Distributed Real-Time Energy Scheduling in Smart Grid: Stochastic Model and Fast Optimization

Chen Gong, Xiaodong Wang, Weiqiang Xu, and Ali Tajer

***Abstract*—We develop a stochastic energy scheduling model for a local-area smart-grid system with a single energy source and multiple energy consumers. The tasks of the energy consumers are classified into two categories, namely, the stochastic background tasks and the deterministic dynamic tasks. The objective is to schedule the energy consumptions of the dynamic tasks to maximize the expected system utility under the given energy consumption and energy generation constraints. To make this problem tractable, using rolling horizon optimization and Gaussian approximation we transform the original stochastic optimization problem into a convex optimization problem with linear constraints. We then derive a distributed Newton's method to solve this problem, and design a message-passing mechanism for a distributed implementation of the algorithm with limited information exchange between the energy consumers and the energy source. In simulations, the proposed distributed Newton's method converges for the system under consideration, while the traditional dual decomposition method does not converge to a primary feasible solution; and thus it is a powerful practical tool for real-time control of smart-grid systems.**

***Index Terms*—Smart grid, energy scheduling, stochastic model, distributed Newton's method.**

I. Introduction

ENERGY scheduling [1]–[4] is one of the key enabling techniques for smart-grid systems by maximizing the total system utility of an electrical power network under various energy usage constraints. Recent research on energy scheduling can be classified into two types, residential energy scheduling [5]–[8] and local-area energy scheduling [9], [10]. For residential energy scheduling, a residential controller schedules the energy consumption of various devices according to the tasks to be fulfilled and the price of electricity at different times. For local-area energy scheduling, local-area energy controllers schedule the energy generation by the sources and the energy consumption of the consumers, to maximize the system utility, usually in a distributive manner.

Manuscript received February 13, 2012; revised May 17, 2012, September 05, 2012; accepted January 01, 2013. Date of publication April 11, 2013; date of current version August 21, 2013. This work was supported in part by the U.S. National Science Foundation under Grant CIF1064575, and in part by the U.S. Office of Naval Research under Grant N000141210043. Paper no. TSG-00063-2012.

C. Gong is with Qualcomm Research San Diego, Qualcomm Inc., San Diego, CA 92121 USA.

X. Wang is with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA, and also with King Abdulaziz University, Jeddah, Saudi Arabia.

W. Xu is with the Department of Electrical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China.

A. Tajer is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2013.2248399

On the other hand, in existing works the system models can be classified as either deterministic [5], [6], [9] or stochastic [7], [10]. Under deterministic models, energy scheduling is formulated as a standard optimization problem with a convex objective function and several linear constraints, which is then solved using either a centralized or a distributed method. Under stochastic models, the energy scheduling problem is typically formulated as a Markov decision process (MDP) [7], [10], which is then solved using either an online Q-learning method or an exhaustive search by enumerating all possible realizations. However, neither method is practical for real-time control since the Q-learning method takes a long time to converge and the exhaustive search has a prohibitively high computational complexity.

In this paper, we consider the real-time local-area energy scheduling for a local-area power network with a single energy source and multiple energy consumers, and with both stochastic and deterministic energy demands. Different from the methods used in [10], in this work, to reduce the complexity of system optimization, we convert the stochastic constraints to linear constraints using Gaussian approximations. To further make the solution tracktable, we employ the rolling horizon optimization [11] to reduce the problem sizes. More specifically, at each time slot we optimize the energy scheduling in a short time window ahead starting from the current time slot, resulting in an online energy scheduler. Moreover, to meet the real-time requirement, we develop a fast distributed solution based on the distributed Newton's method which converges quadratically to a neighborhood of the optimal solution, while the classical dual-decomposition method [12] does not converge.

The remainder of this paper is organized as follows. In Section II, we develop a stochastic model for energy scheduling and formulate the optimization problem. In Section III, we transform the energy scheduling problem into a convex optimization problem with linear constraints using rolling horizon optimization and Gaussian approximation. In Section V, we introduce the distributed Newton's method, and show that it can be applied to any convex optimization problem with linear constraints. In Section V, we develop a distributed Newton's algorithm for solving the energy scheduling problem. Numerical results are given in Section VI; and finally Section VII concludes this paper.

II. System Model and Problem Formulation

Consider a smart-grid system with a single energy source and N energy consumers. The time cycle is divided into H time slots $1, 2, \ldots, H$. Assume that there are energy controllers which control the energy consumption of the energy consumers and the energy generation of the source. There are communication links between the energy consumers and the energy source, which enable message passing for the distributed network optimization.

A. Energy Consumer Task Models

The tasks of the energy consumers are classified into two types, the background tasks and the dynamic tasks. Background tasks are running in an "on-off" manner, to maintain the basic system operating conditions during the entire time cycle, e.g., the lighting and monitoring systems. Dynamic tasks are those which are created at certain times and become inactive after the tasks are completed, e.g., cooking meals and washing clothes. In the following we elaborate on the energy consumption models for the background and dynamic tasks.

1) Background Tasks: Let $\mathcal{B}_i$, $1 \leq i \leq N$, be the set of background tasks of energy consumer i, and z_{ij}^{τ} be the energy consumption of task $j \in \mathcal{B}_i$ in time slot τ.

Assume that there are two states, on and off, for each background task. For task $j \in \mathcal{B}_i$, if in time slot τ it is on, then in the next slot $\tau+1$ it keeps on with probability $q_{ij}^{\text{on}}(\tau)$ and turns off with probability $1 - q_{ij}^{\text{on}}(\tau)$; and if in time slot τ it is off, then in the next slot $\tau+1$ it keeps off with probability $q_{ij}^{\text{off}}(\tau)$ and turns on with probability $1 - q_{ij}^{\text{off}}(\tau)$. Note that the on-off transition probabilities $q_{ij}^{\text{on}}(\tau)$ and $q_{ij}^{\text{off}}(\tau)$ are functions of the current time slot τ. Assume that, when a task $j \in \mathcal{B}_i$ is on, it consumes a fixed amount of energy B_{ij}; otherwise it consumes zero energy.

Here we assume that all transition probability belongs to an interval $(\delta, 1-\delta)$ for some $\delta > 0$, i.e., $q_{ij}^{\text{on}}(\tau), q_{ij}^{\text{off}}(\tau) \in (\delta, 1-\delta)$ for all $1 \leq i \leq N$, $j \in \mathcal{B}_i$, and $1 \leq \tau \leq H$. Assume that at the first time slot, the probability that a background task is on falls between the interval $(\delta, 1-\delta)$. Assume that for all background tasks $(i,j)_{\mathcal{B}}$, the energy consumption $B_{\min} \leq B_{ij} \leq B_{\max}$.

We model the transition probabilities $q_{ij}^{\text{on}}(\tau)$ and $q_{ij}^{\text{off}}(\tau)$ as time-varying variables. Motivated by the fact that they change very insignificantly within a short time interval we divide the entire time cycle into several smaller time intervals, and assume that the parameters $q_{ij}^{\text{on}}(\tau)$ and $q_{ij}^{\text{off}}(\tau)$ remain constant within each time interval. We can estimate the values of $q_{ij}^{\text{on}}(\tau)$ and $q_{ij}^{\text{off}}(\tau)$ based on the observed samples of the background tasks. A simple estimation method is to record the numbers of on-off state transitions, and use the transition frequencies as estimators of the transmission probabilities.

2) Dynamic Tasks: We classify the dynamic tasks into two types, types-1 and type-2 dynamic tasks. For a type-1 dynamic task, it has no total energy consumption constraint, and the utility function is a function of its total energy consumption. For a type-2 dynamic task, its total energy consumption equals to some fixed amount, without utility function. For each dynamic task, we define an associated active interval specified by a starting time slot and an ending time slot, and schedule its energy consumption in each slot within that time interval [13]. We say that a dynamic task is active within that time slot. The energy scheduling is deterministic and no stochastic energy consumption is involved.

For type-1 dynamic tasks, let $\mathcal{D}_i$ be the set of dynamic tasks of consumer i for $1 \leq i \leq N$. Each consumer i dynamically adds tasks into the set $\mathcal{D}_i$ over time. A dynamic task $j \in \mathcal{D}_i$ is specified by three parameters $(T_{ij}^{s1}, T_{ij}^{e1}, U_{ij}(x))$, where T_{ij}^{s1} and T_{ij}^{e1} are the starting and ending time slots, respectively, and $U_{ij}(x)$ is the utility function. Let $\mathcal{D}_i$ be the set of type-1 dynamic tasks of consumer i, and x_{ij}^{τ} be the energy consumption of dynamic task $j \in \mathcal{D}_i$ at slot τ. We have that its utility is given by

$$U_{ij}\left(\sum_{\tau=T_{ij}^{s1}}^{T_{ij}^{e1}} x_{ij}^{\tau}\right). \tag{1}$$

Assume that the consumer utility of each user satisfies the *linear decreasing marginal benefit condition* [9]. Then the utility function $U_{ij}(x)$ is quadratic, given by

$$U_{ij}(x) = \begin{cases} 2b_{ij}x - a_{ij}x^2, & \text{if} \quad 0 \leq x \leq \frac{b_{ij}}{a_{ij}}, \\ \frac{b_{ij}^2}{a_{ij}}, & \text{if} \quad x \geq \frac{b_{ij}}{a_{ij}}, \end{cases} \tag{2}$$

where b_{ij} and a_{ij} are the coefficients of the utility function. Assume that the energy consumption x_{ij}^{τ} for type-1 dynamic task $j \in \mathcal{D}_i$ is upper bounded by $x_{ij}^{\tau} \leq X_{ij}$.

For type-2 dynamic tasks, let $\mathcal{F}_i$ be the set of dynamic tasks of consumer i for $1 \leq i \leq N$. Each consumer i dynamically adds tasks into the set $\mathcal{F}_i$ over time. A dynamic task $j \in \mathcal{F}_i$ is specified by three parameters $(T_{ij}^{s2}, T_{ij}^{e2}, F_{ij})$, where T_{ij}^{s2} and T_{ij}^{e2} are the starting and ending time slots, respectively, and F_{ij} is the total required energy consumption. Let $\mathcal{F}_i^{\tau}$ be the set of type-2 dynamic tasks of consumer i active at time τ, and y_{ij}^{τ} be the energy consumption of dynamic task $j \in \mathcal{F}_i$ at time τ. We have that its energy consumption must satisfy

$$\sum_{\tau=T_{ij}^{s2}}^{T_{ij}^{e2}} y_{ij}^{\tau} = F_{ij}. \tag{3}$$

Also, assume that the energy consumption y_{ij}^{τ} for type-2 dynamic task $j \in \mathcal{F}_i$ is upper bounded by $y_{ij}^{\tau} \leq Y_{ij}$.

In the remainder of this paper we denote each background task $j \in \mathcal{B}_i$ as $(i,j)_{\mathcal{B}}$, the type-1 dynamic task $j \in \mathcal{D}_i$ as $(i,j)_{\mathcal{D}}$, and the type-2 dynamic task $j \in \mathcal{F}_i$ as $(i,j)_{\mathcal{F}}$. In this work, assuming that *the background tasks are stochastic*, we *schedule the energy consumption of the two types of dynamic tasks*, to maximize the expected system utility.

B. System Outage

A system outage occurs when the total energy consumption of all consumers, which is also the energy generated by the energy source, exceeds the maximum value.

Let G^m be the maximum value of the energy generated by the source. Let

$$g^{\tau} \triangleq \sum_{i=1}^{N} \sum_{j \in \mathcal{D}_i^{\tau}} x_{ij}^{\tau} + \sum_{i=1}^{N} \sum_{j \in \mathcal{F}_i^{\tau}} y_{ij}^{\tau} + \sum_{i=1}^{N} \sum_{j \in \mathcal{B}_i} z_{ij}^{\tau} \tag{4}$$

be the energy consumption of all consumers which is also the energy generated by the source in time slot τ. In each slot τ, the system outage occurs if $g^{\tau} > G^m$.

