A Fast Distributed Algorithm for Coupled Utility Maximization Problem with Application for Power Control in Wireless Sensor Networks

Shengbin Liao

Abstract: **This paper investigates a method to distributively solve a Network Utility Maximization (NUM) problem with coupled variables and applies it to study power control in wireless sensor networks (WSNs). We present a dual decomposition-based consistency price algorithm to solve the coupled problem. However, the consistency price algorithm suffers from slow convergence. We then propose a two-step method to address the given issue. The first step is to build up a global consensus problem by introducing slack variables to transform the NUM problem with globally coupled variables into a NUM problem with coupled constraints. The second step is to design a distributed algorithm that combines the first-order gradient/subgradient method and a local consensus algorithm to solve the global consensus problem. The proposed algorithm is a primary algorithm which has faster convergence speed than the consistency price algorithm which is a primary-dual algorithm. Experimental results have demonstrated the effectiveness of our proposed approach.**

Index Terms: **Consensus algorithm, network utility maximization, power control, primary-dual algorithm, wireless sensor networks.**

I. INTRODUCTION

SINCE the publication of the seminal paper [1] by Kelly *et al.*, the framework of network utility maximizaiton (NUM) has attracted a lot of attention. Many important network design and resource allocation problems can be formulated as a NUM model. The utility concept, originally proposed in economics, is used to measure the satisfaction degree of a consumer for a good or service. In the basic NUM model, the utility of a network user is defined as a function of the data rate. The goal of network system is designed to maximize the overall utility of all the users in the network.

Consider a communication network with L links, each with a fixed capacity of c_l bps. Let a route r be a non-empty subset of L, and $\mathcal{R}$ be the set of possible routes. Set $A_{lr} = 1$ if $l \in r$, so that the link l lies on route r, and set $A_{lr} = 0$ otherwise. This defines a 0-1 routing matrix $A = (A_{lr}, l \in L, r \in \mathcal{R})$.

Associate a route r with a user (or a data source), and suppose that if a rate x_r is allocated to user r then this has utility $U_r(x_r)$ to the user. Assume that the utility $U_r(x_r)$ is increasing, strictly concave and continuously differentiable over the range $x_r \geq 0$. Let $U = (U_r(x_r), r \in \mathcal{R})$ and $C = (c_l, l \in L)$. Under this model the network seeks a rate allocation $\boldsymbol{x} = (x_r, r \in \mathcal{R})$ which solves the following optimization problem [2].

$$\begin{array}{ll} \max_{\boldsymbol{x}} & \sum\limits_{r\in\mathcal{R}} U_r(x_r) \\ \text{subject to} & A\boldsymbol{x} \leq C \\ & \boldsymbol{x} \geq 0 \end{array} \tag{1}$$

In the basic NUM model (1), the utility $U_r(x_r)$ of a network user r is defined as function of its data rate x_r, this means that all utility functions are separable. Due to the characteristic of separability of utility functions, a basic distributed NUM algorithm is derived to maximize aggregate user utility by the dual decomposition theory [3]. Along this way, so many extended NUM models and resultant distributed algorithms have been proposed for network architectural decisions, cross-layer optimization and resource allocation in wireless as well as wireline networks [4]-[10].

However, the conventional methods for designing distributed NUM algorithms do not work when utility functions are globally coupled. In this case the utility of a network user r not only depends on its own data rate x_r (or the allocated resource) but also on the rates $(x_s, s \neq r, s \in \mathcal{R})$ of other users. Associated with the allocated network resource $\boldsymbol{x} = (x_r, r \in \mathcal{R})$, each user r obtains a utility $U_r(\boldsymbol{x})$, where x_r is the allocated rate for user r and $U_r(\boldsymbol{x})$ is its utility function. By maximizing the aggregate utility functions under the resource constraints, we can establish the corresponding NUM model as follows.

$$\begin{array}{ll} \max_{\boldsymbol{x}} & \sum\limits_{r\in\mathcal{R}} U_r(\boldsymbol{x}) \\ \text{subject to} & A\boldsymbol{x} \leq C \\ & \boldsymbol{x} \geq 0 \end{array} \tag{2}$$

Some important network design and resource allocation problems, such as power control in wireless sensor networks (WSNs) [11] and spectrum management for digital subscriber lines [12], can be formulated as model (2). Model (2) is a globally coupled NUM problem because the utility function of each user not only depends on the allocated resource but also on the resources obtained by other users. In classical NUM model (1), the utility function is usually defined as $U_s(x_s)$ for user s, which only depends on local variable x_s. The globally coupling feature in model (2) makes it difficult to find the globally optimal resource allocation in a distributed setting.

The common methods to address the coupled NUM model (2) are Lagrangian duality theory and game theory. Based on

Manuscript received January 23, 2021; revised May 9, 2021; approved for publication by Hongjian Sun, Division III Editor, June 26, 2021.

This research was supported by the NSF of China under grants 62077023 and 61937001.

S. Liao is with the National Engineering Center for e-learning, Central China Normal University, email: liaoshengbin@mail.ccnu.edu.cn.

S. Liao is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2021.000022

Lagrangian duality theory, distributed power control schemes were developed in [13] and [14] for some special cases of the model (2). However, these works suffer from the slow convergence speed. There are some works which have investigated the problem by using game theory [17], [18]. These works usually introduce a "weighted parameter", which can be interpreted as "power price". The aglorithms developped in these works are either centralized or they also suffer slow convergence speed because of the update of the dual variables(or power prices). Our approach differs in that: 1) We focus on developping a distributed algorithm which uses the primary variables instead of dual variables. 2) our algorithm combines the first-order gradient method and the local consensus algorithm. 3) the users in our method cooperate with their neighbors by exchanging the information of the primary variables instead of the information of the dual variables such as "power prices".

In this paper, we present a *two-step method* to address the given issue (2). The first step is to build up a global consensus problem [19] by introducing slack variables to transform the NUM problem with globally coupled variables into a NUM problem with coupled constraints. The global consensus problem can be distributively solved by the consistency price approach [20]. The consistency price approach is a kind of primary-dual algorithms and also suffers from the slow rate of convergence [21], [22].

The second step is to design a distributed algorithm that combines the first-order gradient/subgradient method and the local consensus algorithm [23] to solve the global consensus problem. Our proposed algorithm is a primary algorithm which has faster convergence speed compared with the conventional primary-dual algorithm.

In this paper, we apply our proposed algorithm to study power control in WSNs. Optimizing the performance of WSNs (i.e., maximizing the overall utility) is global coupling between the mutual interference of wireless channels. In typical wireless sensor network applications, sensor nodes are energy-constrained and are randomly distributed in a certain region without control center. Therefore, distributively solving the coupled power control problem for WSNs are very important. Our main contribution is that we present a mixed method to address the given issue effectively.

The outline of this paper is as follows. We first discuss the background and related work in Section II, then present the primary-dual algorithm for solving a global consensus problem in Section III. In Section IV, we apply the first-order method and local consensus to obtain a distributed primary algorithm. In Section V, we apply the proposed algorithm to study the power control in WSNs and provide experimental results. Finally, Section VI concludes this paper.

II. BACKGROUND AND RELATED WORK

Before we present our idea, we first introduce a basic distributed optimization algorithm [2] which solves the model (1). The consistency price approach presented in the next section belongs to extensions of this work.

A. Basic Primary-dual Algorithm

Low *et al.* present in [2] the following basic distributed optimization algorithm for solving the model (1).

