

This document is downloaded from DR-NTU, Nanyang Technological University Library, Singapore.

Title	Asynchronous Distributed Optimization for Network Utility Maximization Problems with Event Triggered Communication
Author(s)	Meng, Xiangyu; Chen, Tongwen
Citation	Meng, X., & Chen, T. (2015). Asynchronous Distributed Optimization for Network Utility Maximization Problems with Event Triggered Communication. <i>International Journal of Sensors, Wireless Communications and Control</i> , 5(2), 95-105.
Date	2015
URL	http://hdl.handle.net/10220/38711
Rights	© 2015 Bentham Science Publishers. This is the author created version of a work that has been peer reviewed and accepted for publication by <i>International Journal of Sensors, Wireless Communications and Control</i> , Bentham Science Publishers. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [http://dx.doi.org/10.2174/2210327905666150608225904] .

Asynchronous Distributed Optimization for Network Utility Maximization Problems with Event Triggered Communication

Xiangyu Meng, and Tongwen Chen

Abstract—This paper is concerned with an event triggered distributed optimization algorithm for network utility maximization (NUM) problems. According to an event triggering logic, a source broadcasts its information to links when a local signal exceeds a state dependent threshold. A similar communication logic is executed by each link, where the link broadcasts its information to all sources that use the link. The algorithm is based on a sequential barrier method, which can be applied to optimization problems with constraints. The efficiency of the proposed scheme is verified via simulations. The simulation result shows that the proposed algorithm reduces the number of message exchanges while guaranteeing the converge to the optimal solution.

I. INTRODUCTION

Many problems in control engineering can be formulated as optimization problems [1], [2], such as data gathering problems, estimation and localization problems [3], and maximum life routing problems in sensor networks. Most of these problems require a distributed solution since it is impossible to centralize all information by communication alone in a large-scale network. Another significant requirement in a large-scale network is the need for distributed computation of the optimal solution relying on local measurements, as opposed to requiring a central computation system. Therefore, interest in distributed optimization problems has been rapidly growing in recent years [4], [5]. Common distributed algorithms that solve network optimization problems are distributed gradient-based algorithms which guarantee the converge to optimal points provided that the communication between subsystems is sufficiently frequent. Typically, distributed networks are deployed in resource-constrained environments, and sensors powered by batteries communicate over radio, which is the most energy-consuming function. Energy constraints dictate that distributed optimization problems are best approached in a holistic manner to achieve energy efficiency. As a consequence of these issues, event triggered communication has been used in distributed optimization algorithms to mitigate the above issues, and the resulting optimization problems become tractable [6]. Event triggered distributed algorithms were introduced in a scenario where multiple agents cooperated to control their individual states so as to achieve a common objective while

communicating with each other to exchange state information [7], [8]. Distributed event triggered optimization algorithms were also well suited for solving the optimal active power flow problem [9]. The result in [9] confirms that the approximated solution obtained by event triggered distributed algorithms can enjoy significantly less communication while keeping the same accuracy as solutions computed by traditional ones.

In this paper, we focus on the network utility maximization (NUM) problem, where each source generates a flow over shared links with limited capacities to achieve the maximization of total sources' utility. Various techniques have been used to solve NUM problems distributively, such as decomposition approaches [10], and the Newton method [11]. In the design of distributed algorithms for NUM problems, one important factor to control is communication traffic in the network. In [12], barrier and augmented Lagrangian methods have been proposed to solve NUM problems based on a continuous flow updating rule and a continuous event detection scheme. In contrast to [12], a discrete time framework is presented in this paper to solve the same problem. Here a barrier method is used to transform the constrained optimization problem into a sequence of unconstrained optimization problems. For each unconstrained optimization problem, the calculation of the optimal flow rate is distributed among sources. Thanks to the barrier method, overflow never happens over shared links even though each source updates its own flow rate independently. Each link monitors the available capacity, and send a reciprocal of this information to corresponding sources when it feels "necessary". Note that from a privacy point of view, the link does not send the flow rate information of each source, and its capacity. The transmission instants are determined by a communication logic condition. Similar logic conditions are also configured at sources, where the sources broadcast messages to the corresponding links. The proposed algorithm guarantees a sub-optimal solution with an arbitrary accuracy to the global optimal flow rate but much less communication frequency than traditional methods. Also note that the proposed algorithm is asynchronous in the sense that the step sizes for calculating the flow rates are different for distinct sources. In addition, an upper bound of the step size is also provided.

