseteq E$, where $\{i, v_1, v_2, \dots, v_m, j\} \subseteq V$. A directed graph G is defined as strongly connected if there exists a directed path between each pair of nodes (i,j) in V. In case each edge $(j,i) \in E$ is associated with a positive weight, $a_{ij} > 0$, then the graph G is called weighted. The matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ collecting the weights is defined as *adjacency matrix*, i.e., $a_{ij} > 0$ if $(j,i) \in E$ and $a_{ij} = 0$ otherwise. A square matrix $A \in \mathbb{R}^{n \times n}$ with non-negative entries and with each row (column) summing to 1 is called row (column) stochastic. Moreover, A is called doubly stochastic if it is jointly row and column stochastic. Finally, if the edge weights $a_{ij}(k)$ are time-varying, then the weighted graph is time-varying as well and is denoted by $G(k) = (V, E(k))$. Let $E_B(k)$ be the aggregated set of edges $E(k)$ in the time interval $[k_0, k_0 + B)$, i.e., $E_B(k) = \bigcup_{k=0}^{B-1} E(k_0 + k)$. For $k_0 \in \mathbb{N}$, a time-varying graph $G(k)$ is defined as jointly strongly connected if there exists a finite positive integer B such that the graph $(V, E_B(k))$ is strongly connected for all finite k_0.

III. Problem Statement

Consider a network of TCLs interacting with each other. Assume that each agent i represents a TCL with its power outlet plugged into a Smart Power Socket (SPS) adaptor. These SPSs are equipped with processing units and communication

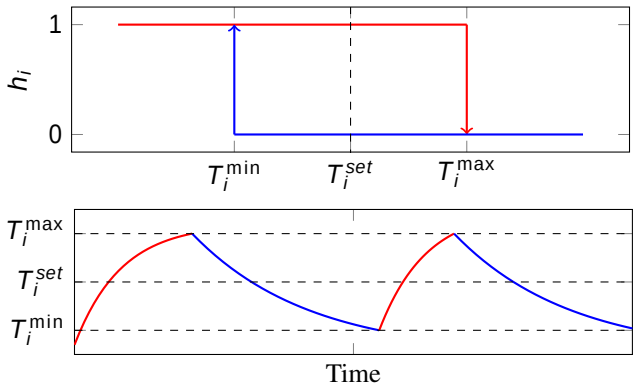

Fig. 1. Reverse hysteretic control of thermostats in TCLs (top), and cycles of thermostatically controlled temperatures within a bound centered on T_i^{set} (bottom). Red and blue lines denote ON and OFF conditions, respectively.

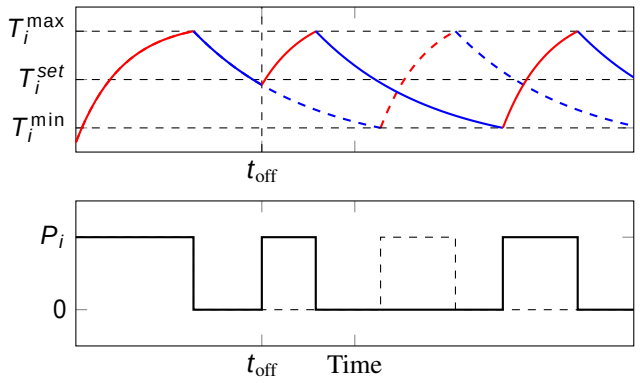

Fig. 2. Shifting the normal power consumption intervals of Type1 TCLs (shown by the dashed line) using control signal (3).

capabilities. The network topology at each time step $t_k \in \mathbb{R}^+$ can be described by the graph $G(t_k) = (V; E(t_k))$, where $E(t_k)$ fV Vg represents the set of edges that indicate interactions among agents.