The Lagrangian dual function for problem (1) is

$$\begin{aligned} D(\mu) &= \max_{x_r\geq 0}\Big\{\sum_{r\in\mathcal{R}} U_r(x_r) - \mu^T(Ax-C)\Big\} \\ &= \max_{x_r\geq 0}\Big\{\sum_{r\in\mathcal{R}}(U_r(x_r) - x_r\sum_{l\in L} A_{lr}\mu_l)\Big\} + \mu^T C \\ &= \sum_{r\in\mathcal{R}}\Big\{\max_{x_r\geq 0}(U_r(x_r) - x_r\sum_{l\in L} A_{lr}\mu_l)\Big\} + \mu^T C, \end{aligned} \tag{3}$$

where $\mu = (\mu_l, l \in L)$ is a vector of lagrange multipliers. Here, the second equality follows due to the definition of the matrix $\mathbf{A}$. Thus, the dual problem for primal problem (1) is

$$\min_{\mu\geq 0} \quad D(\mu). \tag{4}$$

In the dual formulation, Lagrange multiplier μ_l can be interpreted as congestion price on link l. A key observation from equation (3) is that sources can compute their optimal rate individually, based on the total congestion price $\sum_{l\in L} A_{lr}\mu_l$, using the following source rate algorithm

$$x_r = \arg\max_{x_r\geq 0}\Big\{\sum_{r\in\mathcal{R}} U_r(x_r) - x_r\sum_{l\in L} A_{lr}\mu_l\Big\}. \tag{5}$$

To solve the dual problem (4), one can use the following projected gradient method

$$\mu_l(t+1) = \Big[\mu_l(t) - \alpha(t)(c_l - \sum_{r\in\mathcal{R}} A_{lr}x_r)\Big]^+, \tag{6}$$

where $\alpha(t)$ is a positive scalar stepsize, and $[a]^+$ denotes the projection of a onto the set of non-negative real numbers.

According to the duality theory [24] and the assumption that the utility $U_r(x_r)$ is increasing, strictly concave and continuously differentiable over the range $x_r \geq 0$, the optimal solutions to both primal problem (1) and dual problem (4) can be found simultaneously by solving iteratively in equation (5) and (6), respectively. This suggests treating the network links and the sources as processors in a distributed computation system to solve the primary problem (1) and the dual problem (4). The algorithm (5)–(6) is often referred to as the *basic primary-dual algorithm.* The algorithm (5)–(6) is a distributed primary-dual algorithm. Here the word "distributed" means that the algorithm relies only on local information exchange between nodes. A large number of studies based on NUM framework belong to extensions of the basic primary-dual algorithm which has been developed into a mathematical theory of network architectures, interested readers along this line please refer to [25].

B. Related Work

Many research studies have been carried out on the topic of NUM framework [26]. A number of modified and generalized NUM models and distributed algorithms have been proposed in the past two decades [4]–[10]. Most of previous work defined the utility of a network user as a function of its transmission rate, this means that the objective functions of the model are separable. However, we study NUM with coupled utility functions. Defining different utility functions naturally lead to different NUM frameworks, there exist some challenges to define utility functions, the relationship between utility functions and performance metrics is presented in [28]–[29].

There are some studies which have also investigated the coupled NUM issues. Ruilong *et al.* [30] considered NUM problem with coupled constraints. In contrast, we considered NUM problem with coupled utilities. Cendrillon *et al.* [31] constructed a dynamic spectrum management model, which is a special case of NUM with coupled variables. They used the weighted sum method to decouple the model, i.e., each user optimizes the weighted sum of the achievable rates on its own and other users. Shan *et al.* [32] presented a adaptive transmission power control algorithm in WSNs, in which each node builds a model for each of its neighbors, describing the correlation between transmission power and link quality. Each node need employ a feedback-based transmission power control algorithm to dynamically maintain individual link quality over time.

In [13]–[16], the same problem is also considered, which are most close to our work. In [13], Lei *et al.* first transform the coupled NUM into a structured max-min problem by introducing an auxiliary variable. They then use Extended duality [33] and simulated annealing [34] to devise a distributed NUM algorithm. In contrast, in this paper, we transform the coupled NUM into a global consensus problem. Moreover, we do not adopt randomized approaches to construct our algorithm. Their distributed algorithm can obtain global optimality at the cost of slow convergence due to the use of simulated annealing.

In [14], Tan *et al.* also transform the coupled NUM into a global consensus problem, this is similar to ours. A significant difference from our work is that they use the consistency price approach, which is a kind of primary-dual algorithms, to solve the global consensus problem. We design a mixed algorithm to solve distributively the transformed problem. The proposed algorithm combines the ideas of the first-order gradient/subgradient algorithm and the local consensus algorithm and obtains the fast convergence rate.

Our proposed algorithm is a kind of cooperative distributed optimization as it depends on some information exchanges between neighbour users. In [23], Nedić *et al.* introduce the basic idea and algorithms of cooperative distributed multi-agent optimization. They also present a distributed algorithm which combines the first-order gradient/subgradient algorithm and the local consensus algorithm. They use stochastic weight matrix to model information exchange in the local consensus algorithm, this work also suffers from slow convergence speed. Our proposed algorithm uses the deterministic weight sum approach in the local consensus algorithm and the weight coefficients can be determined by the number of neighbour users (including itself) or the rates of neighbour users.

III. CONSISTENCY PRICE ALGORITHM

Here, a key idea to tackle coupled utilities is to introduce multiple slack variables and build up a global consensus problem [19]. By introducing slack variables $(\boldsymbol{y}_r, r \in \mathcal{R})$ and regarding primary variable $\boldsymbol{x}$ as global common variable, the optimization problem (2) is then formulated equivalently as

$$\begin{aligned} &\max_{\{\boldsymbol{y}_r,\boldsymbol{x}\}} \quad \textstyle\sum_{r\in\mathcal{R}} U_r(\boldsymbol{y}_r) \\ &\text{subject to} \\ &\qquad \boldsymbol{y}_r = \boldsymbol{x}, \qquad \forall r \in \mathcal{R} \\ &\qquad A\boldsymbol{x} \le C \\ &\qquad \boldsymbol{x} \ge 0, \end{aligned} \tag{7}$$

where $\boldsymbol{y}_r = (y_{r,1}, \cdots, y_{r,r-1}, x_r, y_{r,r+1}, \cdots, y_{r,R})$, but for notation simplicity, let $\boldsymbol{y}_r = (y_{r,j}, j \in \mathcal{R})$, this means $y_{r,r} = x_r$.