The main contributions of this paper are briefly summarized as follows:

- propose a discrete time distributed interior point method for optimization problems with constraints;

X. Meng is with the EXQUISITUS, Centre for E-City, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xmeng@ntu.edu.sg).

T. Chen is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: tchen@ualberta.ca).

- develop triggering rules to mediate the communication between sources and links;
- solve the network utility maximization problem with guaranteed accuracy.

Notation: The set \mathbb{R} is the set of real numbers, \mathbb{R}^n real n -vectors, $\mathbb{R}^{m \times n}$ real $m \times n$ matrices, \mathbb{R}_+ nonnegative real numbers, and \mathbb{R}_{++} positive real numbers. The symbol X^T denotes the transpose of matrix X . The norm $\|x\|_2$ stands for the Euclidean norm of vector x . We use ∇f represents the gradient of function f and $\nabla^2 f$ the hessian of function f . The generalized inequality $x \preceq y$ means componentwise inequality between vectors x and y .

II. MATHEMATICAL BACKGROUND

In this section we give a brief review of some basic concepts used in this paper, which are mostly borrowed from [4].

A. Derivatives

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $x \in \text{int dom } f$. The function f is differentiable at x if there exists a matrix $Df(x) \in \mathbb{R}^{m \times n}$ that satisfies

$$\lim_{z \in \text{dom } f, z \neq x, z \rightarrow x} \frac{\|f(z) - f(x) - Df(x)(z - x)\|_2}{\|z - x\|_2} = 0,$$

in which case $Df(x)$ is referred to as the *derivative* or *Jacobian* of f at x . The function f is differentiable if $\text{dom } f$ is open, and it is differentiable at every point in its domain. The derivative can be found from partial derivatives:

$$Df(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

When f is real-valued, the derivative $Df(x)$ is a $1 \times n$ matrix, *i.e.*, it is a *row* vector. Its transpose is called the gradient of the function:

$$\nabla f(x) = Df(x)^T,$$

which is a *column* vector in \mathbb{R}^n . Its components are the partial derivatives of f :

$$\nabla f(x)_i = \frac{\partial f(x)}{\partial x_i}, \quad i = 1, \dots, n.$$

As an example of chain rule, suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R} \rightarrow \mathbb{R}$, and $h(x) = g(f(x))$. Taking transpose of $Dh(x) = Dg(f(x))Df(x)$ yields

$$\nabla h(x) = g'(f(x))\nabla f(x).$$

A general chain rule for the second derivative is stated below:

$$\nabla^2 h(x) = g'(f(x))\nabla^2 f(x) + g''(f(x))\nabla f(x)\nabla f(x)^T.$$

B. Convex Functions

Definition 1: A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if $\text{dom } f$ is a convex set and if for all $x, y \in \text{dom } f$, and θ with $0 \leq \theta \leq 1$, it holds

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y). \quad (1)$$

A function f is strictly convex if strict inequality holds in (1) whenever $x \neq y$ and $0 < \theta < 1$. The function f is *concave* if $-f$ is convex, and *strictly concave* if $-f$ is strictly convex.

For example, $\log x$ is concave on \mathbb{R}_{++} .

Now introduce two operations that preserve convexity or concavity of functions.

- 1) A nonnegative weighted sum of convex functions,

$$f = w_1 f_1 + \dots + w_m f_m,$$

is convex. Similarly, a nonnegative weighted sum of concave functions is concave. A nonnegative, nonzero weighted sum of strictly convex (concave) functions is strictly convex (concave).

- 2) Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $A \in \mathbb{R}^{n \times m}$, and $b \in \mathbb{R}^n$. Define $g : \mathbb{R}^m \rightarrow \mathbb{R}$ by

$$g(x) = f(Ax + b),$$

with $\text{dom } g = \{x | Ax + b \in \text{dom } f\}$. Then if f is convex, so is g ; if f is concave, so is g .

Definition 2: A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be *closed* if, for each $\alpha \in \mathbb{R}$, the sublevel set

$$\{x \in \text{dom } f | f(x) \leq \alpha\}$$

is closed.

For example, the functions $f(x) = -\log x$ with $\text{dom } f = \mathbb{R}_{++}$ and $f(x) = -\sum_{i=1}^m \log(b_i - a_i^T x)$ with $\text{dom } f = \{x | a_i^T x < b_i, i = 1, \dots, m\}$ are closed.