C. Optimization Problem Formulation

Our objective is to maximize the total system utility which is defined as the expected sum utilities of dynamic tasks minus the expected cost of the energy source, subject to the system outage probability constraint. The energy source cost, denoted as $C(g)$, should be strictly increasing; and the marginal cost per amount of energy generation should also be increasing, which indicates that $C(g)$ is convex [9]. Assuming a *linearly increasing marginal cost* of the energy source, the energy source cost function is then given by [9]

$$C(g) = c_1 g + c_2 g^2. \tag{5}$$

Let $\mathbb{P}(\mathcal{A})$ denote the probability of the event $\mathcal{A}$, and $\mathbb{E}(\cdot)$ denote the expectation. Let ϵ be the upper bound on the allowable system outage probability. The system optimization problem can be formulated as follows.

$$\max_{\{x_{ij}^\tau, y_{ij}^\tau, g^\tau\}} \sum_{\tau=1}^{H}\sum_{i=1}^{N} U_{ij}\left(\sum_{\tau=T_{ij}^{s1}}^{T_{ij}^{e1}} x_{ij}^\tau\right) - \mathbb{E}\left(\sum_{\tau=1}^{H} C(g^\tau)\right)$$

$$\begin{aligned}
\text{s.t.}\quad & \sum_{\tau=T_{ij}^{s2}}^{T_{ij}^{e2}} y_{ij}^\tau = F_{ij},\ 1 \le i \le N,\ j \in \mathcal{F}_i;\\
& 0 \le x_{ij}^\tau \le X_{ij},\ 1 \le i \le N,\ j \in \mathcal{D}_i,\ 1 \le \tau \le H;\\
& 0 \le y_{ij}^\tau \le Y_{ij},\ 1 \le i \le N,\ j \in \mathcal{F}_i,\ 1 \le \tau \le H;\\
& g^\tau = \sum_{i=1}^{N}\sum_{j\in\mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^{N}\sum_{j\in\mathcal{F}_i} y_{ij}^\tau + \sum_{i=1}^{N}\sum_{j\in\mathcal{B}_i} z_{ij}^\tau,\\
& \quad 1 \le \tau \le H;\\
& \mathbb{P}(g^\tau > G^m) < \epsilon,\ 1 \le \tau \le H.
\end{aligned} \tag{6}$$

The above optimization problem is intractable due to the huge problem size and the probabilistic constraints. Moreover, since the dynamic tasks are generated during the system running process, the associated linear constraints are unknown at time slot $\tau = 1$, which makes the system optimization infeasible at time slot $\tau = 1$. A solution to this issue is to break the entire optimization problem into smaller ones for each time slot, where the optimization for each slot is within a small window ahead of that slot. This results in reducing the impact of the uncertainties due unknown linear constraints on system optimization. Following the above idea, in the next section we transform the optimization problem into a tractable form for which an efficient solution can be devised.

III. Rolling Horizon Optimization and Gaussian Approximation

Using rolling horizon optimization [11], we transform the optimization problem into a tractable form and reduce the problem size, and using Gaussian approximation, we transform the probabilistic constraints to linear constraints.

A. Rolling Horizon Optimization

Rolling horizon optimization [11] is a decomposition method for a large scale optimization problem, which divides the problem into various small windows and transforms the original objective function and constraints into those within the windows. Note that the rolling horizon optimization usually changes the original optimization problem, and the optimal solution to the decomposed optimization problem is not the optimal solution to the original problem. However, the main advantage of rolling horizon optimization is lower computational complexity. In this work, besides the lower complexity, another reason for the rolling horizon optimization is that the objective function and the constraints are unknown at the start of the entire time slots, and thus solving the entire scheduling problem is not feasible. Furthermore, the prediction of the objective function and constraints in a time slot is accurate only several slots after that time slot. Thus, the energy scheduling is accurate only within a small window after a given time slot. In this work, we employ rolling horizon optimization to break down the original optimization problem into smaller scale problems.

Instead of performing energy scheduling for the entire time cycle in the beginning, at each time slot T we schedule the current dynamic tasks in a time window $[T, T+\Delta T]$, i.e., for time slots $T \le \tau \le T+\Delta T$, where the window length $\Delta T + 1$ is usually small, e.g., $\Delta T + 1 = 3$, such that the sizes of the subproblems are small. For $T \le \tau \le T+\Delta T$, recall that $\mathcal{D}_i^\tau$ and $\mathcal{F}_i^\tau$ denote the set of type-1 and type-2 dynamic tasks that are created before or at time slot T and still active in slot τ, respectively. Note that $\cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^\tau$ and $\cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^\tau$ are the set of active type-1 and type-2 tasks during $[T, T+\Delta T]$.

We next address how to transform the energy constraint for the entire time cycle to that for each window $[T, T+\Delta T]$, for both types of tasks. For each type-1 dynamic task, let $\mathcal{T}_{ij}^1(T) \triangleq [T, T+\Delta T] \cap [T_{ij}^{s1}, T_{ij}^{e1}]$ and $\tilde{\mathcal{T}}_{ij}^1(T) \triangleq [T, T_{ij}^{e1}]$ be its active slots in $[T, T+\Delta T]$ and the total remaining time slots, respectively. Since the energy consumptions x_{ij}^τ for $T_{ij}^{s1} \le \tau \le T-1$ are already known, we have that the total energy consumption of task $(i,j)_{\mathcal{D}}$, denoted as e_{ij}, satisfies

$$e_{ij} = \sum_{\tau\in\tilde{\mathcal{T}}_{ij}^1(T)} x_{ij}^\tau + \sum_{\tau=T_{ij}^{s1}}^{T-1} x_{ij}^\tau. \tag{7}$$

Note that e_{ij} may involve the energy consumption x_{ij}^τ for $\tau \ge T + \Delta T$ and thus need prediction. We employ a simple linear prediction $\hat{e}_{ij}$ for e_{ij} as follows:

$$\begin{aligned}
\hat{e}_{ij} &= \frac{\left|\tilde{\mathcal{T}}_{ij}^1(T)\right|}{\left|\mathcal{T}_{ij}^1(T)\right|} \sum_{\tau\in\mathcal{T}_{ij}^1(T)} x_{ij}^\tau + \sum_{\tau=T_{ij}^{s1}}^{T-1} x_{ij}^\tau\\
&\triangleq \alpha_{ij} \sum_{\tau\in\mathcal{T}_{ij}^1(T)} x_{ij}^\tau + \sum_{\tau=T_{ij}^{s1}}^{T-1} x_{ij}^\tau.
\end{aligned} \tag{8}$$

Let $\tilde{X}_{ij} \triangleq \sum_{\tau=T_{ij}^{s1}}^{T-1} x_{ij}^\tau$ and $s_{ij} \triangleq \sum_{\tau\in\mathcal{T}_{ij}^1(T)} x_{ij}^\tau$ be the total energy consumption before time slot T and in the slot window $[T, T+\Delta T]$, respectively. We have that

$$\hat{e}_{ij} = \alpha_{ij} s_{ij} + \tilde{X}_{ij}. \tag{9}$$

For each type-2 task, similarly we let $\mathcal{T}_{ij}^2(T) \triangleq [T, T+\Delta T] \cap [T_{ij}^{s2}, T_{ij}^{e2}]$ and $\tilde{\mathcal{T}}_{ij}^2(T) \triangleq [T, T_{ij}^{e2}]$ be its active slots in $[T, T +$

$\Delta T]$ and the total remaining time slots, respectively. We also use linear prediction to provide a linear constraint for y_{ij}^τ as follows:

$$\sum_{\tau \in \mathcal{T}_{ij}^2(T)} y_{ij}^\tau = \frac{\left|\mathcal{T}_{ij}^2(T)\right|}{\left|\tilde{\mathcal{T}}_{ij}^2(T)\right|} \left(F_{ij} - \sum_{\tau=T_{ij}^{s2}}^{T-1} y_{ij}^\tau \right) \triangleq \tilde{F}_{ij}(T). \tag{10}$$

Then, letting $U_{ij}^T(s_{ij}) = U_{ij}(\alpha_{ij}s_{ij} + \tilde{X}_{ij})$, the energy scheduling optimization problem during the window $[T, T+\Delta T]$ can be formulated as follows:

$$\max_{\{x_{ij}^\tau, y_{ij}^\tau, s_{ij}, g^\tau\}} \sum_{i=1}^N \sum_{j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^\tau} U_{ij}^T(s_{ij}) - \sum_{\tau=T}^{T+\Delta T} \mathbb{E}(C(g^\tau)) \tag{11}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{\tau \in \mathcal{T}_{ij}^1(T)} x_{ij}^\tau - s_{ij} = 0,\ 1 \le i \le N, \\ & \quad j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^\tau; \\ & \sum_{\tau \in \mathcal{T}_{ij}^2(T)} y_{ij}^\tau = \tilde{F}_{ij}(T),\ 1 \le i \le N, \\ & \quad j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^\tau; \\ & 0 \le x_{ij}^\tau \le X_{ij},\ 1 \le i \le N,\ j \in \mathcal{D}_i^\tau, \\ & \quad T \le \tau \le T+\Delta T; \\ & 0 \le y_{ij}^\tau \le Y_{ij},\ 1 \le i \le N,\ j \in \mathcal{F}_i^\tau, \\ & \quad T \le \tau \le T+\Delta T; \\ & g^\tau = \sum_{i=1}^N \sum_{j \in \mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^N \sum_{j \in \mathcal{F}_i^\tau} y_{ij}^\tau \\ & \quad + \sum_{i=1}^N \sum_{j \in \mathcal{B}_i} z_{ij}^\tau, \\ & \quad T \le \tau \le T+\Delta T; \\ & \mathbb{P}(g^\tau > G^m) < \epsilon, \\ & \quad T \le \tau \le T+\Delta T. \end{aligned} \tag{12}$$

B. Gaussian Approximation

We now apply the Gaussian approximation to the background tasks, to transform probabilistic constraint (12) to linear constraint.

Let p_{ij}^τ be the probability that task $(i,j)_\mathcal{B}$ is on at slot τ. We have the following recursion for p_{ij}^τ:

$$p_{ij}^\tau = p_{ij}^{\tau-1} q_{ij}^{\text{on}}(\tau) + \left(1 - p_{ij}^{\tau-1}\right)\left(1 - q_{ij}^{\text{off}}(\tau)\right). \tag{13}$$

with the initial values $p_{ij}^1 = q_{ij}^{\text{on}}$ if task $(i,j)_\mathcal{B}$ is on at time slot $T-1$, and $p_{ij}^1 = 1 - q_{ij}^{\text{off}}$ otherwise. Note that z_{ij}^τ satisfies a binomial distribution with probability p_{ij}^τ for $z_{ij}^\tau = B_{ij}$ and probability $1 - p_{ij}^\tau$ for $z_{ij}^\tau = 0$. Note that $\delta < p_{ij}^\tau < 1-\delta$ for all background tasks $(i,j)_\mathcal{B}$ and $1 \le \tau \le H$.

Then, the mean and variance of the sum energy consumption of background tasks at slot τ, denoted as Z^τ and V^τ, respectively, are given as follows:

$$\begin{aligned} Z^\tau &= \sum_{i=1}^N \sum_{j \in \mathcal{B}_i} p_{ij}^\tau B_{ij} \triangleq \sum_{i=1}^N Z_i^\tau, \\ V^\tau &= \sum_{i=1}^N \sum_{j \in \mathcal{B}_i} \left(1 - p_{ij}^\tau\right) p_{ij}^\tau (B_{ij})^2 \triangleq \sum_{i=1}^N V_i^\tau, \end{aligned} \tag{14}$$

where Z_i^τ and V_i^τ denote the mean and variance of background tasks for consumer i, respectively. Based on Gaussian approximation, we have that

$$g^\tau \sim \mathcal{N}\left(\sum_{i=1}^N \sum_{j \in \mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^N \sum_{j \in \mathcal{F}_i} y_{ij}^\tau + Z^\tau, V^\tau \right). \tag{15}$$

The constraint (12) can be transformed to the following:

$$\sum_{i=1}^N \sum_{j \in \mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^N \sum_{j \in \mathcal{F}_i} y_{ij}^\tau \le G^m - Q^{-1}(\epsilon)\sqrt{V^\tau} - Z^\tau \triangleq X^\tau, \tag{16}$$

where the function $Q^{-1}(\cdot)$ is the inverse function of $Q(x) = \mathbb{P}(n > x)$ for the unit normal distributed variable $n \sim \mathcal{N}(0,1)$.

The Gaussian approximation employed here is based on an extended version of the central limit theorem for independent random variables but not necessarily identically distributed. The justification is provided in the Appendix. Note that it is based on the assumption that $q_{ij}^{\text{on}}(\tau), q_{ij}^{\text{off}}(\tau) \in (\delta, 1-\delta)$ for all $1 \le i \le N$, $j \in \mathcal{B}_i$, and $1 \le \tau \le H$.

Remark: Note that here we transform the stochastic constraints to linear constraints using Gaussian approximation. Another related approach for incorporating the impact of the uncertainties in the analysis is to design a scheduler that is robust against the uncertainties in terms of the model parameters [14], which is not the scenario of this work. Thus, in this work we do not employ robust optimization, but convert the stochastic constraints to linear constraints using Gaussian approximation.