The objective functions are separable in the transformed problem (7) by adding an equality constraint. The problem can be distributively solved by the conventional primary-dual gradient/subgradient algorithm which is similar to the *basic primary-dual algorithm*. We write down the Lagrangian associated with problem (7) as follows,

$$\begin{aligned} L(\boldsymbol{y}_r, \boldsymbol{x}; \mu, \lambda_r) &= \sum_{r\in\mathcal{R}} (U_r(\boldsymbol{x}_r) + {\lambda_r}^T(\boldsymbol{y}_r - \boldsymbol{x})) \\ &\quad + \mu^T(C - A\boldsymbol{x}) \\ &= \sum_{r\in\mathcal{R}} (U_r(\boldsymbol{y}_r) + {\lambda_r}^T \boldsymbol{y}_r \\ &\quad - x_r \sum_{i\in\mathcal{R}} \lambda_{ir} - x_r \sum_{l\in L} A_{lr}\mu_l) + \mu^T C, \end{aligned}$$

where $\mu = (\mu_l, l \in L)$ and $\lambda_r = (\lambda_{rj}, j \in \mathcal{R})$ are lagrange multiplier vectors which are respectively associated with the inequality and equality constraints in model (7). From the above expression, the Lagrangian can be separated into many subproblems, where each subproblem uses only local variables, i.e., the *r*th subproblem uses only variables with the first subscript index *r*. The dual function is given by

$$\begin{aligned} D(\mu, \lambda_r) &= \max_{\{\boldsymbol{y}_r,\boldsymbol{x}\}} L(\boldsymbol{y}_r, \boldsymbol{x}; \mu, \lambda_r) \\ &= \max_{\{\boldsymbol{y}_r,\boldsymbol{x}\}} \sum_{r\in\mathcal{R}} (U_r(\boldsymbol{y}_r) + {\lambda_r}^T \boldsymbol{y}_r - x_r \sum_{i\in\mathcal{R}} \lambda_{ir} \\ &\quad - x_r \sum_{l\in L} A_{lr}\mu_l), \end{aligned}$$

where the second equality follows since the item $\mu^T C$ in the Lagrangian does not depend on variables $\boldsymbol{y}_r$ and $\boldsymbol{x}$. Thus, the dual problem for model (7) is given by

$$\min_{\mu\ge 0, \lambda_r \ge 0} \quad D(\mu, \lambda_r). \tag{8}$$

Since the Lagrangian is separable, this maximization of Lagrangian over $(\boldsymbol{y}_r, \boldsymbol{x})$ can be conducted in parallel at each source *r* as follows,

$$\max_{\{\boldsymbol{y}_r,\boldsymbol{x}\}} U_r(\boldsymbol{y}_r) + {\lambda_r}^T \boldsymbol{y}_r - x_r \sum_{i\in\mathcal{R}} \lambda_{ir} - x_r \sum_{l\in L} A_{lr}\mu_l. \tag{9}$$

While the primal problem (7) is not strictly convex in the primal variables $\{\boldsymbol{y}_r, \boldsymbol{x}\}$, the dual function $D(\mu, \lambda_r)$ is usually piecewise differentiable and corresponding dual problem is a non-differentiable convex optimization problem. Hence we will solve the dual problem (8) using subgradient projection method [35]. Suppose $\{\boldsymbol{y}_r^*, \boldsymbol{x}^*\}$ is the solution of the above problem (9). It is easy to verify that

$$f_l = c_l - \sum_{l \in L} A_{lr} x_r^*$$

and

$$g_{rj} = y_{rj}^* - x_r^*,\ j \in \mathcal{R},$$

are the subgradient of the above dual function with respect to the dual variables μ and λ_r at point (μ, λ_r), respectively. Thus, the dual problem (8) can be solved by using the subgradient projection algorithm as

$$\mu_l(t+1) = \Big[\mu_l(t) - \alpha(t) f_l\Big]^+,\ l \in L$$

and

$$\lambda_{rj}(t+1) = \Big[\lambda_{rj}(t) - \beta(t) g_{rj}\Big]^+,\ r, j \in \mathcal{R},$$

where $\alpha(t)$ and $\beta(t)$ are positive stepsizes.

We now describe the following distributed algorithm, where each source r and each link l solve their own problems with only local information. We can interpret μ_l as the price per unit rate to use link l, and λ_{rj} as the price of the consistency deviation between the source r and j, accordingly, the algorithm is referred to *Consistency Price Algorithm: CPA*.

Consistency Price Algorithm: CPA:

- Step 0: Initialize the dual variables $\{\mu, \lambda_r\}$

At each iteration t

- Step 1: Each source r solves the following problem

$$\max_{\{\boldsymbol{y}_r, \boldsymbol{x}\}} U_r(\boldsymbol{y}_r) + {\lambda_r}^T \boldsymbol{y}_r - x_r \sum_{i \in \mathcal{R}} \lambda_{ir} - x_r \sum_{l \in L} A_{lr} \mu_l.$$

- Step 2: Update the subgradients of the dual function as

$$f_l = c_l - \sum_{l \in L} A_{lr} x_r^*$$

and

$$g_{rj} = y_{rj}^* - x_r^*,\ j \in \mathcal{R}$$

where $\{\boldsymbol{y}_r^*, \boldsymbol{x}^*\}$ is the solution from Step 1.

- Step 3: Each link l updates its price as

$$\mu_l(t+1) = \Big[\mu_l(t) - \alpha(t) f_l\Big]^+,\ l \in L,$$

where f_l is from Step 2 and $\alpha(t)$ is the positive scalar stepsize.

- Step 4: Each source r updates its consistency price as

$$\lambda_{rj}(t+1) = \Big[\lambda_{rj}(t) - \beta(t) g_{rj}\Big]^+,\ r, j \in \mathcal{R},$$

where λ_{rj} is from Step 2 and $\beta(t)$ is the positive scalar stepsize.

Remark 1. In CPA, there have some feedback information interaction. In Step 3, to update the link price, each link needs to know the aggregate data rate of the sources that are using it. This can be measured by the link itself. Hence, each link can adjust its price by the local information. In Step 1, to solve the given problem, each source needs to know the sum of the link prices along its routing path and all of the consistency prices. The link prices can be obtained by the notification from the links through the presence of acknowledgment packets in TCP [36]. However, it is difficult or inefficient for each source to obtained all of the consistency prices. This observation motivates us to devise a more efficient algorithm.

Remark 2. Step 3 and Step 4 in CPA are to update dual variables which will be used in Step 1 in the next iteration. From the duality theory [3], it can be seen that both the primary variables and the dual variables tend to be optimal only when the utility functions are concave. However, there are some applications of NUM in which the utility functions are not concave such as multipath routing [37], [38] and lossy networks [39], [40]. In these cases, we can use the extended Lagrange duality [41] to extend CPA to non-concave utility functions.

IV. MIXED METHOD

In this section, we develop a completely distributed algorithm for solving the problem (7), the algorithm will combine the first-order subgradient method [3], [24] and a local consensus algorithm [23]. For notation simplicity, here we let $\boldsymbol{y}_r(t)$ denote the estimate of user r for the resource allocation vector $\boldsymbol{x}(t)$ at time t and assume that each user r has an initial estimate $\boldsymbol{y}_r(0)$. Then, if we ignore the equation constraints in the problem (7), we have

$$\begin{array}{ll} \max_{\{\boldsymbol{y}_r, \boldsymbol{x}\}} & \sum_{r \in \mathcal{R}} U_r(\boldsymbol{y}_r) \\ \text{subject to} & \\ & A\boldsymbol{x} \le C \\ & \boldsymbol{x} \ge 0. \end{array} \tag{10}$$

Therefore, the first-order subgradient algorithm for updating y_r in the problem (10) can be written as follows,

$$\boldsymbol{y}_r(t+1) = \boldsymbol{y}_r(t) + \alpha(t) \partial U_r(\boldsymbol{y}_r(t)), \tag{11}$$

where $\alpha(t) > 0$ is a stepsize and $\partial U_r(\boldsymbol{y}_r(t))$ is a subgradient of the function $U_r(\boldsymbol{y}_r)$ at time t. We will consider the inequality constraint in the problem (10) later.