The starting point must lie in $\text{dom } f$, and in addition the sublevel set

$$S = \left\{x \in \text{dom } f \mid f(x) \leq f(x^{(0)})\right\}$$

must be closed. Assume f is strongly convex on S , so there are positive constants m and M such that

$$mI \leq \nabla^2 f(x) \leq MI$$

for all $x \in S$. The lower and upper bounds on the Hessian imply for any $x, y \in S$,

$$\begin{aligned} f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}\|y - x\|_2^2 &\leq f(y) \\ &\leq f(x) + \nabla f(x)^T(y - x) + \frac{M}{2}\|y - x\|_2^2. \end{aligned}$$

III. PROBLEM FORMULATION

Consider a network in which two special nodes, called the *source* and the *destination*, are distinguished. The nodes of a network are usually numbered, say, $1, 2, 3, \dots, S$. A network is typically represented as shown in Figure 1. The nodes are designated by circles, with a symbol inside each circle denoting the type and index of that node. Denote $\mathcal{S} = \{1, \dots, S\}$ as the index set of sources in the network. Corresponding

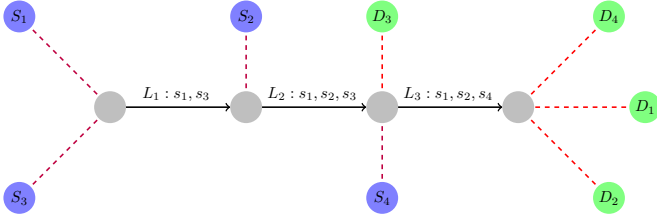


Fig. 1. A sample network with four sources and three links

to each source node $i \in \mathcal{S}$, there is a nonnegative number s_i representing a flow along a predetermined route to a destination. Use $\mathcal{L} = \{1, \dots, L\}$ to denote the index set of directed links. Associated with each link is a finite number $c_l > 0$ for $l \in \mathcal{L}$, representing the maximum allowable flow. The network utility maximization (NUM) problem is that of determining the maximal aggregate sources' utilities subject to links' capacity constraints. When written out, it takes the form

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^S U_i(s_i) \\ & \text{subject to} && R s \leq c, s \geq 0. \end{aligned} \quad (2)$$

Here $U_i : \mathbb{R}_+ \rightarrow \mathbb{R}$ is the utility of source i as a function of the source rate s_i . It is assumed to be twice differentiable, strictly concave in s_i , and strictly monotonically increasing on $(0, \infty)$. The vectors $s = [s_1, \dots, s_S]^T$ and $c = [c_1, \dots, c_L]^T$ are compact forms of the source data rates and link capacities, respectively. The matrix R is the routing matrix of dimension $L \times S$, i.e.,

$$r_{ij} = \begin{cases} 1, & \text{if link } i \text{ is on the route of source } j, \\ 0, & \text{otherwise.} \end{cases}$$

For each link l , let \mathcal{S}_l denote the set of sources that use link l ; for each source i , let \mathcal{L}_i denote the set of links that source i uses. Also assume that the problem is strictly feasible, i.e. the constraint set \mathcal{F} taking the form

$$\mathcal{F} = \{s : R s - c \leq 0, s \geq 0\}$$

has a nonempty interior that is arbitrarily close to any point in \mathcal{F} . Intuitively, this means that the set has an interior and it is possible to get to any boundary point by approaching it from the interior. Such a set is referred to as *robust* [13].

The barrier method is used to approximately formulate the inequality constrained problem in (2) as a sequence of unconstrained problems. The approximation is accomplished by adding terms to the objective function that favors points interior to the feasible region over those near the boundary. The first step is to rewrite the problem in (2), making the inequality constraints implicit in the objective function:

$$\text{minimize} \quad f(s; \lambda, \mu), \quad (3)$$

where

$$\begin{aligned} f(s; \lambda, \mu) = & - \sum_{i \in \mathcal{S}} U_i(s_i) - \sum_{j \in \mathcal{L}} \frac{1}{\lambda_j} \log(c_j - r_j^T s) \\ & - \sum_{i \in \mathcal{S}} \frac{1}{\mu_i} \log s_i \end{aligned}$$

is the *Lagrangian* associated with the problem (2) and r_1^T, \dots, r_L^T are the rows of R . Here $\lambda = [\lambda_1, \dots, \lambda_L]^T$ and

$\mu = [\mu_1, \dots, \mu_S]^T$ are constant vectors with positive entries that set the accuracy of the approximation.