C. Problem Reformulation

Consider $\mathbb{E}(C(g^\tau))$ in the objective function (18). Let $h^\tau = \sum_{i=1}^N \sum_{j \in \mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^N \sum_{j \in \mathcal{F}_i^\tau} y_{ij}^\tau$ be the total energy consumption of all dynamic tasks in time slot τ. We have the total energy consumption $g^\tau = h^\tau + n^\tau$ where $n^\tau \sim \mathcal{N}(Z^\tau, V^\tau)$. From (15)

$$\begin{aligned} \mathbb{E}(C(g^\tau)) &= \mathbb{E}(c_1 g^\tau + c_2 (g^\tau)^2) \\ &= \mathbb{E}(c_1(h^\tau + n^\tau) + c_2(h^\tau + n^\tau)^2) \\ &= (c_2 V^\tau + c_2 (Z^\tau)^2 + c_1 Z^\tau) + (c_1 + 2c_2 Z^\tau) h^\tau \\ &\quad + c_2 (h^\tau)^2 \triangleq C^\tau(h^\tau). \end{aligned} \tag{17}$$

Based on (16) and (17), we reformulate the optimization problem (18)-(12) as follows.

$$\max_{\{x_{ij}^\tau, y_{ij}^\tau, s_{ij}, h^\tau\}} \sum_{i=1}^{N} \sum_{j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^\tau} U_{ij}^T(s_{ij}) - \sum_{\tau=T}^{T+\Delta T} C^\tau(h^\tau) \quad (18)$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{\tau \in \mathcal{T}_{ij}^1(T)} x_{ij}^\tau - s_{ij} = 0,\ 1 \le i \le N, \\
& \quad j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^\tau; \\
& \sum_{\tau \in \mathcal{T}_{ij}^2(T)} y_{ij}^\tau = \tilde{F}_{ij}(T),\ 1 \le i \le N, \\
& \quad j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^\tau; \\
& 0 \le x_{ij}^\tau \le X_{ij},\ 1 \le i \le N,\ j \in \mathcal{D}_i^\tau, \\
& \quad T \le \tau \le T+\Delta T; \\
& 0 \le y_{ij}^\tau \le Y_{ij},\ 1 \le i \le N,\ j \in \mathcal{F}_i^\tau, \\
& \quad T \le \tau \le T+\Delta T; \\
& \sum_{i=1}^{N} \sum_{j \in \mathcal{D}_i^\tau} x_{ij}^\tau + \sum_{i=1}^{N} \sum_{j \in \mathcal{F}_i} y_{ij}^\tau - h^\tau = 0, \\
& \quad T \le \tau \le T+\Delta T; \\
& h^\tau \le X^\tau,\ T \le \tau \le T+\Delta T. \qquad (19)
\end{aligned}$$

We propose a distributed algorithm for solving the energy scheduling problem in hand. The major motivations for implementing a distributed algorithm can be summarized as follows:

1) Computations: The computation load can be distributed among all different computational resources across the network, which in return eliminates the need for having a super-powerful processor in charge of the computations.
2) Robustness: The distributed method is more robust as in practice some components (or subnetworks) may break down or stop functioning. Distributing the computation among all components can reduce the impact brought by such unforeseen impediments.
3) Communication: This also reduces the communication load across the network as part of the computations are carried out locally and subnetworks need to exchange only their processed information that are needed by other agents instead of reporting all the information they have collected.
4) Privacy: As a by-product, a distributed solution also helps to preserve the privacy of the agents and improve their security. The reason is that the agents do not have to reveal the entire information they have accumulated and reveal is as much as other agents need. This, indirectly, improves the privacy of the agents, which is one of the major concerns about the futuristic designs for the smart grids.

We also remark that an alternative and more conventional distributed solution is the dual-decomposition method. However, the convergence of the dual-decomposition method sometimes needs very strong requirements (and as shown in this work, the dual-decomposition method does not converge.); and even if it converges a large number of iterations are needed. This is the reason why other distributed solutions for the energy scheduling problem in hand are of interest. In this work we have adopted distributed Newton method for the following reasons: 1) it is distributed; 2) it converges for the energy scheduling scenario in this work while the dual-decomposition method does not.

IV. Distributed Newton's Method

In this section we first give an overview of the distributed Newton's method for network utility maximization (NUM) problems [15]. Note that for the NUM problem in [15], all constraints are linear with 0–1 coefficients. Here we extend the result in [15] by proving that the distributed Newton's method can be used to solve any convex optimization problem with linear constraints and real coefficients.

A. Network Flow Maximization

Consider a network consisting of S sources and L links, with the binary routing matrix $\boldsymbol{R} = [R_{ij}]_{1 \le i \le L, 1 \le j \le S}$ where $R_{ij} = 1$ if link i is on the route from source j and $R_{ij} = 0$ otherwise. Let s_j be the flow of source j for $1 \le j \le S$, and denote $\boldsymbol{s} \triangleq [s_j]_{1 \le j \le S}$. For $1 \le i \le L$, assume that the sum flow of all sources on link i cannot exceed c_i, and denote $\boldsymbol{c} \triangleq [c_i]_{1 \le i \le L}$. From the standard network flow analysis, we have that $\boldsymbol{Rs} \le \boldsymbol{c}$. Assume that the utility of source j is given by $U_j(s_j)$ where the function $U_j(x)$ is monotonically nondecreasing, strictly concave, twice continuously differentiable, and self-concordant. The network flow maximization problem is formulated as follows:

$$\max_{\boldsymbol{s}} \sum_{j=1}^{S} U_j(s_j), \quad \text{s.t. } \boldsymbol{Rs} \le \boldsymbol{c},\ \boldsymbol{s} \ge 0. \qquad (20)$$

The above optimization problem can be approximately reformulated using the log-barrier representation. In particular, nonnegative slack variables $\boldsymbol{y} = [y_i]_{1 \le i \le L}$ are introduced such that $\boldsymbol{Rs} + \boldsymbol{y} = \boldsymbol{c}$. Let $\boldsymbol{x} = [x_j]_{1 \le j \le S+L}$, where $x_j = s_j$ for $1 \le j \le S$ and $x_j = y_{j-S}$ for $S+1 \le j \le S+L$, and $\boldsymbol{A} = [\boldsymbol{R} \,|\, \boldsymbol{I}]$. Then maximization problem (20) can be approximated using the following log-barrier form:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \triangleq -\sum_{j=1}^{S} U_j(x_j) - \mu \sum_{j=1}^{S+L} \log x_j, \quad \text{s.t. } \boldsymbol{Ax} = \boldsymbol{c}. \quad (21)$$

Note that in order for (21) to well approximate (20), the coefficient μ should be sufficiently small.

B. Distributed Newton's Method

Consider the optimization problem (21). Given an initial feasible point $\boldsymbol{x}^0$, the Newton's method iteratively generates a new feasible solution $\boldsymbol{x}^{k+1}$ given by $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \lambda^k \Delta \boldsymbol{x}^k$, where λ^k is a positive step size and $\Delta \boldsymbol{x}^k$ is the Newton's direction that is determined by the following equation:

$$\begin{pmatrix} \nabla^2 f(\boldsymbol{x}^k) & \boldsymbol{A}^T \\ \boldsymbol{A} & 0 \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{x}^k \\ \boldsymbol{w}^k \end{pmatrix} = - \begin{pmatrix} \nabla f(\boldsymbol{x}^k) \\ 0 \end{pmatrix}; \qquad (22)$$

and the vector $\boldsymbol{w}^k = [w_i^k]_{1 \le i \le L}$ is the collection of dual variables for the link capacity constraints [15].

We introduce the procedure of distributed Newton's method given in [15]. Initialize $\boldsymbol{x}^0$ to be any feasible solution. The dual variables $\boldsymbol{w}^k$ and primal variables $\boldsymbol{x}^k$ are updated iteratively as follows. Note that several rounds of updating the dual variables are needed for each update of the primal variables.

1) Update of Dual Variables: The update of dual variables is to efficiently compute $\boldsymbol{w}^k$ from (22) given the current primal variables $\boldsymbol{x}^k$. According to [15], the dual variables are initialized to be zero, and updated as.

$$\boldsymbol{w}^k \leftarrow (\boldsymbol{D}_k + \bar{\boldsymbol{B}}_k)^{-1}(\bar{\boldsymbol{B}}_k - \boldsymbol{B}_k)\boldsymbol{w}^k - (\boldsymbol{D}_k + \bar{\boldsymbol{B}}_k)^{-1}\left(\boldsymbol{A}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)\right). \quad (23)$$

In (23) $\boldsymbol{D}_k$ is an $L \times L$ diagonal matrix with diagonal components

$$(\boldsymbol{D}_k)_{ii} = \left(\boldsymbol{A}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\right)_{ii} \text{ for } 1 \le i \le L; \quad (24)$$

$\boldsymbol{B}_k = \boldsymbol{A}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T - \boldsymbol{D}_k$; and $\bar{\boldsymbol{B}}_k$ is an $L \times L$ diagonal matrix with diagonal components

$$(\bar{\boldsymbol{B}}_k)_{ii} = \sum_{j=1}^{L}(\boldsymbol{B}_k)_{ij}. \quad (25)$$

In the following we rewrite (23) in the form of row vectors of $\boldsymbol{A}$. For $1 \le i \le L$, let $\boldsymbol{a}_i$ be the ith row of $\boldsymbol{A}$, and $\boldsymbol{a} = \sum_{i=1}^{L}\boldsymbol{a}_i$ be the sum of all rows of $\boldsymbol{A}$. Via algebraic manipulations, (23) can be rewritten as follows:

$$w_i^k \leftarrow w_i^k - \frac{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}^k}{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T} - \frac{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)}{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T}, \quad \text{for } 1 \le i \le L. \quad (26)$$

We iterate (26) until $\boldsymbol{w}^k$ converges. Let $\tilde{\boldsymbol{w}}^k$ be the value of $\boldsymbol{w}^k$ when reaching convergence.

2) Update of Primal Variables: Given $\tilde{\boldsymbol{w}}^k$, the Newton's direction $\Delta\tilde{\boldsymbol{x}}^k$ is given by

$$\Delta\tilde{\boldsymbol{x}}^k = -\boldsymbol{H}_k^{-1}(\nabla f(\boldsymbol{x}^k) + \boldsymbol{A}^T\tilde{\boldsymbol{w}}^k), \quad (27)$$

and the primal variables are updated as $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \lambda^k\Delta\tilde{\boldsymbol{x}}^k$, where the step size is set as

$$\lambda^k = \begin{cases} \frac{5}{6(\theta^k+1)} & \text{if } \theta^k \ge \frac{1}{4}, \\ 1 & \text{otherwise}, \end{cases} \quad (28)$$

with $\theta^k = \sqrt{(\Delta\tilde{\boldsymbol{x}}^k)^T\boldsymbol{H}_k\Delta\tilde{\boldsymbol{x}}^k}$.

Remark: The updates of the dual and primal variables can be made distributed due to the diagonal structure of $\boldsymbol{H}_k^{-1}$. The message passing mechanism can be devised according to (53) and (27). Specifically, when updating $w_i^k(t)$ according to (53), the edges j for which $\boldsymbol{a}_i^T\boldsymbol{H}_k^{-1}\boldsymbol{a}_j^T \neq 0$, and the nodes corresponding to the nonzero elements in $\boldsymbol{a}_i^T$, need pass messages to edge i. When updating $\Delta\tilde{\boldsymbol{x}}^k$, each edge j whose linear constraint involves node i need pass messages to node i. The distributed computation of the step size λ^k is also discussed in [15].

Remark: The distributed Newton's method is different from the decomposition method proposed in [16] and its application to power grid [17]. For the decomposition method proposed in [16] and [17], the linear constraints in (22) are changed to approximate form, which facilitates the problem decomposition and distributed solution. However, in the distributed Newton's method the linear constraints are transformed in an equivalent form that involves no approximation.

C. Sufficient Condition for Convergence

In [15], the convergence proof of the distributed Newton's method is based on the following two arguments:

1) Updated using (53), the dual variable $\boldsymbol{w}^k(t)$ approaches the exact value $\boldsymbol{w}^k$ given by (22). Thus the output $\tilde{\boldsymbol{w}}^k$ of the iteration can be sufficiently close to $\boldsymbol{w}^k$.
2) Since $\tilde{\boldsymbol{w}}^k$ can be sufficiently close to $\boldsymbol{w}^k$, the update direction $\Delta\tilde{\boldsymbol{x}}^k$ of the primary variable can be sufficiently close to the exact direction given by (22). Moreover, it can be proved that the distributed Newton's method converges *quadratically* to *a neighborhood of the optimal solution*.

From the above arguments, it is seen that the key step is to guarantee that the dual variable $\boldsymbol{w}^k(t)$ converges to the exact value $\boldsymbol{w}^k$ given by (22). Using the similar arguments as in the proof of *Theorem 4.3* in [15], it follows that if all components of $\boldsymbol{A}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T$ are nonnegative, the dual variable $\boldsymbol{w}^k(t)$ converges to the exact value $\boldsymbol{w}^k$. Since f is convex, we have that $\boldsymbol{H}_k^{-1} \succeq 0$, and thus a sufficient condition for $\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}_j^T \ge 0$ is that $a_{ik}a_{jk} \ge 0$ for $1 \le k \le S + L$, where a_{ik} and a_{jk} are the kth component of $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ respectively. In other words, for each column of $\boldsymbol{A}$ all components are of the same sign. The following result generalizes *Theorem 4.3* in [15].

Theorem 1: If all components of each column of $\boldsymbol{A}$ are of the same sign, then the distributed Newton's method converges *quadratically* to a neighborhood of the optimal solution.

We can further show that any linear constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$, even if not meeting the sufficient condition of Theorem 1, can be transformed to an equivalent linear constraint that satisfies the sufficient condition of Theorem 1. The proof is given in the Appendix.

Theorem 2: Any linear constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$ can be transformed to an equivalent form $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{c}}$, where all components of each column of $\tilde{\boldsymbol{A}}$ are of the same sign.