Apparently, algorithm (11) cannot make all network users reach a common value for resource allocation because each user independently updates the common resource vector and ignores the discarded constraint. This motivated us to combine algorithm (11) with a neighbour consensus algorithm which involves each user maintaining its estimate $\boldsymbol{y}_r(t)$ and updating it based on local information from its neighbour users (including itself). Thus, user r can update its estimates as follows,

$$\boldsymbol{y}_r(t+1) = \sum_{j \in n(r)} m_{rj} \boldsymbol{y}_j(t) + \alpha(t) \partial U_r(\boldsymbol{y}_r(t)), \forall r \in \mathcal{R}, \tag{12}$$

where $n(r)$ denotes the neighbours of user r and $m_{rj}, j \in n(r)$ are nonnegative weights that user r gives to its neighbour estimates $\boldsymbol{y}_j(t), j \in n(r)$.

Next, we consider the inequality constraint in problem (10). We can use the idea of penalty functions to turn the constrained problem (10) into an unconstrained problem as

$$\max_{\{\boldsymbol{y}_r \geq 0, \boldsymbol{x} \geq 0\}} \quad \sum_{r \in \mathcal{R}} U_r(\boldsymbol{y}_r) + \eta^T(C - A\boldsymbol{x}), \tag{13}$$

where $\eta > 0$ is the penalty factor. The objective function can be rewritten as

$$\begin{aligned} &\sum_{r \in \mathcal{R}} U_r(\boldsymbol{y}_r) + \eta^T(C - A\boldsymbol{x}) \\ &= \sum_{r \in \mathcal{R}} (U_r(\boldsymbol{y}_r) - x_r \sum_{l \in L} A_{lr}\eta_l) + \eta^T C \\ &= \sum_{r \in \mathcal{R}} (U_r(\boldsymbol{y}_r) - y_{rr} \sum_{l \in L} A_{lr}\eta_l) + \eta^T C. \end{aligned} \tag{14}$$

Here, the first equality follows due to the definition of the matrix **A** and the second one follows from the model (7). From the above equation (14), we know that its subgradient over $\boldsymbol{y}_r$ is

$$\bar{U_r} = \partial U_r(\boldsymbol{y}_r) - (0, \cdots, \sum_{l \in L} A_{lr}\eta_l, \cdots, 0).$$

Therefore, we can revise the algorithm (12) to obtain our mixed algorithm as follows,

$$\boldsymbol{y}_r(t+1) = \sum_{j \in n(r)} m_{rj}\boldsymbol{y}_j(t) + \alpha(t)\bar{U_r}, \forall r \in \mathcal{R}. \tag{15}$$

Remark 3. In our proposed algorithm (15), each user can weight equally the information obtained from its neighbour, i.e., $\forall r, m_{rj} = 1/|n(r)|, j \in n(r)$, here $|n(r)|$ denotes the number of elements of the set $n(r)$. Each user r can also allocate different weights $m_{rj}, j \in n(r)$ according to the neighbour numbers of user $j, j \in n(r)$, i.e., $\forall r, m_{rj} = \frac{|n(j)|}{\sum_{j \in n(r)}(|n(j)|)}, j \in n(r)$. Another approach for deciding the weights is to use the ratio of the allocated resource between neighbours, this means, $\forall r, m_{rj} = \frac{y_{rj}}{\sum_{j \in n(r)}(y_{rj})}, j \in n(r)$.

Remark 4. The algorithm (15) consists of two items. The first item is the consensus part, the second item is the subgradient part. The consensus part serves to drive all users to agree on a common value about the global resource allocation, while the subgradient part is taken by each network user to maximize its utility function. As specified in [23], when the network is connected to ensure the proper mixing of the network users' estimates, one would expect that all users have the same estimate after some time in algorithm (12), i.e., all network users cooperatively solve the problem (7) and obtain the globally optimal resource allocation by using local information.

Remark 5. The proposed algorithm (15) also requires some information exchanges, this is similar to the CPA. However, the information exchange is constrained to the neighbour users.

Remark 6. The algorithm (15) is a primary algorithm [42]. There are no updates about dual variables, this is an important difference with the CPA.

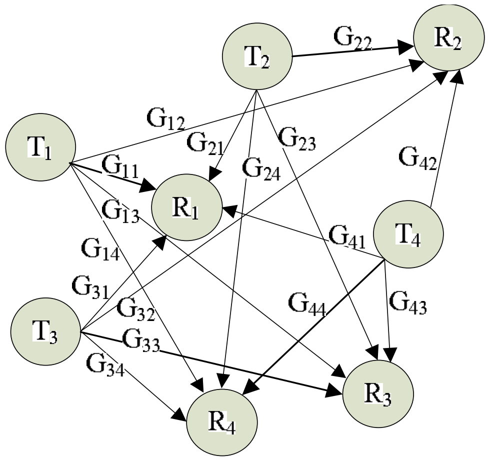

Fig. 1. Network topology.

The following theorem provides the convergence analysis of the algorithm (15).

Theorem 1 Suppose that the utility function is concave and the gradient of utility function is $\beta-$Lipschitz continuity, then the limit of the sequence $\{\boldsymbol{y}_r(t), r \in \mathcal{R}\}$ produced by the algorithm (15) is an optimal solution to the primary the problem (7).
Proof: [Proof] See Appendix A. □

V. POWER CONTROL FOR WSNS

Some WSNs do not rely on a pre-existing infrastructure and sensor nodes are randomly placed in an area. Since every sensor cannot know the global information of the whole network, if sensor nodes only work in a selfish way, this may lead to more serious co-channel interference and significantly deteriorate the performance of WSNs. To mitigate interference in WSNs, power control is one of the most-widely used basic techniques [43]. In this section, we apply the proposed method and algorithms to the power control problem for WSNs. Specially, we justify empirically the effectiveness of the proposed algorithm for power control, and compare the performance with the consistency price algorithm and the alternating direction method of multipliers(ADMM) [44] for power allocation problem in a randomly generated WSNs. We consider an example of a wireless sensor network consisting of four users (pairs of sensor nodes) as depicted in Fig. 1.

In Fig. 1, T_i and R_i denote the transmitter and receiver of user i, respectively, and G_{sj} denote the power gain from the user s to the user j.

We assume that the transmission from user s is interfered by other users' concurrent transmitters, and $n(T_1) = \{T_1, T_2, T_3\}$, $n(T_2) = \{T_1, T_2, T_4\}$, $n(T_3) = \{T_1, T_3\}$, $n(T_4) = \{T_2, T_4\}$, respectively. We also assume that each user weights equally the information obtained from its neighbor, i.e. $\forall s, m_{sj} = 1/|n(s)|, j \in n(s)$.