Each SPS can monitor the active power $p_i(t_k) \in \mathbb{R}^+$ consumed by the associated TCL. Additionally, it can control the power supply to the TCL by manipulating its internal switch state $s_i(t_k) \in \{0; 1\}$. For instance, $s_i(t_k) = 1$ signifies that the i^{th} TCL is connected to the power supply. Conversely, $s_i(t_k) = 0$ indicates that the i^{th} TCL is disconnected, resulting in $p_i(t_k) = 0$. Let $P_i \in \mathbb{R}^+$ represent the nominal power of the TCL. Then, the absorbed power at time t_k 0 (with $k \in \mathbb{N}_0$) is approximated as:

$$p_i(t_k) = P_i \; s_i(t_k) \; h_i(t_k); \tag{1}$$

where $h_i(t_k)$ denotes the internal thermostat state of the i^{th} TCL. For water heaters and radiators, the thermostat state is typically updated using a *reverse hysteretic control* defined as follows:

$$h_i(t_{k+1}) := \begin{cases} 0 & \text{if } T_i(t_k) > T_i^{max}; \quad (2a) \\ 1 & \text{if } T_i(t_k) < T_i^{min}; \quad (2b) \\ h_i(t_k) & \text{otherwise}; \quad (2c) \end{cases}$$

where $T_i(t_k) \in \mathbb{R}$ is the temperature of TCL i, and T_i^{max} T_i^{set} $T_i^{min} > 0$ denote the hysteresis window, and T_i^{set} represents the adjusted setpoint of TCL i as set by its owner. On the other hand, in refrigerators or cold flow conditioners, the high and low conditions in (2a)-(2b) are reversed. Fig. 1 illustrates the hysteresis behavior of the thermostat in TCLs, as introduced in (2a)-(2c), along with cycles of thermostatically controlled temperatures within a range centered on T_i^{set} when no control action takes place (i.e., $s_i(t_k) = 1$ for all t_k 0). The initial thermostatically controlled temperature, $T_i(0)$, is lower than T_i^{min} in Fig. 1. Therefore, according to (2b) and since we assumed $s_i(0) = 1$, the TCL consumes its rated power, P_i, as per (1). The TCL will remain ON until it reaches the thermostat's upper limit, T_i^{max}. Subsequently, the TCL switches off until the temperature drops to T_i^{min}, and the process repeats.

If $T_i^{min} < T_i(0) < T_i^{max}$, the internal thermostat relay state of the TCL in the first time step after plug-in, denoted as $h_i(1)$, is equal to $h_i(0)$. Some TCLs have $h_i(0) = 1$, while others have $h_i(0) = 0$. We categorize these two types of TCLs as *Type1* (with $h_i(0) = 1$) and *Type2* (with $h_i(0) = 0$). This property enables a control room to adjust the power consumption intervals of TCLs by appropriately switching OFF and then ON the SPS, all while maintaining the thermostatically controlled temperature within the range $[T_i^{min}; T_i^{max}]$. For example, the control signal

$$s_i(t_k) = \begin{cases} 0 & \text{if } t_k = t_{off}; \\ 1 & \text{otherwise}. \end{cases} \tag{3}$$

could be applied to a *Type1* TCL. This control shifts the TCL's power consumption, as illustrated in Fig. 2.

For simplicity, let us consider the power consumption of two TCLs. As depicted in the left column of Fig. 3, an unfortunate coincidence can occur where these two TCLs run unnecessarily simultaneously, leading to an increased peak load. A high peak load is undesirable in power systems as it necessitates investments in infrastructure, contributes to environmental pollution by requiring the operation of inefficient backup power plants, and raises power costs in the electricity market, which negatively impacts both residential and industrial customers. Therefore, the preference is to *desynchronize* the power consumption of TCLs. This is illustrated in the second column of Fig. 3, where desynchronizing TCL loads reduces the peak load.

Now, let's discuss the minimum peak load without changing the QoS or, equivalently, without altering power consumption. Consider a sufficiently large time span denoted by K. Let's define the *duty cycle* of each TCL i as $dc_i \in [0; 1]$, representing its ON ratio over the time span K,

$$dc_i = \frac{\sum_{k \in K} s_i(t_k) \; h_i(t_k)}{K}. \tag{4}$$

Since time shifts generally do not change the duty cycle value, we choose not to restrict K to a specific time span. However, it should be sufficiently large to mitigate the effects of noise and disturbances that might alter the actual duty cycle of an appliance in short intervals. In this article, we assume that dc_i is constant in control. To find dc_i, the values