Remark 1: The objective function in (3) is convex. This can be shown via the definition of convex function and operations that preserve convexity of functions.

The functions $\phi_{\mathcal{L}}(s) = -\sum_{j \in \mathcal{L}} \log(c_j - r_j^T s)$, and $\phi_{\mathcal{S}}(s) = -\sum_{i \in \mathcal{S}} \log s_i$ with $\text{dom} \phi_{\mathcal{L}} \cap \text{dom} \phi_{\mathcal{S}} = \{s \in \mathbb{R}^S \mid s_i > 0, r_j^T s < c_j, i = 1, \dots, S, j = 1, \dots, L\}$, are called the *logarithmic barriers* or *log barriers* for the problem in (2).

Remark 2: The above domain is the set of points that satisfy the inequality constraints in (2) strictly. No matter what values the positive parameters $\lambda_j, j = 1, \dots, L$ and $\mu_i, i = 1, \dots, S$ have, the logarithmic barriers grow without bound if $c_j - r_j^T s \rightarrow 0$ or $s_i \rightarrow 0$, for any i and j . Of course, the problem in (3) is only an approximation of the original problem in (2). The quality of the approximation improves as the parameters λ_j and μ_i grow. It can be soon shown that the original problem can be circumvented by solving a *sequence* of problems of the form in (3), increasing the parameters λ_j and μ_i , and starting each minimization at the solution to the problem with the previous values of λ_j and μ_i .

For $\lambda \geq 0$ and $\mu \geq 0$, define $s^*(\lambda, \mu)$ as the optimal solution to (3). The *central path* associated with the problem in (2) is defined as the set of points $s^*(\lambda, \mu), \lambda \geq 0$ and $\mu \geq 0$, which are called the *central points*. Points on the central path are characterized by the following necessary and sufficient conditions: $s^*(\lambda, \mu)$ is strictly feasible, i.e., it satisfies

$$r_j^T s < c_j, j = 1, \dots, L, \quad s_i > 0, i = 1, \dots, N,$$

and

$$0 = - \sum_{i \in \mathcal{S}} \nabla U_i(s_i) - \sum_{i \in \mathcal{S}} \frac{1}{\mu_i} \frac{1}{s_i} + \sum_{j \in \mathcal{L}} \frac{1}{\lambda_j} \frac{1}{c_j - r_j^T s} \quad (4)$$

holds.

Every central point yields a dual feasible point, and hence a lower bound on the optimal value p^* . More specifically, define

$$\begin{aligned} \nu_i^*(\mu_i) &= \frac{1}{\mu_i} \frac{1}{s_i}, i = 1, \dots, S, \\ v_j^*(\lambda_j) &= \frac{1}{\lambda_j} \frac{1}{c_j - r_j^T s}, j = 1, \dots, L. \end{aligned}$$

The pair $(\nu^*(\mu), v^*(\lambda))$ is dual feasible.

First, it is clear that $\nu^*(\mu) \succ 0$ and $v^*(\lambda) \succ 0$ because $s_i > 0, i = 1, \dots, S$, and $r_j^T s < c_j, j = 1, \dots, L$. By expressing the optimality condition in (4) as

$$- \sum_{i \in \mathcal{S}} \nabla U_i(s_i^*(\mu, \lambda)) - \sum_{i \in \mathcal{S}} \nu_i^*(\mu_i) + \sum_{j \in \mathcal{L}} v_j^*(\lambda_j) = 0,$$

it can be seen that $s^*(\mu, \lambda)$ minimizes the Lagrangian $f(s; \lambda, \mu)$ for $\nu = \nu^*(\mu)$ and $v = v^*(\lambda)$, which means that

$(\nu^*(\mu), v^*(\lambda))$ is a dual feasible pair. Therefore, the dual function $g(\nu^*(\mu), v^*(\lambda))$ is finite, and

$$\begin{aligned} g(\nu^*(\mu), v^*(\lambda)) &= - \sum_{i \in \mathcal{S}} U_i(s_i^*) - \sum_{i \in \mathcal{S}} \nu_i^*(\mu) s_i \\ &\quad + \sum_{j \in \mathcal{L}} v_j^*(\lambda) (r_j^T s - c_j) \\ &= - \sum_{i \in \mathcal{S}} U_i(s_i^*) - \sum_{i \in \mathcal{S}} \frac{1}{\mu_i} - \sum_{j \in \mathcal{L}} \frac{1}{\lambda_j}. \end{aligned}$$