Let p_s denote the transmission power of node T_s with $P_s^{\max}$ being its maximum power constraint. The received signal-to-interference-plus-noise ratio (SINR) for user s's transmission is

$$\gamma_s(\boldsymbol{p}) = \frac{G_{ss}p_s}{\sum\limits_{k\neq s} G_{ks}p_k + \sigma_s}, \tag{16}$$

where σ_s is the receiving noise at user s and $\boldsymbol{p} = (p_1, p_2, p_3, p_4)$ is a vector of the transmission powers of nodes T_1, T_2, T_3 and T_4. Assume that $\sigma_s = 10^{-4}, p_s^{\max} = 1, \forall s \in \{1,2,3,4\}$. The power gains G_{lk} are equal to d_{lk}^{-4}, where d_{lk} represents the distance between the transmitter of link l and the receiver of link k. We consider a randomly generated realization of channel gains given by the matrix G of power gain is given as

$$G = \begin{pmatrix} 0.9610 & 0.0108 & 0.0123 & 0.0163 \\ 0.4823 & 0.6830 & 0.0141 & 0.0141 \\ 0.0767 & 0.0084 & 0.0219 & 0.0767 \\ 0.0625 & 0.0625 & 0.1197 & 0.0514 \end{pmatrix}.$$

We consider the logarithmic utility function, i.e. for each user s, $U_s(\gamma_s(\boldsymbol{p})) = \log(\gamma_s(\boldsymbol{p}))$. Then, we can establish the global power allocation problem as follows

$$\begin{aligned} &\max\nolimits_{\boldsymbol{p}} \textstyle\sum_{s=1}^{4} U_s(\gamma_s(\boldsymbol{p})) \\ &\qquad = \log(\gamma_1(\boldsymbol{p})) + \log(\gamma_2(\boldsymbol{p})) \\ &\qquad + \log(\gamma_3(\boldsymbol{p})) + \log(\gamma_4(\boldsymbol{p})) \\ &\text{subject to} \quad 0 \le \boldsymbol{p} \le \boldsymbol{p}^{\max} \end{aligned} \tag{17}$$

where $\boldsymbol{p}^{\max} = (p_1^{\max}, p_2^{\max}, p_3^{\max}, P_p^{\max})^T$, $p_s^{\max}, \forall s \in \{1,2,3,4\}$ denotes the maximum power of user s.

As specified above, we first transform the problem (17) into an equivalent NUM problem with coupled constraints by introducing slack variables as follows,

$$\begin{aligned} &\max\nolimits_{\boldsymbol{p},\boldsymbol{y}_s} \textstyle\sum_{s=1}^{4} U_s(\gamma_s(\boldsymbol{y}_s)) \\ &\qquad = \log(\gamma_1(\boldsymbol{y}_1)) + \log(\gamma_2(\boldsymbol{y}_2)) \\ &\qquad + \log(\gamma_3(\boldsymbol{y}_3)) + \log(\gamma_4(\boldsymbol{y}_4)) \\ &\text{subject to} \\ &\qquad \boldsymbol{y}_s = \boldsymbol{p}, \forall s \in \{1,2,3,4\} \end{aligned} \tag{18}$$

A. Covergence Evaluation

Next, we use our proposed algorithm to solve the equivalence problem (18). We first evaluate the convergence of the proposed algorithm with constant stepsize $\alpha = 0.01$, $\alpha = 0.005$ and a varying stepsize way for solving the problem (18). Let $U(t)$ and $U^*(t)$ denote the total utility at iteration number t and the optimal utility, respectively. Figs. 2–4 show the estimation error or corresponding primal function residual value $|U(t) - U^*(t)|$ in our proposed algorithm with $\alpha = 0.01$, $\alpha = 0.005$ and $\alpha = 1/t$, here t denotes the iteration number.

From Fig. 2 , we can see that our proposed primary algorithm is able to approach the optimal utility very fast with an estimation error less than 0.01. However, there are some oscillations while the iterations is close to the neighbour of the primal optimum. As we decrease the stepsize, we can see that shows that the iteration curve is more smoother now from Fig. 3. Therefore, we can decrease the oscillation by decreasing the stepsize, however, there is a trade-off: A smaller stepsize will lead to a slow convergence speed.

We also study the impact of a varying stepsize on the convergence of the proposed algorithm. Fig. 4 depicts the iteration curve with a varying stepsize $\alpha = 1/t$. From Fig. 4, we can know that the oscillation along the iteration curve is more bigger when compared with the constant stepsize.

Next, we consider another network topology by changing the distances between all pairs of sensor nodes in Fig. 1 and generating another realization of channel gains given by the matrix G

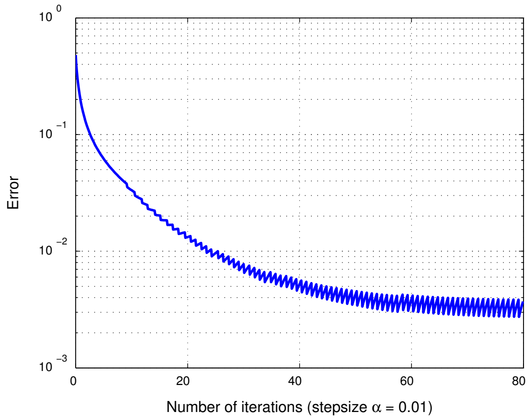

Fig. 2. The error of primary function values over optimal value versus iteration number t for the proposed algorithm with $\alpha = 0.01$.

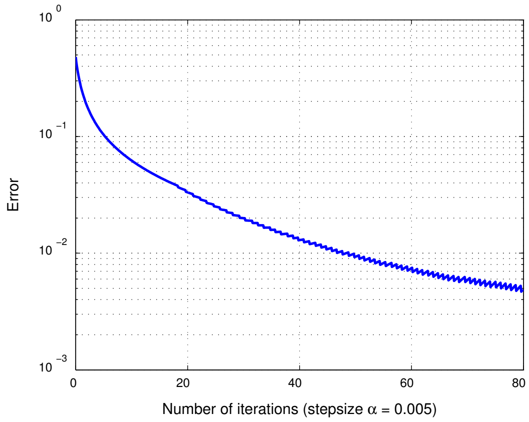

Fig. 3. The error of primary function values over optimal value versus iteration number t for the proposed algorithm with $\alpha = 0.005$.

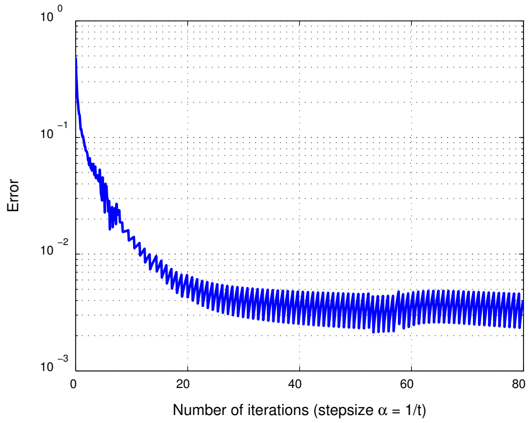

Fig. 4. The error of primary function values over optimal value versus iteration number t for the proposed algorithm $\alpha = 1/t$.

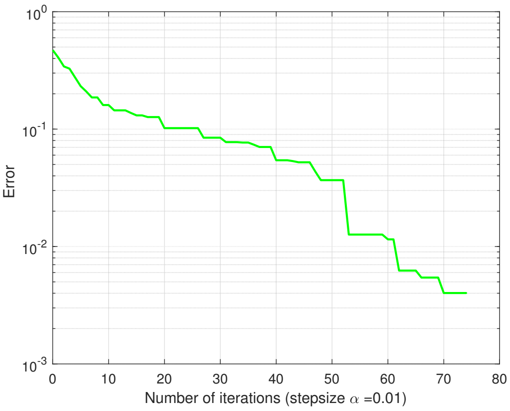

Fig. 5. The error of primary function values over optimal value versus iteration number t for the proposed algorithm with $\alpha = 0.01$ and $\xi = 9$.