By applying the transform given in the proof of *Theorem 2*, the distributed Newton's method can be used to solve any convex optimization problem with linear constraints and concordant objective function. Note that although the condition $\mu \ge 1$ is a sufficient condition for the objective function to be concordant, it significantly change the original optimization problem [cf. (20) and (21)]. Thus, in practice a small value μ, usually $\mu \ll 1$, is employed, and empirically fast convergence of the distributed Newton's method is always observed in simulations.

V. Distributed Newton's Algorithm for On-Line Energy Scheduling

In this section, we derive the distributed Newton's algorithm for solving the optimization problem (18). We first rewrite (18) in the standard formulation of Theorem 1 as follows.

$$\min_{\{x_{ij}^\tau, y_{ij}^\tau, s_{ij}, m_{ij}^\tau, n_{ij}^\tau, h^\tau, r^\tau\}} -\sum_{i=1}^{N}\sum_{j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^\tau} U_{ij}^T(s_{ij})$$

$$
\begin{aligned}
&+\sum_{\tau=T}^{T+\Delta T} C^{\tau}(h^{\tau}) - \mu \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^{N}\sum_{j\in\mathcal{D}^{\tau}} \log x_{ij}^{\tau} \\
&-\mu \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^{N}\sum_{j\in\mathcal{F}^{\tau}} \log y_{ij}^{\tau} - \mu \sum_{i=1}^{N}\sum_{j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^{\tau}} \log s_{ij} \\
&-\mu \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^{N}\sum_{j\in\mathcal{D}^{\tau}} \log m_{ij}^{\tau} - \mu \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^{N}\sum_{j\in\mathcal{F}^{\tau}} \log n_{ij}^{\tau} \\
&-\mu \sum_{\tau=T}^{T+\Delta T} \log h^{\tau} - \mu \sum_{\tau=T}^{T+\Delta T} \log r^{\tau} \qquad (29)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\quad & \sum_{\tau\in\mathcal{T}_{ij}^{1}(T)} x_{ij}^{\tau} - s_{ij} = 0, \\
& \quad 1\le i\le N,\ j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^{\tau}; \qquad (30)\\
& \sum_{\tau\in\mathcal{T}_{ij}^{2}(T)} y_{ij}^{\tau} = \tilde{F}_{ij}(T), \\
& \quad 1\le i\le N,\ j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^{\tau}; \qquad (31)\\
& x_{ij}^{\tau} + m_{ij}^{\tau} = X_{ij}, \\
& \quad 1\le i\le N,\ j\in\mathcal{D}_i^{\tau},\ T\le\tau\le T+\Delta T; \qquad (32)\\
& y_{ij}^{\tau} + n_{ij}^{\tau} = Y_{ij}, \\
& \quad 1\le i\le N,\ j\in\mathcal{F}_i^{\tau},\ T\le\tau\le T+\Delta T; \qquad (33)\\
& \sum_{i=1}^{N}\sum_{j\in\mathcal{D}_i^{\tau}} x_{ij}^{\tau} + \sum_{i=1}^{N}\sum_{j\in\mathcal{F}_i^{\tau}} y_{ij}^{\tau} - h^{\tau} = 0, \\
& \quad T\le\tau\le T+\Delta T; \qquad (34)\\
& \sum_{i=1}^{N}\sum_{j\in\mathcal{D}_i^{\tau}} x_{ij}^{\tau} + \sum_{i=1}^{N}\sum_{j\in\mathcal{F}_i^{\tau}} y_{ij}^{\tau} + r^{\tau} = X^{\tau}, \\
& \quad T\le\tau\le T+\Delta T; \\
& x_{ij}^{\tau}\ge 0,\ y_{ij}^{\tau}\ge 0,\ s_{ij}(T)\ge 0,\ m_{ij}^{\tau}\ge 0, \\
& \quad n_{ij}^{\tau}\ge 0,\ h^{\tau}\ge 0,\ r^{\tau}\ge 0. \qquad (35)
\end{aligned}
$$

Remark: (34) also can be written in the following form:

$$h^{\tau} + r^{\tau} = X^{\tau},\ T\le\tau\le T+\Delta T. \qquad (36)$$

However, then the sign of h^{τ} is negative in (31) but positive in (33), which violates the sufficient condition in *Theorem 1*. Indeed, numerical results show that if (34) is replaced by (36), then the distributed Newton's method does not converge. Hence, in order to apply the distributed Newton's method, it is crucial to represent the optimization problem in a form that satisfies the sufficient condition given by *Theorem 2*.

A. Notations

Let f be the objective function of the optimization problem (29). For variable $\gamma = x_{ij}^{\tau}, y_{ij}^{\tau}, s_{ij}, m_{ij}^{\tau}, n_{ij}^{\tau}, h^{\tau}$, or r^{τ}, define the following first and second derivatives:

$$\nabla_1(\gamma) \triangleq \frac{\partial f}{\partial\gamma} \text{ and } \nabla_2(\gamma) \triangleq \frac{\partial^2 f}{(\partial\gamma)^2}, \qquad (37)$$

and the corresponding inverses $\nabla_1^{-1}(\gamma) \triangleq (\nabla_1(\gamma))^{-1}$ and $\nabla_2^{-1}(\gamma) \triangleq (\nabla_2(\gamma))^{-1}$. The expressions of the above *auxiliary variables* are given in the Appendix, Section D. Let $\boldsymbol{H}_k^{-1}$ be the diagonal Hessian matrix with the diagonal elements

$$\boldsymbol{H}_k^{-1} = \operatorname{diag}(\nabla_2(\gamma)), \qquad (38)$$

for $\gamma = x_{ij}^{\tau}, y_{ij}^{\tau}, s_{ij}, m_{ij}^{\tau}, n_{ij}^{\tau}, h^{\tau}$ and r^{τ} sequentially in the same order as that for those shown in (39).

The matrix form of the linear constraints in (30)–(35) are shown in (), where the rows of the overall linear constraint matrix, denoted as $\boldsymbol{A}$, are partitioned into six parts $\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{A}^{(3)}$, $\boldsymbol{A}^{(4)}$, $\boldsymbol{A}^{(5)}$, and $\boldsymbol{A}^{(6)}$, corresponding to the constraints in (30), (31), (32), (33), (34), and (35), respectively, where the coefficient vectors $\boldsymbol{a}_{i,j}^{(1)}$, $\boldsymbol{a}_{i,j}^{(2)}$, $\boldsymbol{a}_{i,j,\tau}^{(3)}$, $\boldsymbol{a}_{i,j,\tau}^{(4)}$, $\boldsymbol{a}_{\tau}^{(5)}$, and $\boldsymbol{a}_{\tau}^{(6)}$ denotes the linear constraints with the corresponding indices (i,j) in (30) and (31), (i,j,τ) in (32) and (33), and τ in (34) and (35), where the ranges of indices are the same as those given by (30)–(35).

$$
\begin{bmatrix} \boldsymbol{A}^{(1)} \\ \hline \boldsymbol{A}^{(2)} \\ \hline \boldsymbol{A}^{(3)} \\ \hline \boldsymbol{A}^{(4)} \\ \hline \boldsymbol{A}^{(5)} \\ \hline \boldsymbol{A}^{(6)} \end{bmatrix}
\begin{bmatrix} x_{ij}^{\tau} \\ \hline y_{ij}^{\tau} \\ \hline s_{ij} \\ \hline m_{ij}^{\tau} \\ \hline n_{ij}^{\tau} \\ \hline h^{\tau} \\ \hline r^{\tau} \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{a}_{i,j}^{(1)} \\ \hline \boldsymbol{a}_{i,j}^{(2)} \\ \hline \boldsymbol{a}_{i,j,\tau}^{(3)} \\ \hline \boldsymbol{a}_{i,j,\tau}^{(4)} \\ \hline \boldsymbol{a}_{\tau}^{(5)} \\ \hline \boldsymbol{a}_{\tau}^{(6)} \end{bmatrix}
\begin{bmatrix} x_{ij}^{\tau} \\ \hline y_{ij}^{\tau} \\ \hline s_{ij} \\ \hline m_{ij}^{\tau} \\ \hline n_{ij}^{\tau} \\ \hline h^{\tau} \\ \hline r^{\tau} \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{0} \\ \hline \tilde{F}_{ij} \\ \hline X_{ij} \\ \hline Y_{ij} \\ \hline \boldsymbol{0} \\ \hline X^{\tau} \end{bmatrix}. \qquad (39)
$$

Furthermore, we note that a large portion of the linear constraint matrix $\boldsymbol{A}$ is zero. Partitioning the rows of $\boldsymbol{A}$ into seven parts corresponding to the variables $x_{ij}^{\tau}, y_{ij}^{\tau}, s_{ij}, m_{ij}^{\tau}, n_{ij}^{\tau}, h^{\tau}$, and r^{τ} from left to right, we have the representation shown in (40) at the bottom of the page.

Similarly, we partition the dual variables $\boldsymbol{w}$ into four parts:

$\boldsymbol{w}^{(1)} \triangleq [w_{i,j}^{(1)}]_{1\le i\le N, j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^{\tau}}$,

$\boldsymbol{w}^{(2)} \triangleq [w_{i,j}^{(2)}]_{1\le i\le N, j\in\cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^{\tau}}$,

$\boldsymbol{w}^{(3)} \triangleq [w_{i,j,\tau}^{(3)}]_{1\le i\le N, T\le\tau\le T+\Delta T, j\in\mathcal{D}_i^{\tau}}$,

$\boldsymbol{w}^{(4)} \triangleq [w_{i,j,\tau}^{(4)}]_{1\le i\le N, T\le\tau\le T+\Delta T, j\in\mathcal{F}_i^{\tau}}$,

$$
\boldsymbol{A} = \left[\begin{array}{c|c|c|c|c|c|c}
\boldsymbol{A}_1^{(1)} & 0 & \boldsymbol{A}_3^{(1)} & 0 & 0 & 0 & 0 \\ \hline
0 & \boldsymbol{A}_2^{(2)} & 0 & 0 & 0 & 0 & 0 \\ \hline
\boldsymbol{A}_1^{(3)} & 0 & 0 & \boldsymbol{A}_4^{(3)} & 0 & 0 & 0 \\ \hline
0 & \boldsymbol{A}_2^{(4)} & 0 & 0 & \boldsymbol{A}_5^{(4)} & 0 & 0 \\ \hline
\boldsymbol{A}_1^{(5)} & \boldsymbol{A}_2^{(5)} & 0 & 0 & 0 & \boldsymbol{A}_6^{(5)} & 0 \\ \hline
\boldsymbol{A}_1^{(6)} & \boldsymbol{A}_2^{(6)} & 0 & 0 & 0 & 0 & \boldsymbol{A}_7^{(6)}
\end{array}\right]. \qquad (40)
$$

TABLE I
SUMMARY OF NOTATIONS IN THE PROPOSED DISTRIBUTED NEWTON'S ALGORITHM

Linear Constraint	Vector	Dual Variables	Auxiliary Variables
0 [c.f.(30)]	$\boldsymbol{a}_{i,j}^{(1)}$	$w_{i,j}^{(1)}$	$v_{i,j}^{(1)}$, $K_{i,j}^{(1)}$, $L_{i,j}^{(1)}$
$\tilde{F}_{ij}$ [c.f.(31)]	$\boldsymbol{a}_{i,j}^{(2)}$	$w_{i,j}^{(2)}$	$v_{i,\tau}^{(2)}$, $K_{i,\tau}^{(2)}$, $L_{i,\tau}^{(2)}$
X_{ij} [c.f.(32)]	$\boldsymbol{a}_{i,j,\tau}^{(3)}$	$w_{i,j,\tau}^{(3)}$	$v_{i,j,\tau}^{(3)}$, $K_{i,j,\tau}^{(3)}$, $L_{i,j,\tau}^{(3)}$
Y_{ij} [c.f.(33)]	$\boldsymbol{a}_{i,j,\tau}^{(4)}$	$w_{i,j,\tau}^{(4)}$	$v_{i,j,\tau}^{(4)}$, $K_{i,j,\tau}^{(4)}$, $L_{i,j,\tau}^{(4)}$
0 [c.f.(34)]	$\boldsymbol{a}_{\tau}^{(5)}$	$w_{\tau}^{(5)}$	$v_{\tau}^{(5)}$, $K_{\tau}^{(5)}$, $L_{\tau}^{(5)}$
X^{τ} [c.f.(35)]	$\boldsymbol{a}_{\tau}^{(6)}$	$w_{\tau}^{(6)}$	$v_{\tau}^{(6)}$, $K_{\tau}^{(6)}$, $L_{\tau}^{(6)}$

$\boldsymbol{w}^{(5)} \triangleq [w_{\tau}^{(5)}]_{T\le\tau\le T+\Delta T}$,
$\boldsymbol{w}^{(6)} \triangleq [w_{\tau}^{(6)}]_{T\le\tau\le T+\Delta T}$,
corresponding to the constraints (30), (31), (32), (33), (34), and (35), respectively.

Considering the update of dual variables according to (53), we define the following *auxiliary variables*. Let $\boldsymbol{a}$ be the sum of all rows of $\boldsymbol{A}$.