In particular, the duality gap associated with $s^*(\mu, \lambda)$ and the dual feasible pair $(\nu^*(\mu), v^*(\lambda))$ is simply $-\sum_{i \in \mathcal{S}} \mu_i^{-1} - \sum_{j \in \mathcal{L}} \lambda_j^{-1}$. As an important consequence, it follows that

$$- \sum_{i \in \mathcal{S}} U_i(s_i^*) - p^* \leq \sum_{i \in \mathcal{S}} \frac{1}{\mu_i} + \sum_{j \in \mathcal{L}} \frac{1}{\lambda_j},$$

i.e., $s^*(\lambda, \mu)$ is no more than $\sum_{i \in \mathcal{S}} \mu_i^{-1} + \sum_{j \in \mathcal{L}} \lambda_j^{-1}$ -suboptimal. This confirms the intuitive idea that $s^*(\lambda, \mu)$ converges to an optimal point as $\lambda_i \rightarrow \infty$ and $\mu_j \rightarrow \infty$.

Remark 3: Barrier methods are also referred to as interior point methods. They work by establishing a barrier on the boundary of the feasible region that prevents a search procedure from leaving the region. This suggests that the link capacity constraints can always be guaranteed. Therefore, this method is reliable enough to be embedded in real-time control applications with little or no human oversight.

Define the j th link's local state as

$$\alpha_j(k) = \frac{1}{\lambda_j} \frac{1}{c_j - a_j^T s(k)}.$$

Link j is able to measure the total flow that goes through it and calculate its link state. The benefit of introducing event triggered techniques lies in reduced computation cost and communication cost at each link. Links do not need to calculate the state information and send it to sources at each time instant.

The sampling strategy is push type, which means that sources do not request information from links; links send their information to sources when it is necessary. The same strategy applies for sources.

IV. CONVERGENCE ANALYSIS FOR EVENT TRIGGERED BARRIER ALGORITHMS

Here we introduce two types of communication protocols: broadcasting and point-to-point communication.

A. Broadcasting Communication Protocol

Let $T_n^{L_j}$, $n = 1, 2, \dots$, denote the time when link j samples its link state α_j and broadcasts it to sources $i \in \mathcal{S}_j$. Therefore, the sampled link state is a piecewise constant function in which

$$\hat{\alpha}_j(k) = \alpha_j(T_n^{L_j})$$

for any $k \in [T_n^{L_j}, T_{n+1}^{L_j})$. Define

$$z_i(k) = \nabla U_i(s_i(k)) + \frac{1}{\mu_i} \frac{1}{s_i(k)} - \sum_{j \in \mathcal{L}_i} \hat{\alpha}_j(k)$$

as the i th source state for all $i \in \mathcal{S}$. Let $T_n^{S_i}$, $n = 1, 2, \dots$, denote the time when source i samples its source state $z_i(k)$

and broadcasts it to links $j \in \mathcal{L}_i$. Therefore, the sampled source state is also a piecewise constant function in which

$$\hat{z}_i(k) = z_i(T_n^{S_i})$$

for any $k \in [T_n^{S_i}, T_{n+1}^{S_i})$. Let \mathcal{E}_{L_j} and \mathcal{E}_{S_i} be the set of indices in $\{T_n^{L_j}\}$ and $\{T_n^{S_i}\}$ corresponding to events at link j and source i , respectively. The communication event instants at source i and link j are determined by the violation of the following inequalities:

$$z_i^2(k) \geq \rho_i \hat{z}_i^2(k), \quad (5)$$

and

$$(\alpha_j(k) - \hat{\alpha}_j(k))^2 \leq \frac{\sum_{i \in \mathcal{S}_j} \frac{\rho_i \gamma_i}{|\mathcal{L}_i|} (1 - \frac{\varepsilon_i}{2} - \frac{M}{2} \gamma_i) \hat{z}_i^2}{\sum_{i \in \mathcal{S}_j} \frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i}}, \quad (6)$$

respectively.

The minimizer of the Lagrangian $L(s; \lambda, \mu)$ for fixed λ and μ can be searched using the basic descent method. Given an initial feasible vector $s(0)$, the iteration of event triggered optimization algorithms for source i is given by

$$s_i(k+1) = s_i(k) + \gamma_i z_i(k), \quad (7)$$

where the positive scalar γ_i is called the *step length*. In other words, communication is invoked only when the event conditions are not satisfied. When a communication event is triggered, \hat{z}_i or $\hat{\alpha}_j$ is set to z_i or α_j . Therefore, the event conditions are satisfied instantaneously.