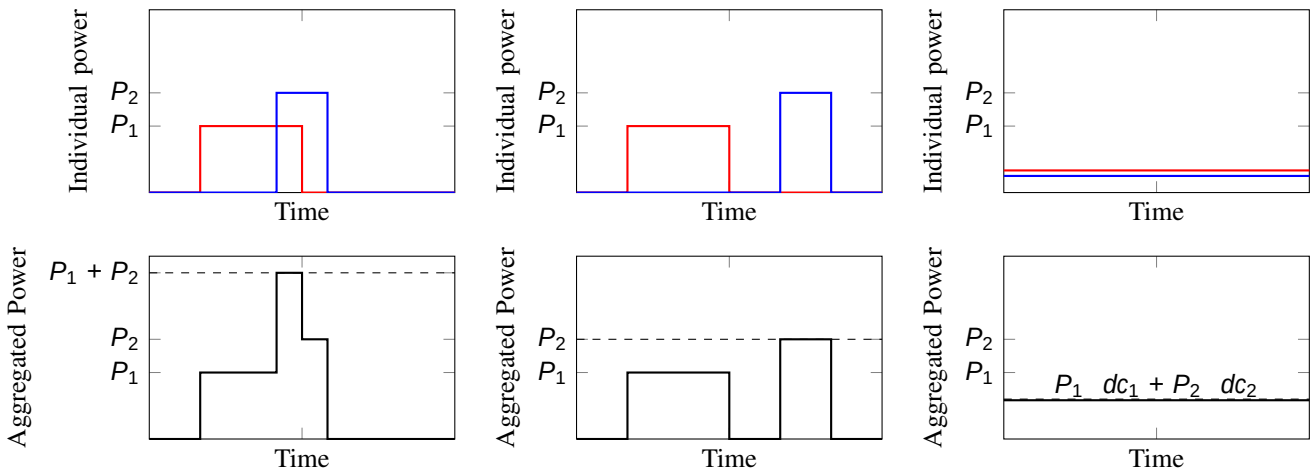

Fig. 3. Individual (top) and aggregated (bottom) power consumption of two TCLs. The left column represents the worst-case scenario where TCLs simultaneously consume energy within an interval. The second column depicts a favorable scenario in which the power consumption of TCLs does not overlap. The third column illustrates the best achievable scenario in which the power consumption of TCLs is spread out over the entire time domain.

of $s_i(t_k)$ and $h_i(t_k)$ should be measured over a sufficiently large time span K. These measurements are conducted over an uncontrolled TCL i, as the value of dc_i is necessary for executing our proposed control algorithm. It is worth mentioning that although we assume dc_i to be constant in short horizons, it could be sensible to update dc_i occasionally, e.g., every week or month.

From (1), the average power consumption of TCL i over time span K is

$$\frac{\sum_{k2K} p_i(t_k)}{K} = \frac{P_i \sum_{k2K} s_i(t_k) \quad h_i(t_k)}{K} \tag{5}$$

Therefore, according to (4), $P_i \quad dc_i$ represents the average power consumption of TCL i over the same time span, K.

As shown in the third column of Fig. 3 in a simple example of two TCLs, the lowest peak load involves spreading the power consumption across the entire time domain.

Let's define the total instantaneous absorbed power associated with the network of TCLs as

$$P^t(t_k) = \sum_{i=1}^{X} p_i(t_k): \tag{6}$$

In our problem, we assume that switching SPSs ON and OFF is the only control action. In other words, $s_i(t_k)$ is the control signal, and our goal is to maintain $P^t(t_k)$ close to the desired aggregated power, which leads to the lowest peak load, given by:

$$P_d^t = \sum_{i=1}^{X} dc_i \quad P_i; \tag{7}$$

while ensuring the privacy of customers.