- For $1 \le i \le N$ and $j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^{\tau}$, define $v_{i,j}^{(1)} \triangleq \boldsymbol{a}_{i,j}^{(1)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{i,j}^{(1)} \triangleq \boldsymbol{a}_{i,j}^{(1)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{i,j}^{(1)} \triangleq \boldsymbol{a}_{i,j}^{(1)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.
- For $1 \le i \le N$ and $j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^{\tau}$, define $v_{i,j}^{(2)} \triangleq \boldsymbol{a}_{i,j}^{(2)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{i,j}^{(2)} \triangleq \boldsymbol{a}_{i,j}^{(2)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{i,j}^{(2)} \triangleq \boldsymbol{a}_{i,j}^{(2)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.
- For $1 \le i \le N$, $T \le \tau \le T+\Delta T$, and $j \in \mathcal{D}_i^{\tau}$, define $v_{i,j,\tau}^{(3)} \triangleq \boldsymbol{a}_{i,j}^{(3)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{i,j,\tau}^{(3)} \triangleq \boldsymbol{a}_{i,j,\tau}^{(3)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{i,j,\tau}^{(3)} \triangleq \boldsymbol{a}_{i,j,\tau}^{(3)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.
- For $1 \le i \le N$, $T \le \tau \le T+\Delta T$, and $j \in \mathcal{F}_i^{\tau}$, define $v_{i,j,\tau}^{(4)} \triangleq \boldsymbol{a}_{i,j,\tau}^{(4)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{i,j,\tau}^{(4)} \triangleq \boldsymbol{a}_{i,j,\tau}^{(4)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{i,j,\tau}^{(4)} \triangleq \boldsymbol{a}_{i,j,\tau}^{(4)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.
- For $T \le \tau \le T+\Delta T$, define $v_{\tau}^{(5)} \triangleq \boldsymbol{a}_{\tau}^{(5)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{\tau}^{(5)} \triangleq \boldsymbol{a}_{\tau}^{(5)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{\tau}^{(5)} \triangleq \boldsymbol{a}_{\tau}^{(5)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.
- For $T \le \tau \le T+\Delta T$, define $v_{\tau}^{(6)} \triangleq \boldsymbol{a}_{\tau}^{(6)}\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)$, $K_{\tau}^{(6)} \triangleq \boldsymbol{a}_{\tau}^{(6)}\boldsymbol{H}_k^{-1}\boldsymbol{a}^T$, and $L_{\tau}^{(6)} \triangleq \boldsymbol{a}_{\tau}^{(6)}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}$.

The linear constraints, the dual variables, and the auxiliary variables are summarized in Table I.

B. Distributed Newton's Algorithm

1) Initialization: We initialize the primal variables x_{ij}^{τ}, y_{ij}^{τ}, $s_{ij}(T)$, m_{ij}^{τ}, n_{ij}^{τ}, h^{τ}, and r^{τ} as a feasible solution subject to the constraint of optimization problem (29).

When updating the direction $\Delta\tilde{\boldsymbol{x}}^0$ based on the initial primal value $\boldsymbol{x}^0$, we initialize the dual $\boldsymbol{w}^0(0) = \boldsymbol{0}$; and when updating the direction $\Delta\tilde{\boldsymbol{x}}^k$ for $k \ge 1$, we initialize $\boldsymbol{w}^k(0) = \tilde{\boldsymbol{w}}^{k-1}$ as the final value of the dual in the previous iteration.

2) First and Second-Order Derivatives of Primal Variables: The first and second derivatives of the primal variables in (37) are specified as follows:

$$\nabla_1\left(x_{ij}^{\tau}\right) = -\frac{\mu}{x_{ij}^{\tau}}, \quad \nabla_2\left(x_{ij}^{\tau}\right) = \frac{\mu}{\left(x_{ij}^{\tau}\right)^2};$$
$$\nabla_1\left(y_{ij}^{\tau}\right) = -\frac{\mu}{y_{ij}^{\tau}}, \quad \nabla_2\left(y_{ij}^{\tau}\right) = \frac{\mu}{\left(y_{ij}^{\tau}\right)^2};$$
$$\nabla_1(s_{ij}) = -\frac{\partial U_{ij}^T}{\partial s_{ij}} - \frac{\mu}{s_{ij}},$$
$$\nabla_2(s_{ij}) = -\frac{\partial^2 U_{ij}^T}{(\partial s_{ij})^2} + \frac{\mu}{(s_{ij})^2};$$
$$\nabla_1\left(m_{ij}^{\tau}\right) = -\frac{\mu}{m_{ij}^{\tau}}, \quad \nabla_2\left(m_{ij}^{\tau}\right) = \frac{\mu}{\left(m_{ij}^{\tau}\right)^2};$$
$$\nabla_1\left(n_{ij}^{\tau}\right) = -\frac{\mu}{n_{ij}^{\tau}}, \quad \nabla_2\left(n_{ij}^{\tau}\right) = \frac{\mu}{\left(n_{ij}^{\tau}\right)^2};$$
$$\nabla_1(h^{\tau}) = \frac{\partial C^{\tau}}{\partial h^{\tau}} - \frac{\mu}{h^{\tau}}, \ \nabla_2(h^{\tau}) = \frac{\partial^2 C^{\tau}}{(\partial h^{\tau})^2} + \frac{\mu}{(h^{\tau})^2};$$
$$\nabla_1(r^{\tau}) = -\frac{\mu}{r^{\tau}}, \quad \nabla_2(r^{\tau}) = \frac{\mu}{(r^{\tau})^2}. \tag{41}$$

3) Dual Variable Update: First, the auxiliary variables are computed according to Appendix, Section D. Note that the distributed implementation can be facilitated via computing the variables $\bar{v}_{i,\tau}^{(2)}$, $\bar{K}_{i,\tau}^{(2)}$, and $\bar{L}_{i,\tau}^{(2)}$ at each energy consumer, and passing them to the energy source to compute $v_{\tau}^{(5)}$, $v_{\tau}^{(6)}$, $K_{\tau}^{(5)}$, $K_{\tau}^{(6)}$, $L_{\tau}^{(5)}$, and $L_{\tau}^{(6)}$.

According to (26), the update of the dual variables can be expressed via the auxiliary variables as follows:

$$w_{i,j}^{(1)} \leftarrow w_{i,j}^{(1)} - \frac{L_{i,j}^{(1)} + v_{i,j}^{(1)}}{K_{i,j}^{(1)}}, \quad 1 \le i \le N, \ j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^{\tau};$$
$$w_{i,j}^{(2)} \leftarrow w_{i,j}^{(2)} - \frac{L_{i,j}^{(2)} + v_{i,j}^{(2)}}{K_{i,j}^{(2)}}, \quad 1 \le i \le N, \ j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^{\tau};$$
$$w_{i,j,\tau}^{(3)} \leftarrow w_{i,j,\tau}^{(3)} - \frac{L_{i,j,\tau}^{(3)} + v_{i,j,\tau}^{(3)}}{K_{i,j,\tau}^{(3)}}, \quad 1 \le i \le N, \ T \le \tau \le T+\Delta T, \ j \in \mathcal{D}_i^{\tau};$$
$$w_{i,j,\tau}^{(4)} \leftarrow w_{i,j,\tau}^{(4)} - \frac{L_{i,j,\tau}^{(4)} + v_{i,j,\tau}^{(4)}}{K_{i,j,\tau}^{(4)}}, \quad 1 \le i \le N, \ T \le \tau \le T+\Delta T, \ j \in \mathcal{F}_i^{\tau};$$
$$w_{\tau}^{(5)} \leftarrow w_{\tau}^{(5)} - \frac{L_{\tau}^{(5)} + v_{\tau}^{(5)}}{K_{\tau}^{(5)}}, \ T \le \tau \le T+\Delta T;$$
$$w_{\tau}^{(6)} \leftarrow w_{\tau}^{(6)} - \frac{L_{\tau}^{(6)} + v_{\tau}^{(6)}}{K_{\tau}^{(6)}}, \ T \le \tau \le T+\Delta T. \tag{42}$$

4) Primal Variable Update: After the iterative dual variable update in (42) reaches convergence, we compute the directions for the primal variable updates as follows:

$$\Delta x_{ij}^{\tau} = -\frac{\nabla_1\left(x_{ij}^{\tau}\right) + w_{i,j}^{(1)} + w_{i,j,\tau}^{(3)} + \left(w_{\tau}^{(5)} + w_{\tau}^{(6)}\right)}{\nabla_2\left(x_{ij}^{\tau}\right)},$$
$$\Delta y_{ij}^{\tau} = -\frac{\nabla_1\left(y_{ij}^{\tau}\right) + w_{i,j}^{(2)} + w_{i,j,\tau}^{(4)} + \left(w_{\tau}^{(5)} + w_{\tau}^{(6)}\right)}{\nabla_2\left(y_{ij}^{\tau}\right)},$$
$$\Delta s_{ij} = -\frac{\nabla_1(s_{ij}) - w_{i,j}^{(1)}}{\nabla_2(s_{ij})},$$
$$\Delta m_{ij}^{\tau} = -\frac{\nabla_1\left(m_{ij}^{\tau}\right) + w_{i,j,\tau}^{(3)}}{\nabla_2\left(m_{ij}^{\tau}\right)},$$

$$\Delta n_{ij}^{\tau} = -\frac{\nabla_1 \left(s_{ij}^{\tau}\right) + w_{i,j,\tau}^{(4)}}{\nabla_2 \left(n_{ij}^{\tau}\right)},$$
$$\Delta h^{\tau} = -\frac{\nabla_1(h^{\tau}) - w_{\tau}^{(5)}}{\nabla_2(h^{\tau})},$$
$$\Delta r^{\tau} = -\frac{\nabla_1(r^{\tau}) + w_{\tau}^{(6)}}{\nabla_2(r^{\tau})}. \quad (43)$$

We then update the primal variables as $\gamma \leftarrow \gamma + \lambda^k \Delta\gamma$ for $\gamma = x_{ij}^{\tau}, y_{ij}^{\tau}, s_{ij}, m_{ij}^{\tau}, n_{ij}^{\tau}, h^{\tau}$, or r^{τ}. When computing the step size λ^k according to (27), the parameter θ^k is given by

$$(\theta^k)^2 = \sum_{i=1}^{N} \nu_i + \sum_{\tau=T}^{T+\Delta T} (\Delta h^{\tau})^2 \nabla_2(h^{\tau}) + \sum_{\tau=T}^{T+\Delta T} (\Delta r^{\tau})^2 \nabla_2(r^{\tau}), \quad (44)$$

where the contribution from energy consumer i, denoted as ν_i for $1 \le i \le N$, is given by

$$\nu_i = \sum_{\tau=T}^{T+\Delta T} \sum_{j \in \mathcal{D}_i^{\tau}} (\Delta x_{ij}^{\tau})^2 \nabla_2(x_{ij}^{\tau}) + \sum_{\tau=T}^{T+\Delta T} \sum_{j \in \mathcal{F}_i^{\tau}} \left(\Delta y_{ij}^{\tau}\right)^2 \nabla_2\left(y_{ij}^{\tau}\right) + \sum_{j \in \mathcal{D}_i^{\tau}} \sum_{j \in \mathcal{D}_i^{\tau}} \left(\Delta m_{ij}^{\tau}\right)^2 \nabla_2\left(m_{ij}^{\tau}\right) + \sum_{\tau=T}^{T+\Delta T} \sum_{j \in \mathcal{F}_i^{\tau}} \left(\Delta n_{ij}^{\tau}\right)^2 \nabla_2\left(n_{ij}^{\tau}\right) + \sum_{\tau=T}^{T+\Delta T} (\Delta s_{ij})^2 \nabla_2(s_{ij}). \quad (45)$$

C. Message Passing Scheme

We now elaborate on the message passing scheme between the energy source and the energy consumers during the distributed implementation of the Newton's algorithm given in Section V-B.

First we describe the storage scheme for the primal, dual, and auxiliary variables as follows.

- **Energy Consumers:** For $1 \le i \le N$, each energy consumer i stores the energy consumption $\{x_{ij}^{\tau}, \Delta x_{ij}^{\tau}, y_{ij}^{\tau}, \Delta y_{ij}^{\tau}\}$, the slack and dual variables $\{m_{ij}^{\tau}, \Delta m_{ij}^{\tau}, n_{ij}^{\tau}, \Delta n_{ij}^{\tau}, s_{ij}, \Delta s_{ij}, w_{i,j}^{(1)}, w_{i,j}^{(2)}, w_{i,j,\tau}^{(3)}, w_{i,j,\tau}^{(4)}\}$ for tasks $j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^{\tau}$ and $j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^{\tau}$.
- **Energy source:** The energy source stores the energy generation, the slack and dual variables $\{h^{\tau}, r^{\tau}, \Delta h^{\tau}, \Delta r^{\tau}, w_{\tau}^{(5)}, w_{\tau}^{(6)}\}$ for time slots $T \le \tau \le T + \Delta T$.