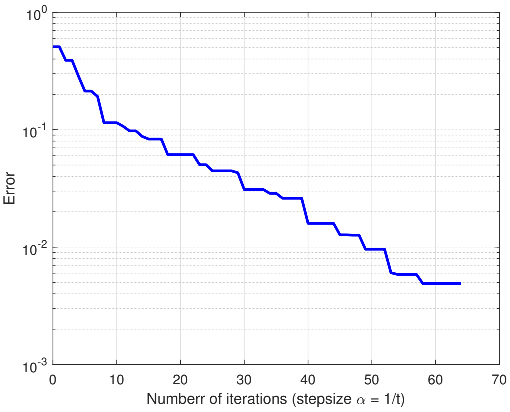

Fig. 6. The error of primary function values over optimal value versus iteration number t for the proposed algorithm $\alpha = 1/t$ and $\xi = 9$.

of power gain is given as

$$G = \begin{pmatrix} 0.8227 & 0.0323 & 0.0020 & 0.0017 \\ 0.0384 & 0.3501 & 0.0095 & 0.0028 \\ 0.0034 & 0.0309 & 0.2256 & 0.0211 \\ 0.0009 & 0.0032 & 0.0132 & 0.1197 \end{pmatrix}.$$

In order to verify the convergence of our proposed algorithm for different utility functions, we used another utility function defined as

$$U_s(\gamma_s) = \frac{1}{1-\xi}\gamma_s^{1-\xi}, \quad \xi > 1.$$

This is a widely used utility function in WSNs [45]. Here ξ is a constant, and the value of ξ is usually set to be greater than or equal to 8 [9]. In our next implementation, the parameter ξ was set to 9. Figs. 5 and 6 present the convergence behavior for our proposed algorithm with fixed stepsize and dimnishing stepsizes.

From Figs. 5 and 6, we can see that our our proposed primary algorithm is able to approach the optimal utility very fast with an estimation error less than 0.01 by using fixed stepsize and diminishing stepsizes. Since the diminishing stepsizes are bigger than the fixed stepsize (here it is 0.01) at the beginning of the iterations, therefore, the case with dimishing stepsize obtains a faster convergence rate compared to the case with fixed stepsize.

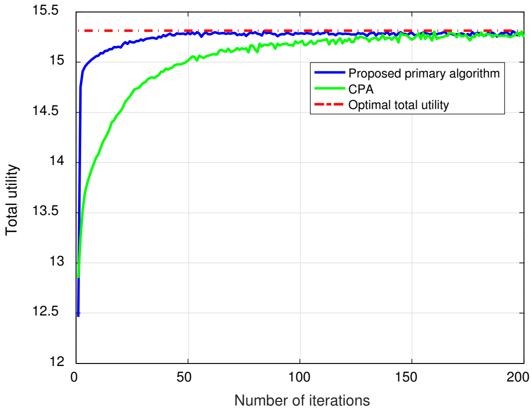

Fig. 7. The convergence comparison of the proposed algorithm and CPA with $\alpha = \beta = 0.01$

B. Comparisons of the Covergence Speed

CPA is a kind of primary-dual algorithms, which is often used to solve the utility maximization problem with coupled utility functions. Therefore, we use the CPA with $\alpha = 0.01, \beta = 0.01$ as a benchmark to evaluate the convergence speed for solving the problem (18). As shown in Fig. 7, our proposed primary algorithm and the CPA all approach the optimal total utility. However, the convergence rate of our proposed primary algorithm is faster than that obtained by the CPA. As CPA needs to update primary variables and dual variables, and let primary variables converge to the optimal values by decreasing the duality gap, this usually needs many iteration times. On the contrary, our proposed algorithm directly updates the primary variables, thus it obtains a faster convergence rate.

As described above, the stepsize has a great influence on the convergence speed of the algorithms. Fig. 8 presents the convergence speeds of the CPA and our proposed algorithm for solving the problem (18) with a different stepsize $\alpha = 0.005, \beta = 0.005$. Figs. 7 and 8 show that the CPA and our proposed algorithm reduce their convergence rates and obtain more smoother curves as the step length decreases. However, our proposed algorithm also achieves a faster convergence rate compared with the CPA.

As ADMM can quickly converge to the consensus problem. Next, we compare the convergence speed of the our proposed algorithm with ADMM for solving the problem (18). In the experiment, the stepsize α of our proposed algorithm and the penalty factor ρ of ADMM are set to 0.01 and 15, respectively.

From Fig. 9, we can see that our proposed primary algorithm and ADMM need about 50 and 100 iterations to approach the optimal value, respectively. The convergence speed of our proposed primary algorithm is faster than that obtained by ADMM for solving the problem (18). Moreover, as can be seen from Fig. 9, we also know that the iteration curve obtained by ADMM has a bigger oscillation than ours.

Next, we compare the convergence speed of our proposed algorithm with CPA and ADMM on another network scenario as

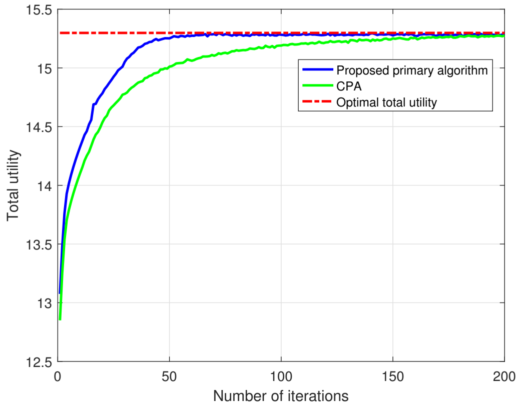

Fig. 8. The convergence comparison of the proposed algorithm and CPA with $\alpha = \beta = 0.005$

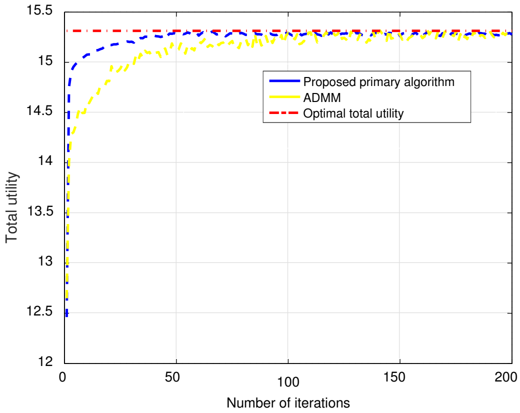

Fig. 9. The convergence comparison of the proposed algorithm and ADMM with $\alpha = 0.01$ and $\rho = 15$.

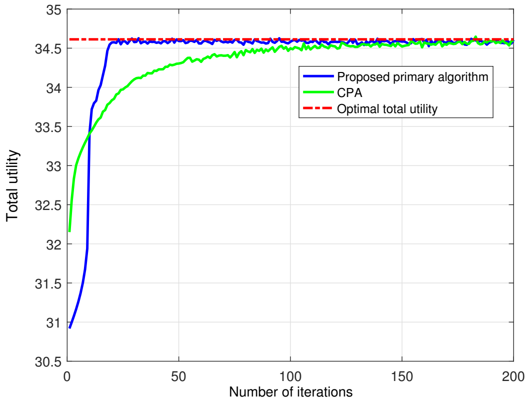

Fig. 10. The convergence comparison of the proposed algorithm and CPA with $\xi = 9$ and $\alpha = \beta = 0.01$

specified in the previous section. In Figs. 10 and 11, we provide the comparisons of convergence rates of these three algorithms.

From Figs. 10 and 11, we can see that the convergence speed of our proposed algorithm is faster compared with CPA and ADMM. These results are similar to those presented in Figs. 7 and 9. These validate the conclusion that our proposed algorithm obtains a faster convergence speed.