IV. Privacy-preserving distributed greedy control framework

To obtain the necessary information for implementing our proposed algorithm, we first introduce a distributed optimization problem. The solution of this problem is not going to be used for resource allocation but rather to determine, in a distributed and privacy-preserving manner, whether the current accumulated power consumption of the TCLs' network is less than, or greater than P_d^t. Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{P} (x_i \quad p_i(t_k))^2; \\ \text{subject to:} \quad & \sum_{i=1}^{P} (x_i \quad dc_i \quad P_i) \quad 0: \end{aligned} \tag{8}$$

where x_i is a virtual decision variable. It's important to emphasize that x_i does not correspond to a physical parameter of the TCLs. Consequently, there is no need to impose boundaries on x_i. Problem (8) relates only indirectly to our main objective, which is finding an appropriate control signal $s_i(t_k)$, as it doesn't include $s_i(t_k)$, and in addition, it doesn't explicitly address the tracking of the desired aggregated power in (6). However, we demonstrate how one can determine whether the current accumulated power consumption of the TCLs' network is less than, or greater than $P_d^t$when applying primal-dual methods to solve (8). Consider the Lagrangian function as

$$\begin{aligned} (x; \ ) &= \sum_{i=1}^{X} {}_i(x_i; \ ) = \\ &\sum_{i=1}^{X} \Big\{ (x_i \quad p_i(t_k))^2 + \quad (x_i \quad dc_i \quad P_i) \Big\}; \end{aligned} \tag{9}$$

where $\quad 2 \ \mathbb{R}_+$ is the Lagrange multiplier. Therefore, the dual function is

$$'(\ ) = \min_{x2\mathbb{R}^n} f \ (x; \ )g: \tag{10}$$

where $x = [x_1; x_2; \dots; x_n]^T \ 2 \ \mathbb{R}^n$. Eq. (10) can be written as

$$'(\ ) = \sum_{i=1}^{X} '_i(\ ) = \sum_{i=1}^{X} \min_{x_i 2\mathbb{R}} f \ {}_i(x_i; \ )g \tag{11}$$

Therefore, the dual of (8) can be expressed as

$$\max_{2\mathbb{R}_+} \min_{x2\mathbb{R}^n} \quad (x; \ ) = \max_{2\mathbb{R}_+} \sum_{i=1}^{X} '_i(\ ); \tag{12}$$

In this paper, we utilize the constraint-coupled distributed optimization algorithm introduced in [20], which is based on dual decomposition, to solve (8). Through this approach, we determine whether the trivial optimal solution of (8), $x_i = p_i(t_k)$ for all $i \in V$, satisfies the coupling constraint. Notably, if the trivial solution meets the coupling constraint in (8), then by substituting x_i with $p_i(t_k)$ in the coupling constraint, $\sum_{i=1}^{} p_i(t_k) \sum_{i=1}^{} (dc_i \ P_i)$, and considering (6) and (7), it leads to the inequality $P^t(t_k) \ P_d^t$. Conversely, if $\sum_{i=1}^{} p_i(t_k) > \sum_{i=1}^{} (dc_i \ P_i)$, then the trivial optimal solution of (8), $x_i = p_i(t_k)$ for all $i \in V$, is no longer valid.

Let $_i(t_k)$ denote the Lagrangian multiplier associated with agent i at time step t_k. By implementing the algorithm introduced in [20], if $_i(t_k)$ converges to 0, agent i can infer that the coupling constraint is satisfied by the trivial optimal solution of (8), $x_i = p_i(t_k)$ for all $i \in V$, and consequently, $P^t(t_k) < P_d^t$. Otherwise, if $_i(t_k) > 0$, then agent i infers that the aggregated power consumption exceeds the threshold, i.e., $P^t(t_k) > P_d^t$. Thus, intuitively, each agent i takes a control action (switching on or off the SPS) based on the value of $_i(t_k)$. $_i(t_k) > 0$ indicates that the total aggregated power consumption exceeds the desired threshold, while $_i(t_k) = 0$ signifies that the total aggregated power consumption is at or below the desired threshold, as defined in (7).