Next we describe the message passing mechanism for the distributed Newton's algorithm, which is illustrated in Fig. 1, including the problem transformation step $(\mathbf{T_1})$, the primal variable initialization steps $(\mathbf{I_1})$–$(\mathbf{I_3})$, the dual variable update steps $(\mathbf{D_1})$–$(\mathbf{D_2})$, and the primal variable update steps $(\mathbf{P_1})$–$(\mathbf{P_4})$. The messages being passed at each step are also shown in Fig. 1. We elaborate on these steps as follows.

1) Problem Transformation:

$(\mathbf{T_1})$: For $1 \le i \le N$, each energy consumer i passes Z_i^{τ} and V_i^{τ} to the energy source. The energy source computes $Z^{\tau} = \sum_{i=1}^{N} Z_i^{\tau}$ and $V^{\tau} = \sum_{i=1}^{N} V_i^{\tau}$, and obtains the expected cost $C^{\tau}(h^{\tau})$ according to (17).

2) Primal Variable Initialization:

$(\mathbf{I_1})$: Each energy consumer i, $1 \le i \le N$, passes $\mathcal{D}_i^{\tau}$ and $\mathcal{F}_i^{\tau}$ for $T \le \tau \le T+\Delta T$, and $\tilde{F}_{ij}(T)$ for $j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^{\tau}$, to the energy source.

$(\mathbf{I_2})$: The energy source broadcasts feasible solution of x_{ij}^{τ} and y_{ij}^{τ} to all energy consumers for initialization.

3) Dual Variable Update:

$(\mathbf{D_1})$: The energy source broadcasts $(w_{\tau}^{(5)} + w_{\tau}^{(6)})$, $T \le \tau \le T + \Delta T$, to all energy consumers. Then each energy consumer updates the auxiliary variables $\{v_{i,j}^{(1)}, v_{i,j}^{(2)}, v_{i,j,\tau}^{(3)}, v_{i,j,\tau}^{(4)}\}$, $\{K_{i,j}^{(1)}, K_{i,j}^{(2)}, K_{i,j,\tau}^{(3)}, K_{i,j,\tau}^{(4)}\}$, and $\{L_{i,j}^{(1)}, L_{i,j}^{(2)}, L_{i,j,\tau}^{(3)}, L_{i,j,\tau}^{(4)}\}$, according to (58), (59) and (60); and then updates the dual variables $w_{i,j}^{(1)}$, $w_{i,j}^{(2)}$, $w_{i,j,\tau}^{(3)}$, and $w_{i,j,\tau}^{(4)}$, according to (42).

$(\mathbf{D_2})$: Each energy consumer i passes $(\bar{K}_{i,\tau}^{(2)}, \bar{v}_{i,\tau}^{(2)}, \bar{L}_{i,\tau}^{(2)})$ to the energy source. Then the energy source updates the auxiliary variables $\{v_{\tau}^{(5)}, v_{\tau}^{(6)}\}$, $\{K_{\tau}^{(5)}, K_{\tau}^{(6)}\}$, and $\{L_{\tau}^{(5)}, L_{\tau}^{(6)}\}$ according to (58), (59), and (60), and then updates the dual variables $w_{\tau}^{(5)}$ and $w_{\tau}^{(6)}$ according to (42).

4) Primal Variable Update:

$(\mathbf{P_1})$: After updating the dual variables $\boldsymbol{w}$, the energy source broadcasts the updated $(w_{\tau}^{(5)} + w_{\tau}^{(6)})$, $T \le \tau \le T + \Delta T$, to all energy consumers. Then each energy consumer i obtains the update direction $\Delta x_{i,j}^{\tau}$ for $j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{D}_i^{\tau}$ and $\Delta y_{i,j}^{\tau}$ for $j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^{\tau}$, according to (43).

$(\mathbf{P_2})$: Each energy consumer i computes ν_i according to (45) and passes it to the energy source. Then the energy source computes the step size λ^k based on the received ν_i from all energy consumers.

$(\mathbf{P_3})$: The energy source broadcasts the step size λ^k to all energy consumers. Then each energy consumer updates the energy consumption x_{ij}^{τ} and y_{ij}^{τ} using the received step size.

$(\mathbf{P_4})$: Each energy consumer i, $1 \le i \le N$, computes the variables s_{ij}, m_{ij}^{τ}, and n_{ij}^{τ}, and then passes its sum energy consumption $\sum_{j \in \mathcal{D}_i^{\tau}} x_{ij}^{\tau} + \sum_{j \in \mathcal{F}_i^{\tau}} y_{ij}^{\tau}$ corresponding to each time slot $T \le \tau \le T + \Delta T$ to the energy source. Then the energy source obtains h^{τ} and r^{τ} for $T \le \tau \le T + \Delta T$ according to (34) and (35).

Remark: In this work, we have considered two types of dynamic tasks, and have proposed a distributed Newton's method for solving an energy scheduling problem. However, it is noteworthy that the proposed framework is not limited to the two types of dynamic tasks under consideration. Specifically, based on the needs of the scheduler and in the circumstances that there exist additional dynamic tasks that satisfy the linear energy constraint, the framework can accommodate such additional dynamic tasks as well without affecting the basic structures of the approximations and message passing processes.

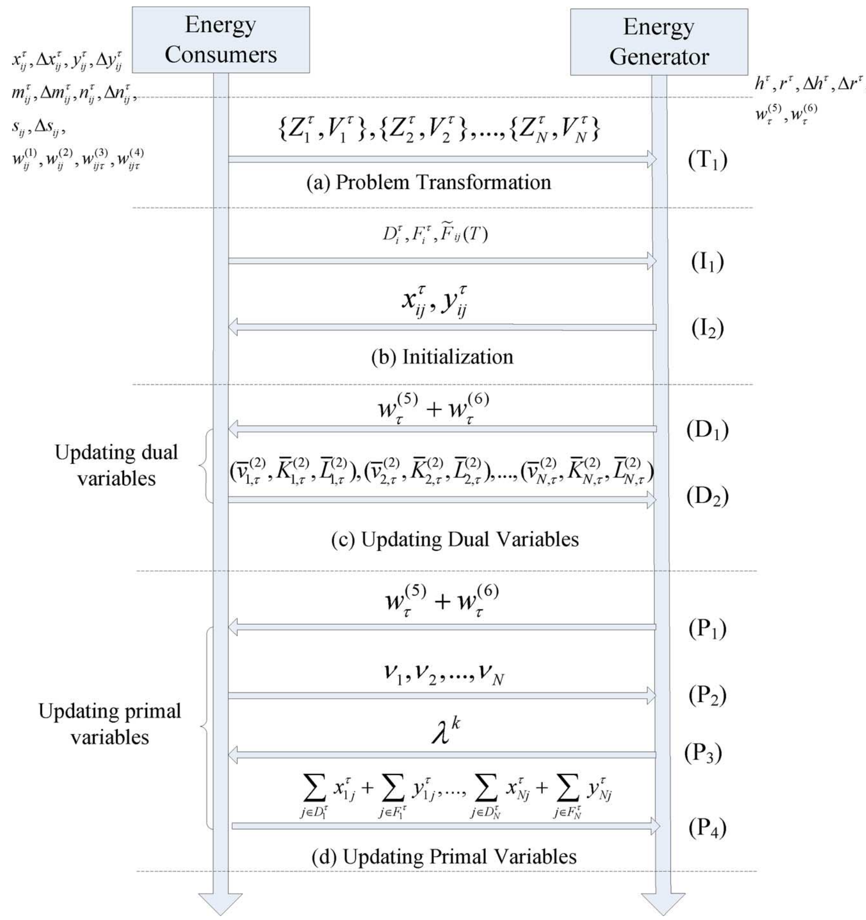

Fig. 1. The detailed message-passing mechanism for the proposed distributed Newton's algorithm.

VI. Numerical Results

We consider an electrical power network with a single energy source and 40 energy consumers, during the time slots for $1 \leq \tau \leq 1000$. Assume that for each energy consumer there are 10 background tasks, i.e., $|\mathcal{B}_i| = 10$ for $1 \leq i \leq 40$. The energy consumption B_{ij} of each background task $(i,j)_{\mathcal{B}}$ is uniformly distributed over [0.05, 0.1]. Both probabilities $q_{ij}^{\rm on}(\tau)$ and $q_{ij}^{\rm off}(\tau)$ are uniformly distributed over [0.8, 0.9] for $\tau \leq 500$, and over [0.7, 0.8] for $501 \leq \tau \leq 1000$. For both types of dynamic tasks, assume that the number of new tasks starting at each time slot is uniformly distributed over $\{0,1\}$, and the length of each dynamic task is uniformly distributed over $\{0,1,2,3,4\}$. Assume that the upper bound of energy consumption all dynamic tasks satisfies the uniform distribution over [0.2, 0.4]. For each type-1 dynamic task $(i,j)_{\mathcal{D}}$, assume that the utility coefficients $a_{ij} = -0.5$, and b_{ij} is uniformly distributed over [0.8, 1.2]. For each type-2 dynamic task $(i,j)_{\mathcal{F}}$, assume that the total energy consumption satisfy $R \times (T_{ij}^{e2} - T_{ij}^{s2} + 1)$ where R is uniformly distributed over [0.05, 0.15]. The maximum energy generation $G^m = 100.0$. Let the cost function of the energy source $C(g) = 0.05g^2 + 0.1g$. The threshold for outage probability $\epsilon = 0.001$. In the simulations, we ran three iterations of dual variable update for each update of primal variables.

We consider the optimization results of problem (18) for different log-barrier coefficients $\mu = 0.10, 0.50$, and 1.00, using the distributed Newton's method. Fig. 2 plots the values of the objective function against the number of primal variable iterations for the above values of μ. It is seen that a larger μ exhibits faster convergence but poorer final convergence performance, while a smaller μ exhibits slower convergence but better final convergence performance. This is because larger μ significantly changes the original problem [cf. (20) and (21)]. In the following we set the log-barrier parameter $\mu = 0.10$, and fast

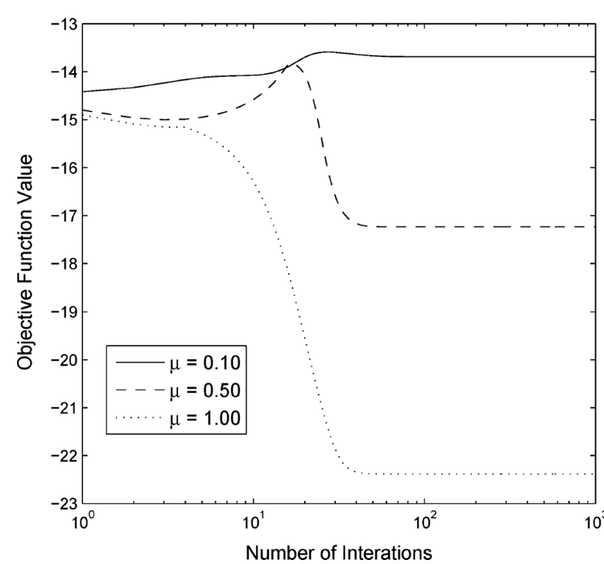

Fig. 2. Convergence of the proposed distributed Newton's algorithm under different μ.

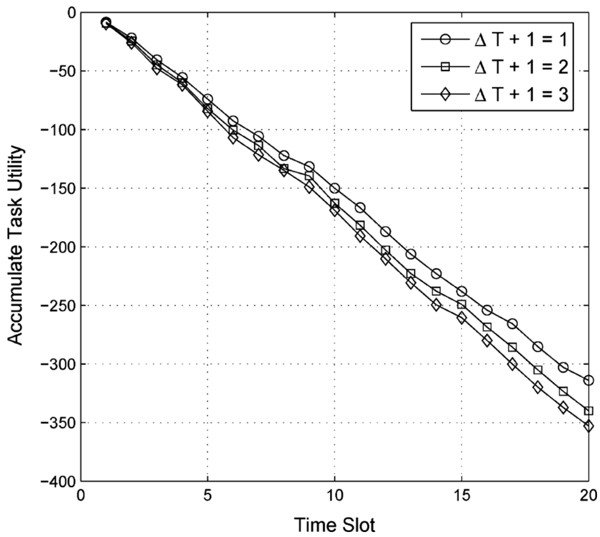

Fig. 3. The total system utility under different window lengths.

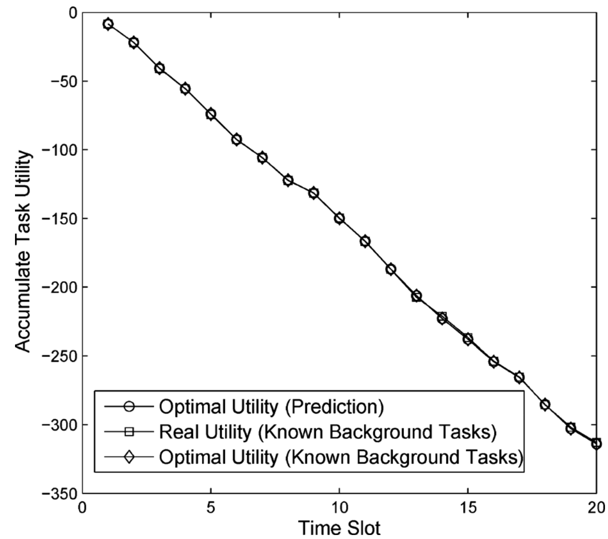

Fig. 4. Validity of Gaussian assumption.

convergence is always observed in all simulations. Usually 50 iterations are sufficient for convergence.