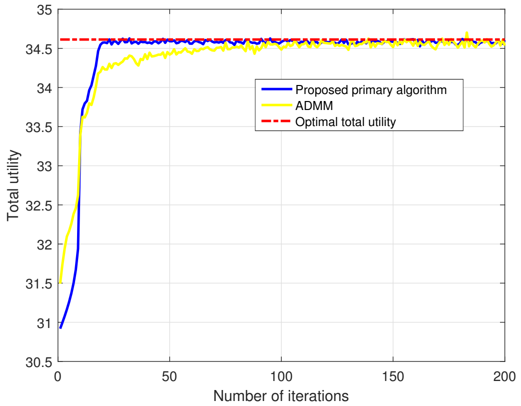

Fig. 11. The convergence comparison of the proposed algorithm and ADMM with $\xi = 9$, $\alpha = 0.01$ and $\rho = 15$.

VI. CONCLUSIONS

We propose a primary algorithm for solving globally coupled NUM problems based on local consensus algorithm and subgradient algorithm, and apply our proposed algorithm to power control in WSNs. Based on the simulation results, it can be seen that our proposed algorithm is effective in convergence speed compared with the classical primary-dual algorithm for the power control of WSNs.

The distributed NUM algorithms rely heavily on information exchanges between different network elements, such as traffic sources and routers. However, the feedback information is usually inaccurate because of time-varying networks and noise interference. Thus, another interesting direction for future research would be to examine the impact of feedback noises on our proposed method.

REFERENCES

[1] Kelly FP, Maulloo A, and Tan D, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J Oper Res Soc* , vol. 49, pp. 237–252, Mar. 1998.

[2] S. H. Low and D. E. Lapsley, "Optimal flow control, I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[3] Boyd, S and Vandenberghe, L *Convex optimization*. Cambridge University Press, Cambridge, U.K., 2004.

[4] P. Zhang, Y. Gang, X. Huang, S. Zeng and K. Xie, "Bandwidth allocation with utility maximization in the hybrid segment routing network." *IEEE Access*, vol. 7, pp. 85253–85261, June 2019.

[5] Y. Dong, S. Jin, S. Yang and H. H. F. Yin, "Network utility maximization for BATS code enabled multihop wireless networks," in *Proc. IEEE ICC*, 2020.

[6] Y. Im *et al.*, "AMUSE: Empowering users for cost-aware offloading with throughput-delay tradeoffs," *IEEE Trans. Mobile Comput.*, vol 15, no. 5, pp. 1065–1076, May 2016.

[7] M. Wang , H. Xu, and X. Zhou, "Cooperative dynamic game-based optimal power control in wireless sensor network powered by RF energy," *Sensors*, vol. 18, no. 7, pp. 1–10, July 2018.

[8] N. Maeryo *et al.*, "Fair bandwidth allocation algorithm for PONS based on network utility maximization," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 75–86, 2017.

[9] A. Verma and M. K. Hanawal, "Stochastic network utility maximization with unknown utilities: Multi-armed bandits approach," in *Proc. IEEE INFOCOM*, 2020.

[10] AA. Sinha and E. Modiano, “Network utility maximization with heterogeneous traffic flows,” in *Proc. IEEE WiOpt*, 2018.

[11] M. Chiang, P. Hande, T. Lan, and C. W. Tan, “Power control in wireless cellular networks,” *Foundations and Trends in Networking*, vol. 2, no. 4, pp. 381–533, June 2008.

[12] R. Cendrillon, Wei Yu, M. Moonen, J. Verlinden, and T. Bostoen, “Optimal multi-user spectrum management for digital subscriber lines,” *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 922–933, May 2006.

[13] L. Yang, Y. E. Sagduyu, J. Zhang, and J. H. Li, “Distributed stochastic power control in ad hoc networks: A nonconvex optimization case,” *EURASIP J. Wireless Commun. Netw.*, vol. 232, pp.1–14, July 2012.

[14] C. W. Tan, D. P. Palomar, and M. Chiang, “Distributed optimization of coupled systems with applications to network utility maximization” in *Proc. IEEE ICASSP*, 2006.

[15] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, “Distributed algorithms for transmission power control in wireless sensor networks,” in *IEEE WCNC*, 2003.

[16] M. Ahmed and S.-G. Yoon, “Dynamic access and power control scheme for interference mitigation in Femtocell networks, *KSII Trans. Internet and Inf. Syst.*, vol. 9, no. 11, pp. 4331–4346, Nov. 2015.

[17] S. D’Oro, A. Zappone, S. Palazzo, and M. Lops, “A learning approach for low-complexity optimization of energy efficiency in multi-carrier wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3226–3241, May 2018.

[18] J. Huang, R. A. Berry, and M. L. Honig, “Distributed interference compensation for wireless networks,”, *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp 1074–1084, May 2006.

[19] S. Boyd *et al.*, Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 2010; 3: 1-122.

[20] M. Chiangm “Network utility maximization: A survey on nonconvex and coupled formulations,” ACM Sigmetrics Hot Topic Session, 2005.

[21] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, “Accelerated dual descent for network flow optimization,” *IEEE Trans. Authomatic Control* , vol. 59, no. 4, pp. 905–920, Apr. 2014.

[22] E. Nekouei, T. Alpcan, G. N. Nair, and R. J. Evans, “Convergence analysis of quantized primal-dual algorithms in network utility maximization problems,” *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 284–297, Mar. 2018.

[23] A. Nedić and A. Ozdaglar, Cooperative distributed multi-agent optimization. *Convex optimization in signal processing and communications* , 2018, Cambridge University Press, pp. 340–386.

[24] D. Bertsekas, *Nolinear Programming*, Second ed., Athena scientific press, Athena, 1999.

[25] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.

[26] Q. Pham and W. Hwang, “Network utility maximization-based congestion control over wireless networks: A survey and potential directives,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1173–1200, Secondquarter 2017.

[27] M. Thai and P. Pardalos, *Handbook of Optimization in Complex Networks*. Springer-Verlag, New York, NY, USA, 2011.

[28] J. Sun and N. Liu, “Incentivizing verifiable privacy-protection mechanisms for offline crowdsensing applications,” *Sensors*, vol. 17, no. 9, pp. 1–25, Sept. 2017.

[29] Y. Zhao, S. Mao, J. O. Neel ,and J. H. Reed, “Performance evaluation of cognitive radios: Metrics, utility functions, and methodology,” *Proc. IEEE*, vol. 97, no. 4, pp. 642–659, Apr. 2009.

[30] R. Deng, Y. Zhang, S. He, J. Chen, and X. Shen, “Maximizing network utility of rechargeable sensor networks with spatiotemporally-coupled constraints,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1307–1319, May 2016.

[31] R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, “Autonomous spectrum balancing for digital subscriber lines,” *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4241–4257, Aug. 2007.

[32] S Lin *et al.*, “ATPC: Adaptive transmission power control for wireless sensor networks,” *ACM Trans. Sensor Netw.*, vol 12, no. 1, pp. 1–31, Mar. 2016.

[33] Y. Chen and M. Chen, “Extended duality for nonlinear programming,” *Comput. Optim. Appl.*, vol 47, pp. 33–59, Sept. 2008.

[34] S. Kirkpatrick, C. D. Gelatt, and J. M. P.Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[35] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer-Verlag, New York, NY, USA, 1985.