Remark 1. *The convergence rate of the distributed resource allocation algorithm introduced in [20] is* $O(\log(k))$. *Several methods in the literature utilize primal-dual perturbation algorithms to solve* (8)*, such as [21]–[23], which may exhibit better convergence rates. However, we have selected [20] for our framework due to the following reasons:*

- *In both [21], [22], information related to the primal problem is exchanged among agents, potentially raising privacy concerns.*
- *In [21], [22], each agent needs to be aware of the coupling constraint within the primal problem.*
- *The application of algorithms such as [21]–[23] requires the existence of a known Slater point known to all agents. In contrast, in [20], only the existence of such a point is necessary.*
- *While the communication network is assumed to be time-invariant in [23], it can be time-varying in [20].*

The distributed resource allocation solver in [20] requires convexity and zero duality gap, which are satisfied in (8), and the graph representing the network of TCLs should satisfy the following necessary assumptions:

Assumption 1. *There exists a constant* $0 < \ < 1$ *such that for all* $i; j \in V$ *and* $k \ \ 0$, $a_{ii}(k) \ \ $ *and either* $a_{ij}(k) = 0$ *or* $a_{ij}(k) \in [\ ; 1)$.

Assumption 2. *The graph representing the network of TCLs is jointly strongly connected and its adjacency matrix is doubly stochastic.*

Algorithm 1 presents our distributed framework for desynchronizing the aggregated power consumption of the network. First, (8) is solved. Since convergence to the consensus value of is asymptotic, we choose a sufficiently small threshold, denoted as > 0, and consider values less than this threshold as zero. As discussed earlier, the consensus value of $_i(t_k)$ allows us to determine whether $P^t(t_k) < P_d^t$ or $P^t(t_k) > P_d^t$.

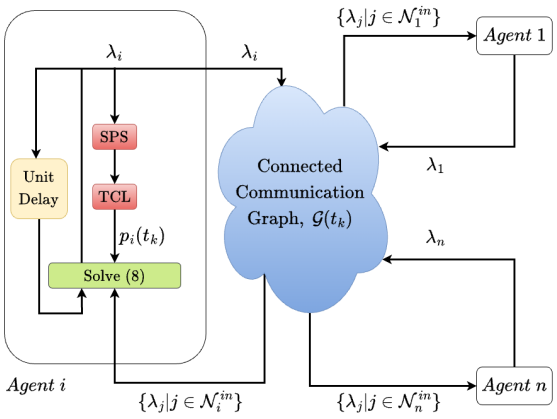

Fig. 4. Block diagram of the network connecting TCLs to implement Algorithm 1.

Next, appropriate control actions are applied. $_i(t_k) < \ $, implies $P^t(t_k) < P_d^t$. Therefore, if $h_i(t_k) = 0$ and $_i(t_k) < \ $, then $s_i(t_k^+ : t_k + \ _i) = 0$. This indicates that the SPS disconnects the power for $_i$ seconds in Type 1 TCLs. Here, $_i$ represents the required time for resetting the internal thermostat. This control action turns ON the Type1 TCLs if $T_i(t_k)$ falls within the range $[T_i^{\min}; T_i^{\max}]$.

Elseif $_i(t_k) > \ $ and $h_i(t_k) = 1$, the SPS disconnects the power of Type2 TCLs for a duration of $_i$ seconds, as specified in lines 5 to 7 of Algorithm 1. This action turns OFF the TCLs if $T_i(t_k)$ falls within the range $[T_i^{\min}; T_i^{\max}]$. The block diagram depicting the network connecting TCLs to implement Algorithm 1 is shown in Fig. 4.

Algorithm 1 (Asynchronously implemented within each TCL $i \in V$)

1: **Solve** (8) **and find** $_i(t_k)$
2: **if** $i \in$ Type1 TCLs **then**
3: **if** $h_i(t_k) = 0$ **AND** $_i(t_k) <$ **then**
4: $s_i(t_k^+ : t_k + \ _i) = 0$
5: **if** $i \in$ Type2 TCLs **then**
6: **if** $h_i(t_k) = 1$ **AND** $_i(t_k)$ **then**
7: $s_i(t_k^+ : t_k + \ _i) = 0$

Remark 2. *It's important to note that solving* (8) *using the algorithm introduced in [20] only requires agents to share their estimates of Lagrangian multipliers. The Lagrangian multiplier does not encompass any private details about local states, power consumption, or appliance temperatures. As a result, customer privacy is preserved during the execution of Algorithm 1. Interested readers can refer to [20] for an in-depth privacy analysis of the distributed resource allocation solver selected in this article. .*

Remark 3. *While security concerns extend beyond the focus of this work, it's important to acknowledge that adversaries or*

system failures can potentially disrupt distributed optimization algorithms [24]. It is noteworthy that Algorithm 1 can enhance resilience against cyberattacks or failures affecting a subset of TCLs, particularly when a resilient distributed resource allocation algorithm, such as the algorithms proposed in [25], [26], are employed to solve (8).