We now consider the solution to the optimization problem (18) using different window lengths in the rolling horizon optimization. Fig. 3 shows the final values of the objective function values for window lengths $\Delta T + 1 = 1, 2$ and 3 in time slots 1 to 20. The intuitive reason behind this observation is that at each certain time slot, when optimization is carried out there exists some uncertainties about the dynamic tasks generated after that slot (and thus unknown during optimization). Such uncertainties will increase by increasing the window length, which subsequently degrades the system performance. As a result, it is most efficient to set the window length to be one. In practice, the window length can be set to $\Delta T + 1 = 1$, i.e., the energy scheduling at each time slot is just for the current slot.

Finally, using $\Delta T + 1 = 1$, we test the fidelity of the Gaussian approximation for the background tasks. We consider a hypothetical case that the energy consumption of the background tasks are known, such that the energy generation g^τ and thus the energy cost $C(g^\tau)$ are both deterministic and known. In Fig. 4, we compare the optimization results based on such hypothetical deterministic model for the background tasks, and those based on the proposed stochastic model using Gaussian approximation. It is seen that the two results are very close, corroborating the validity of the Gaussian assumption employed in the proposed model.

Finally we adopt the distributed solution based on the dual decomposition method to solve the energy scheduling problem. Unfortunately, the dual decomposition-based distributed solution does not converge to a primal feasible solution. The convergence conditions of the dual decomposition-based distributed solution are not satisfied for the current optimization problem.

VII. Conclusion

We have developed a stochastic model for a local-area smart-grid network, and formulated an energy scheduling problem. To obtain a real-time fast solution, we have transformed the original stochastic optimization into a convex optimization with linear constraints using rolling horizon optimization and Gaussian approximation. Furthermore, we have developed a distributed Newton's method and designed the corresponding message passing mechanism between the energy source and energy consumers. Numerical results show that the proposed online energy scheduling framework has a fast convergence, and thus it is a promising practical solution for smart-grid systems.

Appendix

A. Justification of Gaussian Approximation

We first outline the Lindeberg condition [18] of central limit theorem for independent but not necessarily identically distributed random variables. Assume M random variables X_k, $1 \leq k \leq M$, with the expectation $\mathbb{E}(X_k) = \mu_k$ and the variance $\mathbb{D}(X_k) = \sigma_k^2$. Let $s_M^2 = \sum_{k=1}^{M} \sigma_k^2$. Then, the distribution of the following standard sums

$$t_M = \frac{\sum_{k=1}^{M}(X_k - \mu_k)}{s_M^2} \tag{46}$$

converges to the normal distribution $\mathcal{N}(0,1)$ if the following Lindeberg condition is satisfied:

$$\lim_{M \to +\infty} \frac{\sum_{k=1}^{M} \mathbb{E}\left((X_k - \mu_k)^2 \cdot \mathbf{1}_{|X_k - \mu_k| > \epsilon s_M}\right)}{s_M^2} = 0. \tag{47}$$

We prove that for any $1 \leq i \leq N$, $j \in \mathcal{B}_i$, and $1 \leq \tau \leq H$, $\delta \leq p_{ij}^\tau \leq 1 - \delta$. Note that $\delta \leq p_{ij}^1 \leq 1 - \delta$, and from (13) we have that

$$\begin{aligned} p_{ij}^\tau &\leq p_{ij}^{\tau-1}(1-\delta) + \left(1 - p_{ij}^{\tau-1}\right)(1-\delta) = 1 - \delta; \\ p_{ij}^\tau &\geq p_{ij}^{\tau-1}\delta + \left(1 - p_{ij}^{\tau-1}\right)\delta = \delta. \end{aligned} \tag{48}$$

On the other hand, for any $1 \leq \tau \leq H$, we have that for any background task $(i,j)_\mathcal{B}$, we have that $\mathbb{E}(z_{ij}^\tau) = p_{ij}^\tau B_{ij}$ and $\mathbb{D}(z_{ij}^\tau) = p_{ij}^\tau(1 - p_{ij}^\tau)B_{ij}^2$. Note that, since $\delta \leq p_{ij}^\tau \leq 1 - \delta$, we have that

$$\mathbb{D}\left(z_{ij}^\tau\right) = p_{ij}^\tau\left(1 - p_{ij}^\tau\right)B_{ij}^2 \geq \delta(1-\delta)B_{\min}^2. \tag{49}$$

Then, for M background tasks, we have that the sum variance $s_M^2 \geq M\delta(1-\delta)B_{\min}^2$. Note that, for any $\epsilon > 0$ and sufficient large M, $\epsilon s_M > B$ and thus $\mathbf{1}_{|z_{ij}^\tau - p_{ij}^\tau B_{ij}| > \epsilon s_M} = 0$, and thus the Lindeberg condition (47) is satisfied. Therefore, the Gaussian approximation for the sum of the background tasks is justified.

B. Proof of (26)

Then for $1 \leq i, j \leq L$, denote

$$\left(\boldsymbol{A}\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\right)_{ij} = \boldsymbol{a}_i \boldsymbol{H}_k^{-1}\boldsymbol{a}_j^T. \tag{50}$$

Hence $(\boldsymbol{D}_k)_{ii} = \boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}_i^T$. The elements of $(\bar{\boldsymbol{B}}_k - \boldsymbol{B}_k)$ are given by

$$(\bar{\boldsymbol{B}}_k - \boldsymbol{B}_k)_{ii} = \boldsymbol{a}_i\boldsymbol{H}_k^{-1}\sum_{l\neq i}\boldsymbol{a}_l^T = \boldsymbol{a}_i\boldsymbol{H}_k^{-1}(\boldsymbol{a}^T - \boldsymbol{a}_i),$$
$$\text{and} \quad (\bar{\boldsymbol{B}}_k - \boldsymbol{B}_k)_{ij} = -\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}_j^T, \quad \text{for } j \neq i. \tag{51}$$

Thus denoting $\boldsymbol{u}^k(t) = [u_i^k(t)]_{1\leq i\leq L}^T = (\bar{\boldsymbol{B}}_k - \boldsymbol{B}_k)\boldsymbol{w}^k(t)$ and $\boldsymbol{w}^k(t) = [w_i^k(t)]_{1\leq i\leq L}^T$, we have

$$\begin{aligned} u_i^k(t) &= \left(\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T\right) w_i^k(t) - \sum_{j=1}^{L}\left(\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}_j^T\right) w_j^k(t) \\ &= \left(\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T\right) w_i^k(t) - \boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}^k(t). \end{aligned} \tag{52}$$

Therefore, (23) can be rewritten as

$$w_i^k = w_i^k - \frac{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{A}^T\boldsymbol{w}^k}{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T} - \frac{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\nabla f(\boldsymbol{x}^k)}{\boldsymbol{a}_i\boldsymbol{H}_k^{-1}\boldsymbol{a}^T}, \quad \text{for } 1 \leq i \leq L. \tag{53}$$

C. Proof of Theorem 2

Consider the following linear constraints:

$$\sum_{j=1}^{J} a_{ij}x_j = c_i, \text{ for } 1 \leq i \leq I. \tag{54}$$

For each $1 \leq i \leq I$, let $\mathcal{A}_i^+ = \{j : a_{ij} > 0\}$ and $\mathcal{A}_i^- = \{j : a_{ij} < 0\}$. Then (54) can be rewritten as

$$\sum_{j\in\mathcal{A}_i^+} a_{ij}x_j + \sum_{j\in\mathcal{A}_i^-} a_{ij}x_j = c_i, \text{ for } 1 \leq i \leq I. \tag{55}$$

Let $\mathcal{A}^- = \cup_{i=1}^{I}\mathcal{A}_i^-$. The key step is to introduce auxiliary variables $y_j = x_j$ for $j \in \mathcal{A}^-$, and replace x_j with y_j for $j \in \mathcal{A}_i^-$ in constraint $\sum_{j\in\mathcal{A}_i^+} a_{ij}x_j + \sum_{j\in\mathcal{A}_i^-} a_{ij}x_j = c_i$. Therefore we have

$$\sum_{j\in\mathcal{A}_i^+} a_{ij}x_j + \sum_{j\in\mathcal{A}_i^-} a_{ij}y_j = c_i, \text{ for } 1 \leq i \leq I, \tag{56}$$

with the additional linear constraints

$$x_j - y_j = 0, \text{ for } j \in \mathcal{A}^-. \tag{57}$$

Thus, the signs of x_j are a_{ij} for $j \in \mathcal{A}_i^+$ in (56) and 1 in (57) and thus all positive; the signs of y_j are a_{ij} for $j \in \mathcal{A}_i^-$ in (56) and -1 in (57) and thus all negative. In this way, the original arbitrary linear constraint (54) is transformed into the form of (56) and (57) such that for each column of $\tilde{\boldsymbol{A}}$ all coefficients are of the same sign.

D. Computation of Auxiliary Variables

Let $\bar{v}_{i,\tau}^{(2)} = \sum_{j\in\mathcal{D}_i^\tau}(\nabla_1(x_{ij}^\tau))/(\nabla_2(x_{ij}^\tau)) + \sum_{j\in\mathcal{F}_i^\tau}(\nabla_1(y_{ij}^\tau))/(\nabla_2(y_{ij}^\tau))$. Then, for $v_{i,j}^{(1)}$, $v_{i,j}^{(2)}$, $v_{i,j,\tau}^{(3)}$, $v_{i,j,\tau}^{(4)}$, $v_\tau^{(5)}$, and $v_\tau^{(6)}$, we have

$$\begin{aligned}
v_{i,j}^{(1)} &= \sum_{\tau\in\mathcal{T}_{ij}^1}\frac{\nabla_1\left(x_{ij}^\tau\right)}{\nabla_2\left(x_{ij}^\tau\right)} - \frac{\nabla_1(s_{ij}(T))}{\nabla_2(s_{ij}(T))}, \\
&\qquad 1 \leq i \leq N,\ j \in \cup_{T\leq\tau\leq T+\Delta T}\mathcal{D}_i^\tau; \\
v_{i,j}^{(2)} &= \sum_{\tau\in\mathcal{T}_{ij}^2}\frac{\nabla_1\left(y_{ij}^\tau\right)}{\nabla_2\left(y_{ij}^\tau\right)}, \\
&\qquad 1 \leq i \leq N,\ j \in \cup_{T\leq\tau\leq T+\Delta T}\mathcal{F}_i^\tau; \\
v_{i,j,\tau}^{(3)} &= \frac{\nabla_1\left(x_{ij}^\tau\right)}{\nabla_2\left(x_{ij}^\tau\right)} + \frac{\nabla_1\left(m_{ij}^\tau\right)}{\nabla_2\left(m_{ij}^\tau\right)}, \\
&\qquad 1 \leq i \leq N,\ j \in \cup_{T\leq\tau\leq T+\Delta T}\mathcal{D}_i^\tau; \\
v_{i,j,\tau}^{(4)} &= \frac{\nabla_1\left(y_{ij}^\tau\right)}{\nabla_2\left(y_{ij}^\tau\right)} + \frac{\nabla_1\left(n_{ij}^\tau\right)}{\nabla_2\left(n_{ij}^\tau\right)}, \\
&\qquad 1 \leq i \leq N,\ j \in \cup_{T\leq\tau\leq T+\Delta T}\mathcal{F}_i^\tau; \\
v_\tau^{(5)} &= \sum_{i=1}^{N}\sum_{j\in\mathcal{D}_i^\tau}\frac{\nabla_1\left(x_{ij}^\tau\right)}{\nabla_2\left(x_{ij}^\tau\right)} \\
&\quad + \sum_{i=1}^{N}\sum_{j\in\mathcal{F}_i^\tau}\frac{\nabla_1\left(y_{ij}^\tau\right)}{\nabla_2\left(y_{ij}^\tau\right)} - \frac{\nabla_1(h^\tau)}{\nabla_2(h^\tau)} \\
&= \sum_{i=1}^{N}\bar{v}_{i,\tau}^{(2)} - \frac{\nabla_1(h^\tau)}{\nabla_2(h^\tau)},\ T \leq \tau \leq T+\Delta T; \\
v_\tau^{(6)} &= \sum_{i=1}^{N}\sum_{j\in\mathcal{D}_i^\tau}\frac{\nabla_1\left(x_{ij}^\tau\right)}{\nabla_2\left(x_{ij}^\tau\right)} \\
&\quad + \sum_{i=1}^{N}\sum_{j\in\mathcal{F}_i^\tau}\frac{\nabla_1\left(y_{ij}^\tau\right)}{\nabla_2\left(y_{ij}^\tau\right)} + \frac{\nabla_1(r^\tau)}{\nabla_2(r^\tau)} \\
&= \sum_{i=1}^{N}\bar{v}_{i,\tau}^{(2)} + \frac{\nabla_1(r^\tau)}{\nabla_2(r^\tau)},\ T \leq \tau \leq T+\Delta T.
\end{aligned} \tag{58}$$