[36] S. H. Low, “A duality model of TCP and queue management algorithms,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[37] S. Li, W. Sun, and C. Hua, “Optimal resource allocation for heterogeneous traffic in multipath networks,” *Int. J. Commun. Syst.*, vol. 29, no. 1, pp. 84–98, Jan. 2016.

[38] Q.-V. Pham and W.-J. Hwang, “Network utility maximization in multipath lossy wireless networks,” *Int. J. Commun. Syst.*, vol 30, no. 5, pp. 1–18, Dec. 2015.

[39] Q.-V. Pham and W.-J. Hwang, “A multi-timescale cross-layer approach for wireless ad hoc networks,” *Computer Networks*, vol. 91, no. 14, pp. 471–482, Nov. 2015.

[40] Q. Gao, J. Zhang, and S. V. Hanly, “Cross-layer rate control in wireless networks with lossy links: Leaky-pipe flow, effective network utility maximization and hop-by-hop algorithms,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3068–3076, June 2009.

[41] Y. Chen Y and M. Chen, Extended duality for nonlinear programming. *Springer Comput. Optim. Appl.* 2010; 47: 33–59.

[42] R. Srikant, *The Mathematics of Internet Congestion Control*. Kluwer, Norwell, MA, USA, 1999.

[43] L. Zhang, H. Xu, F. Zhuo, and H. Duan, “Achieving congestion mitigation using distributed power control for spectrum sensor nodes in sensor network-aided cognitive radio Ad Hoc networks,” *Sensors*, vol. 17, no. 9, Sept. 2017.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations & Trends in Machine Learning* , vol. 3, no. 1, pp. 1–122, 2010.

[45] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen, “Utility-based asynchronous flow control algorithm for wireless sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1116–1126, Sept. 2010.

[46] J. Noceda and S. J. Wright, *Numerical optimization*, Springer-Verlag, New York, 1999.

APPENDIX A

Proof: [Proof of Theorem 1] We need the following Lemma 1 to finish our proof. The proof of Lemma 1 is simple, please refer to [46]

Lemma 1. Assume that function $f(x)$ is convex and its gradient is $\beta-$Lipschitz continuity, this means,

$$|| \bigtriangledown f(x) - \bigtriangledown f(y)||^2 \leq \beta ||x-y||^2. \tag{19}$$

Then we have,

$$f(x)-f(y) \leq \bigtriangledown f(x)^T(x-y) - \frac{1}{2\beta}||\bigtriangledown f(x) - \bigtriangledown f(y)||^2. \tag{20}$$

Assume that y_r^* is the optimal solution of the primary the problem (7). Base on the algorithm (15), we have

$$\begin{aligned}
&||y_r(t+1) - y_r^*||^2 \\
&= || \sum_{j\in n(r)} m_{rj} y_j(t) + \alpha(t)\bar{U}_r - y_r^*||^2 \\
&= || \sum_{j\in n(r)} m_{rj} y_j(t) + \alpha(t)\bar{U}_r - \sum_{j\in n(r)} m_{rj} y_r^*||^2 \\
&= || \sum_{j\in n(r)} m_{rj}(y_j(t) - y_r^*) + \alpha(t)\bar{U}_r||^2,
\end{aligned} \tag{21}$$

where we use the fact that $\sum_{j\in n(r)} m_{rj} = 1$ in the second step. Next, for the simplicity of the notation, we assume that all coefficients $m_{rj}, j \in n(r)$ are equal, and let $m_{rj} = \mu, j \in n(r)$, $\alpha(t) = \alpha$. Then, we can rewrite the equation (21) as follows

$$\begin{aligned}&||\, y_r(t+1)-y_r^*||^2\\ =&\mu^2||y_r(t)-y_r^*||^2+2\mu\alpha\bar{U}_r(y_r(t)-y_r^*)\\ &+2\mu\alpha\sum_{j\in n(r)\backslash r}\bar{U}_r(y_j(t)-y_r^*)\\ &+\sum_{j\in n(r)\backslash r}\mu^2||(y_j(t)-y_r^*)||^2+\alpha^2||\bar{U}_r||^2.\end{aligned}\tag{22}$$

By assumption that U_r is concave and $\bar{U}_r$ is $\beta-$Lipschitz continuity, combining with Lemma 1, we have

$$\begin{aligned}&-U_r(y_r(t))-(-U_r(y_r^*))\\ &\le-\bar{U}_r(y_r(t)-y_r^*)-\frac{1}{2\beta}||\bar{U}_r(y_r(t))-\bar{U}_r(y_r^*)||^2.\end{aligned}\tag{23}$$

As $U_r(y_r^*)\ge U_r(y_r(t))$ and $\bar{U}_r(y_r^*)=0$, from equation (23), we have

$$\bar{U}_r(y_r(t)-y_r^*)\le-\frac{1}{2\beta}||\bar{U}_r(y_r(t))||^2\tag{24}$$

Combining equations (22) and (24), it then yields that

$$\begin{aligned}&||\, y_r(t+1)-y_r^*||^2\\ \le&\mu^2||y_r(t)-y_r^*||^2+(\alpha^2-\frac{\mu\alpha}{\beta})||\bar{U}_r||^2\\ &+2\mu\alpha\sum_{j\in n(r)\backslash r}\bar{U}_r(y_j(t)-y_r^*)+\sum_{j\in n(r)\backslash r}\mu^2||(y_j(t)-y_r^*)||^2.\end{aligned}\tag{25}$$

Next, using this fact that $y_i^*=y_j^*,\forall i,j\in R$, the equation (25) can be rewritten as

$$\begin{aligned}&||\, y_r(t+1)-y_r^*||^2\\ \le&\mu^2||y_r(t)-y_r^*||^2+(\alpha^2-\frac{\mu\alpha}{\beta})||\bar{U}_r||^2\\ &+2\mu\alpha\sum_{j\in n(r)\backslash r}\bar{U}_r(y_j(t)-y_j^*)+\sum_{j\in n(r)\backslash r}\mu^2||(y_j(t)-y_j^*)||^2.\end{aligned}\tag{26}$$

For the last two items in equation (26), using Lemma 1 again, we have

$$\begin{aligned}&||y_r(t+1)-y_r^*||^2\\ &\le\sum_{j\in n(r)}(\mu^2||y_j(t)-y_r^*||^2+(\alpha^2-\tfrac{\mu\alpha}{\beta})||\bar{U}_r||^2).\end{aligned}\tag{27}$$

If $\alpha^2-\frac{\mu\alpha}{\beta}\le 0$, we then have

$$\begin{aligned}&||y_r(t+1)-y_r^*||^2\\ &\le\sum_{j\in n(r)}\mu^2||y_j(t)-y_r^*||^2.\end{aligned}\tag{28}$$

By the summation of the above equation (28), it yields

$$\begin{aligned}&\sum_{r\in R}||y_r(t+1)-y_r^*||^2\\ &\le\sum_{r\in R}\mu^2||y_r(t)-y_r^*||^2.\end{aligned}\tag{29}$$

Therefore, if $\alpha^2-\frac{\mu\alpha}{\beta}\le 0$, it follows that $y_r(t)$ converges to y_r^* when t tends to infinity. □

Shengbin Liao received his Ph.D. degree from the Department of Electronics and Information Engineering, HuaZhong University of Science and Technology in 2008. He is currently an Associate Professor at the National Engineering Research Center for E-Learning, Collaborative & Innovative Center for Educational Technology, Central China Normal University. His current research interests include machine learning and big data, distributed optimization and control, and smart education based on deep learning.