In this step, let's discuss and compare the communication burden and scalability of Algorithm 1 with approaches like [9], which require a central server to aggregate agent data. In Algorithm 1, solving (8) to a certain degree using a distributed resource allocation algorithm is necessary, and this process inherently demands communication among agents over iterations.

Let m_{t_k} represent the count of iterations needed by the distributed resource allocation algorithm to address (8) within the interval $(t_{k-1}; t_k)$. Accordingly, the cumulative number of transmitted data packets for control at t_k would be $m_{t_k} \sum_{i=1}^{n} |N_i^{out}|$. In contrast, star networks generally entail agents only sending information to the central server and receiving control decisions from it. Hence, the total transmitted data packets in a single control step amount to $2n$. Recall that for a graph representing the network, strong connectivity is a requirement. Consequently, $|N_i^{out}| \geq 1$ holds for all $i \in \mathcal{V}$. Hence, $m_{t_k} > 2$ indicates that executing Algorithm 1 leads to a greater overall package transfer compared to typical server-based methods. However, the advantage of Algorithm 1 lies in the fact that communication is distributed across all agents within the network. In Algorithm 1, each agent i is responsible for compiling $|N_i^{in}|$ data packages per iteration, irrespective of the total number of agents, denoted by n. Conversely, in a star-graph topology, a server must receive and analyze n data packages, which could potentially pose scalability challenges and result in a substantial communication load concentrated at a single point.

Subsequently, we present a formal proof that the execution of Algorithm 1 effectively aids in desynchronizing the power consumption of appliances.

Theorem 1. *Let Assumptions 1 and 2 hold. Let us define the aggregated absolute desynchronization error by*

$$e(K) = \sum_{t_k=0}^{K} |P^{\mathrm{t}}(t_k) - P_d^{\mathrm{t}}| \tag{13}$$

and assume that $e_c(K)$ and $e_u(K)$ represent the desynchronization error in a network controlled by Algorithm 1 and an uncontrolled network, respectively. Then,

$$e_c(K) \leq e_u(K).$$

Proof. As previously discussed, γ_i converges to 0 for all $i \in \mathcal{V}$ when $P^{\mathrm{t}}(t_k) - P_d^{\mathrm{t}} \leq 0$, and $\gamma_i > 0$ for all $i \in \mathcal{V}$ when $P^{\mathrm{t}}(t_k) - P_d^{\mathrm{t}} > 0$. According to line 3 of Algorithm 1, the control action and activation of the TCLs occur when $\alpha_i(t_k) < \gamma_i$, which is equivalent to $P^{\mathrm{t}}(t_k) - P_d^{\mathrm{t}} < 0$.

Applying Algorithm 1 does not alter the network's energy demand. Consequently, if Algorithm 1 turns on a TCL during a period when $P^{\mathrm{t}}(t_k) - P_d^{\mathrm{t}} < 0$, the corresponding energy consumption must be offset by reductions from other time steps, such as t_r. When $P^{\mathrm{t}}(t_r) - P_d^{\mathrm{t}} < 0$, it does not alter the desynchronization error. However, if $P^{\mathrm{t}}(t_r) - P_d^{\mathrm{t}} > 0$, the desynchronization error is decreased. This results in:

$$e_c(K) \leq e_u(K).$$

A similar argument applies when $P^{\mathrm{t}}(t_r) - P_d^{\mathrm{t}} < 0$. □

TABLE I
WATER HEATER MODEL PARAMETERS

	Water density	1	[kg/dm^3]
c_p	Water specific heat	4186	[J/(C kg)]
R_i	Thermal resistance	0.0488	[m^{2}C /W]
S_i	Tank surface	0.536	[m^2]
V_i	Tank volume	100	[dm^3]
P_i	Heater power	1500	[W]

V. NUMERICAL SIMULATION

To evaluate Algorithm 1, we consider a network of $n = 1000$ water heaters, whose temperatures $T_i(t_k)$ evolve according to the model described in [27].