Let $\bar{K}_{i,\tau}^{(2)} = \sum_{j\in\mathcal{D}_i^\tau}\nabla_2^{-1}(x_{ij}^\tau) + \sum_{j\in\mathcal{F}_i^\tau}\nabla_2^{-1}(y_{ij}^\tau)$. Then, for $K_{i,j}^{(1)}$, $K_{i,j}^{(2)}$, $K_{i,j,\tau}^{(3)}$, $K_{i,j,\tau}^{(4)}$, $K_\tau^{(5)}$, and $K_\tau^{(6)}$, we have

$$\begin{aligned}
K_{i,j}^{(1)} &= 4\sum_{\tau\in\mathcal{T}_{ij}^1}\nabla_2^{-1}\left(x_{ij}^\tau\right) + \nabla_2^{-1}(s_{ij}), \\
&\qquad 1 \leq i \leq N,\ j \in \cup_{T\leq\tau\leq T+\Delta T}\mathcal{D}_i^\tau \\
K_{i,j}^{(2)} &= 4\sum_{\tau\in\mathcal{T}_{ij}^2}\nabla_2^{-1}\left(y_{ij}^\tau\right),
\end{aligned}$$

$$1 \le i \le N,\ j \in \cup_{T \le \tau \le T+\Delta T} \mathcal{F}_i^\tau$$

$$K_{i,j,\tau}^{(3)} = 4\nabla_2^{-1}\left(x_{ij}^\tau\right) + \nabla_2^{-1}\left(m_{ij}^\tau\right),$$
$$1 \le i \le N,\ T \le \tau \le T+\Delta T,\ j \in \mathcal{D}_i^\tau$$

$$K_{i,j,\tau}^{(4)} = 4\nabla_2^{-1}\left(y_{ij}^\tau\right) + \nabla_2^{-1}\left(n_{ij}^\tau\right),$$
$$1 \le i \le N,\ T \le \tau \le T+\Delta T,\ j \in \mathcal{F}_i^\tau,$$

$$K_\tau^{(5)} = 4\sum_{i=1}^N \left(\sum_{j\in\mathcal{D}_i^\tau} \nabla_2^{-1}\left(x_{ij}^\tau\right) + \sum_{j\in\mathcal{F}_i^\tau} \nabla_2^{-1}\left(y_{ij}^\tau\right)\right) + \nabla_2^{-1}(h^\tau)$$
$$= 4\sum_{i=1}^N \bar{K}_{i,\tau}^{(2)} + \nabla_2^{-1}(h^\tau),\ T \le \tau \le T+\Delta T,$$

$$K_\tau^{(6)} = 4\sum_{i=1}^N \left(\sum_{j\in\mathcal{D}_i^\tau} \nabla_2^{-1}\left(x_{ij}^\tau\right) + \sum_{j\in\mathcal{F}_i^\tau}\right) \nabla_2^{-1}\left(y_{ij}^\tau\right) + \nabla_2^{-1}(r^\tau)$$
$$= 4\sum_{i=1}^N \bar{K}_{i,\tau}^{(2)} + \nabla_2^{-1}(r^\tau),\ T \le \tau \le T+\Delta T. \tag{59}$$

Let $\bar{L}_{i,\tau}^{(2)} = \sum_{j\in\mathcal{D}_i^\tau}(w_{i,j}^{(1)} + w_{i,\tau}^{(2)})\nabla_2^{-1}(x_{ij}^\tau)$. Then, for $L_{i,j}^{(1)}$, $L_{i,j}^{(2)}$, $L_{i,j,\tau}^{(3)}$, $L_{i,j,\tau}^{(4)}$, $L_\tau^{(5)}$, and $L_\tau^{(6)}$, we have

$$L_{i,j}^{(1)} = \sum_{\tau\in\mathcal{T}_{ij}^1}\left(w_{i,j}^{(1)} + w_{i,j}^{(3)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(x_{ij}^\tau\right) + w_{i,j}^{(1)}\nabla_2^{-1}(s_{ij}(T)),$$
$$\text{for } 1 \le i \le N,\ j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{D}_i^\tau;$$

$$L_{i,j}^{(2)} = \sum_{\tau\in\mathcal{T}_{ij}^2}\left(w_{i,j}^{(2)} + w_{i,j}^{(4)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(y_{ij}^\tau\right)$$
$$\text{for } 1 \le i \le N,\ j \in \cup_{T\le\tau\le T+\Delta T}\mathcal{F}_i^\tau;$$

$$L_{i,j,\tau}^{(3)} = \left(w_{i,j}^{(1)} + w_{i,j}^{(3)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(x_{ij}^\tau\right)$$
$$\text{for } 1 \le i \le N,\ T \le \tau \le T+\Delta T,\ j \in \mathcal{D}_i^\tau;$$

$$L_{i,j,\tau}^{(4)} = \left(w_{i,j}^{(2)} + w_{i,j}^{(4)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(y_{ij}^\tau\right)$$
$$\text{for } 1 \le i \le N,\ T \le \tau \le T+\Delta T,\ j \in \mathcal{F}_i^\tau;$$

$$L_\tau^{(5)} = \sum_{i=1}^N\sum_{j\in\mathcal{D}_i^\tau}\left(w_{i,j}^{(1)} + w_{i,j}^{(3)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(x_{ij}^\tau\right) + \sum_{i=1}^N\sum_{j\in\mathcal{F}_i^\tau}\left(w_{i,j}^{(2)} + w_{i,j}^{(4)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(y_{ij}^\tau\right) + w_\tau^{(5)}\nabla_2^{-1}(h^\tau)$$
$$= \sum_{i=1}^N \bar{L}_{i,\tau}^{(2)} + \left(w_\tau^{(5)} + w_\tau^{(6)}\right)\sum_{i=1}^N \bar{K}_{i,\tau}^{(2)} + w_\tau^{(5)}\nabla_2^{-1}(h^\tau),$$
$$\text{for } T \le \tau \le T+\Delta T; \tag{60}$$

$$L_\tau^{(6)} = \sum_{i=1}^N\sum_{j\in\mathcal{D}_i^\tau}\left(w_{i,j}^{(1)} + w_{i,j}^{(3)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(x_{ij}^\tau\right) + \sum_{i=1}^N\sum_{j\in\mathcal{F}_i^\tau}\left(w_{i,j}^{(2)} + w_{i,j}^{(4)} + w_\tau^{(5)} + w_\tau^{(6)}\right)\nabla_2^{-1}\left(y_{ij}^\tau\right) + w_\tau^{(6)}\nabla_2^{-1}(r^\tau)$$
$$= \sum_{i=1}^N \bar{L}_{i,\tau}^{(2)} + \left(w_\tau^{(5)} + w_\tau^{(6)}\right)\sum_{i=1}^N \bar{K}_{i,\tau}^{(2)} + w_\tau^{(6)}\nabla_2^{-1}(r^\tau),$$
$$\text{for } T \le \tau \le T+\Delta T; \tag{61}$$

Finally from (44) we have

$$(\theta^k)^2 = \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^N\sum_{j\in\mathcal{D}_i^T}\left(\Delta x_{i,j}^\tau\right)^2\nabla_2\left(x_{i,j}^\tau\right) + \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^N\sum_{j\in\mathcal{F}_i^T}\left(\Delta y_{i,j}^\tau\right)^2\nabla_2\left(y_{i,j}^\tau\right) + \sum_{\tau=T}^{T+\Delta T}\sum_{i=1}^N(\Delta s_{ij}(T))^2\nabla_2(s_{ij}(T)) + \sum_{i=1}^N\sum_{j\in\mathcal{D}_i^T}\left(\Delta m_{ij}^\tau\right)^2\nabla_2\left(m_{ij}^\tau\right) + \sum_{i=1}^N\sum_{j\in\mathcal{F}_i^T}\left(\Delta n_{ij}^\tau\right)^2\nabla_2\left(n_{ij}^\tau\right) + \sum_{\tau=T}^{T+\Delta T}(\Delta h^\tau)^2\nabla_2(h^\tau) + \sum_{\tau=T}^{T+\Delta T}(\Delta r^\tau)^2\nabla_2(r^\tau). \tag{62}$$

References

[1] D. Pendarakis, N. Shrivastava, L. Zhen, and R. Ambrosio, "Information aggregation and optimized actuation in sensor networks: Enabling smart electrical grids," in *Proc. Int. Conf. Comput. Commun. (INFOCOM)*, Anchorage, AL, USA, May 2007, pp. 2386–2390.

[2] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart transmission grid: Vision and framework," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 168–177, Sep. 2010.

[3] J. Medina, N. Muller, and I. Roytelman, "Demand response and distribution grid operations: Opportunities and challenges," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 193–198, Sep. 2010.

[4] A. J. Conejo, J. M. Morales, and L. Baringo, "Real-time demand response model," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec. 2010.

[5] A. H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 120–133, Sep. 2010.

[6] M. Pedrasa, T. Spooner, and I. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 134–143, Sep. 2010.

[7] D. O. N. , M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *Proc. IEEE Int. Conf. Smart Grid Commun. 2010*, Gaithersburg, MD, USA, Oct. 2010.

[8] L. Chen, N. Li, S. H. Low, and J. C. Doyle, "Two market models for demand response in power networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. 2010*, Gaithersburg, MD, USA, Oct. 2010.

[9] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *Proc. IEEE Int. Conf. Smart Grid Commun. 2010*, Gaithersburg, MD, USA, Oct. 2010.

[10] M. Parvania and M. Fotuhi-Firuzabad, "Demand response scheduling by stochastic SCUC," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 89–98, Jun. 2010.

[11] K. D. Le, "Rolling horizon method: A new optimization technique for generation expansion studies," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 9, pp. 3112–3116, Sep. 1982.

[12] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mamthematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[13] N. Gatsis and G. Giannakis, "Residential load control: Distributed scheduling and convergence with lost ami messages," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 770–786, Jun. 2012.

[14] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *Soc. Ind. Appl. Math. (SIAM) Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[15] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed network method for network utility maximization," May 14, 2010 [Online]. Available: arXiv:1002.2633v1 [Math. OC]

[16] A. J. Conejo, F. J. Nogales, and F. J. Prieto, "A decomposition procedure based on approximate newton directions," *Math. Program.*, vol. 93, no. 3, pp. 495–515, Dec. 2002.

[17] F. J. Nogales, F. J. Prieto, and A. J. Conejo, "A decomposition methodology applied to the multi-area optimal power flow problem," *Ann. Oper. Res.*, vol. 120, no. 1–4, pp. 99–116, Apr. 2003.

[18] J. W. Lindeberg, "Eine neue herleitung des exponentialgesetzes in der wahrscheinlichkeitsrechnung," *Mathematische Zeitschrift*, vol. 15, no. 1, pp. 211–225, 1922.

Chen Gong received the B.S. degree in electrical engineering and mathematics (minor) from Shanghai Jiaotong University, Shanghai, China, in 2005, M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2008, and the Ph.D. degree from Columbia University, New York, in 2012. He received Jury Award from Columbia University for his outstanding contribution in the area of signal processing and communication during his Ph.D. study. His research interests are in the area of transmission and signal processing techniques for wireless and optical communications, as well as energy scheduling in smart grid systems. Now he is a senior system engineer in Qualcomm Research San Diego, Qualcomm Inc., San Diego, CA, USA.

Xiaodong Wang (S'98–M'98–SM'04–F'08) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA. He is a Professor of Electrical Engineering at Columbia University, New York. Dr. Wang's research interests fall in the general areas of computing, signal processing and communications, and has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception*, published by Prentice Hall in 2003. His current research interests include wireless communications, statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award, and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He has served as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY.

Weiqiang Xu (M'09) received his M.Sc. degree in communications and information system from Southwest Jiao-Tong University, China, and his Ph.D. degree in control science and engineering from Zhejiang University, China, in 2003 and 2006, respectively. He also was a Postdoctoral Research Fellow with the group of Networked Sensing and Control in the State Key laboratory of Industrial Control Technology, Zhejiang University, China. From October 2009 to October 2010, he visited Prof. Xiaodong Wang's research group in Electrical Engineering Department at Columbia University, New York. He is currently a professor with the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China. His research interests include multi-cell networks, Ad Hoc networks, wireless sensor networks, wireless optical networks, congestion control, and smart grid, etc. He has served as a TPC member for IEEE ICC 2013, IEEE Globecom 2012, IWCMC 2009, IWCMC 2010, PMSN 2009, IHMSC 2009, IHMSC 2010, IHMSC 2011, IHMSC 2012. He has also served as a peer reviewer for a variety of IEEE journals and conferences.

Ali Tajer (S'05–M'10) received the B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Iran, and the M.A. degree in statistics and the Ph.D. in electrical engineering from Columbia University, New York. During 2010–2012 he was a Postdoctoral Research Associate at Princeton University and an Adjunct Assistant Professor at Columbia University. Since August 2012 he has been an Assistant Professor of Electrical and Computer Engineering at Wayne State University, Detroit, MI, USA. His research interests lie in the general areas of network information theory, applied statistics, and energy systems.