$$T_i(t_{k+1}) = A_k T_i(t_k) + B_k \left(\eta_i T_i^{\mathrm{r}} + \eta_i(t_k) T_i^{\mathrm{in}} + \eta_i h_i(t_k) s_i(t_k) \right),$$

$$A_k = \mathrm{e}^{-(\eta_i + \eta_i(t_k))\,t}, B_k = \frac{1 - \mathrm{e}^{-(\eta_i + \eta_i(t_k))\,t}}{\eta_i + \eta_i(t_k)}, \tag{14}$$

$$\eta_i = \frac{S_i}{c_p R_i V_i}, \quad \eta_i(t_k) = \frac{w_i(t_k)}{V_i}, \quad \eta_i = \frac{c_p P_i}{V_i}, \tag{15}$$

where $s_i(t_k)$ is the control input designed to desynchronize the power consumption of TCLs while ensuring that $T_i(t_k)$ remains within the range of $T_i^{\min} = 50$ C and $T_i^{\max} = 60$ C. $T_i^r = 20$ C represents the room temperature, and $w_i(t_k)$ accounts for an unknown disturbance simulating the cold water refill process within each water heater following water withdrawal. Additionally, $T_i^{\mathrm{in}} = 15$ C stands for the inlet cold water temperature. We assume that the graph representing the communication network of water heaters is *directed and randomly generated* 100*-regular*.

For our simulation, we assume that all water heaters are identical, sharing the model parameters outlined in Table I. At the beginning of the simulation, the initial temperatures TCLs are randomly chosen from the interval $[T_i^{\min}; T_i^{\max}]$, with half of them having an initial thermostat state of $h_i(0) = 1$.

The disturbance $w_i(t_k)$ is modeled as a specially crafted stochastic process to replicate increased hot water demand during peak hours, as depicted in Fig.5. Lastly, we model calls to Algorithm 1 as a Poisson point process with an average rate of 20 calls per second. These calls are evenly distributed between Type1 and Type2 TCLs.

We compare our simulation results with the priority-based control algorithm introduced in [9] to evaluate our proposed algorithm. The algorithm proposed in [9] shares similarities with our approach as it is also model-free and aims to desynchronize the power consumption of TCLs in a network, same as the objective of Algorithm 1. However, in the priority-based control method outlined in [9], all water heaters' energy

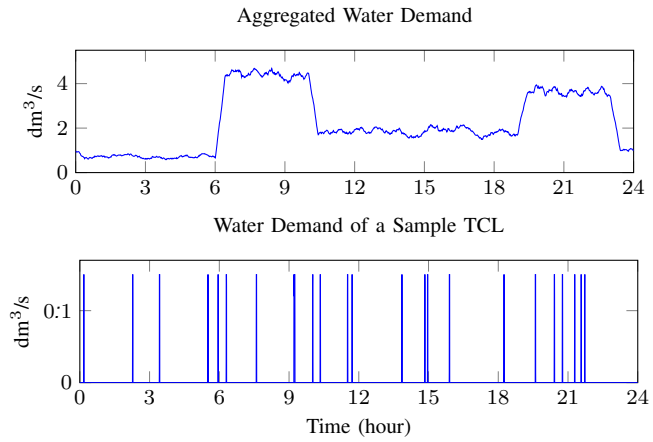

Fig. 5. Top: Aggregated daily hot water demand for the considered network of water heaters. Bottom: Daily water demand for a generic water heater.

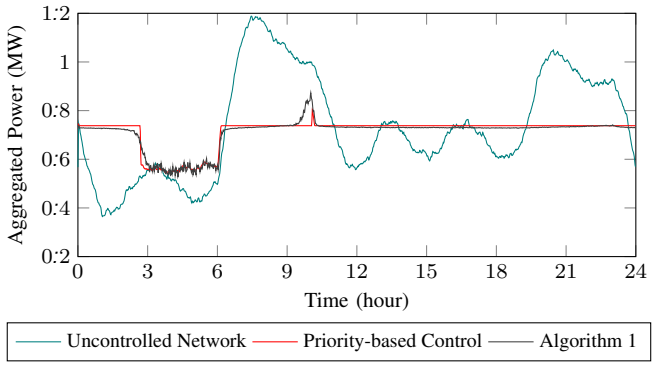

Fig. 6. Aggregated power consumption of TCLs' network by implementing Algorithm 1, priority-based control [9], and uncontrolled network.

consumption must be transmitted to a central server at each time step. Consequently, the server can readily determine whether the aggregated network power consumption surpasses the average at any given time step. This ease of access enables the server to apply appropriate control actions effectively. In contrast, our proposed algorithm equips water heaters with data from only a limited subset of neighboring appliances as depicted in Fig. 4. Furthermore, these water heaters do not receive private information concerning their neighbors' power consumption. As a result, an individual water heater cannot directly ascertain the aggregated energy consumption of the TCL network.

Given these explanations, we do not anticipate achieving superior results compared to the priority-based control method. Our objective, instead, is to present relatively comparable results while incorporating scalability, privacy preservation, a fully distributed architecture, and compatibility with existing appliances. In Fig. 6, a comparison is presented between the aggregated power consumption of the uncontrolled TCL network, the TCL network controlled by the priority-based method, and the network implementing Algorithm 1.

As depicted in Fig. 6, during the time frame from 3 am to 6 am, the aggregated power consumption remains below the average for both the priority-based control and Algorithm 1. This outcome arises due to the preservation of customers' authority to set the temperature of their respective appliances. Consequently, water heaters avoid increased consumption to avoid violating the upper temperature limit $T_i^{\max}$ during off-peak hours.

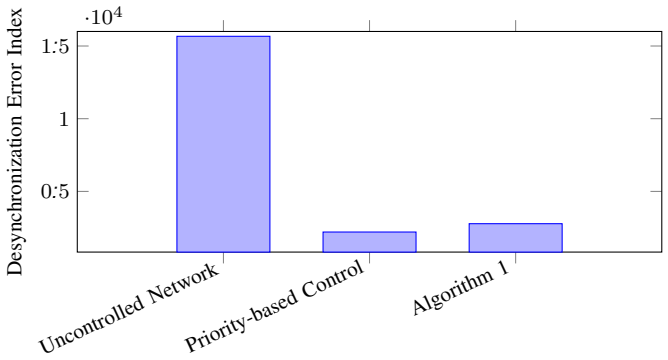

Fig. 7. Comparison of the aggregated absolute desynchronization error index defined in (13) by implementing Algorithm 1, priority-based control [9], and uncontrolled network.

Fig. 7 compares the aggregated absolute desynchronization error index, as defined in (13). It is evident that the implementation of Algorithm 1 leads to a substantial reduction in the error index compared to the uncontrolled scenario, and the reduction is relatively similar to that achieved by the priority-based control method.

VI. Conclusions

This paper introduces a novel model-free, privacy-preserving, asynchronous greedy control framework designed to desynchronize the power consumption of TCLs within a fully distributed architecture. In our framework, agents exclusively share their Lagrange multiplier estimates, preserving the confidentiality of agents' private data regarding power consumption and temperature. We establish a distributed optimization problem where the consensus value of the Lagrange multiplier, once solved, provides agents with meaningful information to guide appropriate control actions. In the simulation section, we benchmark our proposed method against a priority-based control approach, feasible only within a star architecture requiring agents to transmit private data to a central server. Our simulation results showcase the potential of achieving comparable outcomes while incorporating critical attributes such as scalability, privacy preservation, a fully distributed architecture, and adaptability to off-the-shelf appliances.

References

[1] J. Knudsen, J. Hansen, and A. M. Annaswamy, "A dynamic market mechanism for the integration of renewables and demand response," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 940–955, 2016.

[2] K. Alshehri, J. Liu, X. Chen, and T. Başar, "A game-theoretic framework for multiperiod-multicompany demand response management in the smart grid," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 1019–1034, 2021.

[3] H. C. Gils, "Economic potential for future demand response in germany – modeling approach and case study," *Appl. Energy*, vol. 162, pp. 401–415, 2016.

[4] V. Trovato, I. M. Sanz, B. Chaudhuri, and G. Strbac, "Advanced control of thermostatic loads for rapid frequency response in great britain," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2106–2117, 2017.