Theorem 1: Under the communication logic in (5) and (6), and the flow update scheme in (7), if the step length is set by

$$0 < \gamma_i < \frac{2 - \varepsilon_i}{M}$$

with $0 < \varepsilon_i < 2$, then the source rates $s(k)$ asymptotically converge to the unique minimizer of $f(s; \lambda, \mu)$.

Proof. Given a suitable starting point $s^0 \in \mathbf{dom} f$, the sublevel set

$$A = \{s \in \mathbf{dom} f | f(s) \leq f(s^0)\}$$

is closed since the function f is closed. Since the objective function is strongly convex on A , which means that there exists an $m > 0$ such that

$$\nabla^2 f(s) \geq mI \quad (8)$$

for all $s \in A$, and the inequality

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|y - x\|_2^2 \quad (9)$$

holds for all x and y in A . The inequality in (9) implies that the sublevel set contained in A is bounded, so in particular, A is bounded. Therefore the maximum eigenvalue of $\nabla^2 f(s)$, which is a continuous function of s on A , is bounded above on A , i.e., there exists a constant such that

$$\nabla^2 f(x) \leq MI$$

for all $s \in A$. This upper bound on the Hessian implies for $s(k+1) \in A$ and $s(k) \in A$,

$$\begin{aligned} f(s(k+1)) &\leq f(s(k)) + \nabla f(s(k))^T (\gamma \cdot z(k)) \\ &\quad + \frac{M}{2} \|\gamma \cdot z(k)\|_2^2, \end{aligned}$$

where \cdot denotes the element-wise multiplication of two vectors.

From the definition of $z_i(k)$, $\alpha_j(k)$, and $\hat{\alpha}_j(k)$, we have

$$\nabla_i f(s(k); \lambda, \mu) = -z_i(k) + \sum_{j \in \mathcal{L}_i} [\alpha_j(k) - \hat{\alpha}_j(k)].$$

For convenience, parameter dependence is omitted temporarily to save writing.

Define $V(s(k)) = f(s(k); \lambda, \mu) - f(s^*; \lambda, \mu)$ as a Lyapunov function candidate for the system in (7), where $s^*(\lambda, \mu)$ is the minimizer for any fixed λ and μ , and the corresponding Lagrangian function is $f(s^*; \lambda, \mu)$. By using the properties of $U_i(s_i)$, it is easy to show that such a minimizer is unique. It is trivial to see that $\Delta V(s) = \Delta f(s; \lambda, \mu)$.

For all $k \geq 0$, we have

$$\Delta V \leq \sum_{i=1}^S \left\{ \gamma_i z_i \left[\sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j) - z_i \right] + \frac{M}{2} \gamma_i^2 z_i^2 \right\}.$$

Using Young's inequality

$$xy \leq \frac{x^2}{2\varepsilon} + \frac{\varepsilon y^2}{2},$$

we get

$$\Delta V \leq \sum_{i=1}^S \left\{ -\gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} \gamma_i \right) z_i^2 \right\} + \sum_{i=1}^S \left\{ \frac{\gamma_i}{2\varepsilon_i} \left[\sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j) \right]^2 \right\}.$$

Remember there are $|\mathcal{L}_i|$ terms in the sum $\sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j)$ and then by using the sum of squares inequality

$$\left[\sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j) \right]^2 \leq |\mathcal{L}_i| \sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j)^2,$$

we have

$$\Delta V \leq \sum_{i=1}^S \left[-\gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2 \right] + \sum_{i=1}^S \left[\frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i} \sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j)^2 \right].$$

We can write the last term as

$$\sum_{i=1}^S \frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i} \sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j)^2 = \sum_{j=1}^L (\alpha_j - \hat{\alpha}_j)^2 \sum_{i \in S_j} \frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i}.$$

This means

$$\Delta V \leq \sum_{i=1}^S -\gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2 + \sum_{j=1}^L (\alpha_j - \hat{\alpha}_j)^2 \sum_{i \in S_j} \frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i}.$$

Considering the term $\sum_{i=1}^S \rho_i \gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2$, we have

$$\sum_{i=1}^S \rho_i \gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2 = \sum_{j=1}^L \sum_{i \in S_j} \frac{\rho_i \gamma_i}{|\mathcal{L}_i|} \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2.$$

Adding and subtracting the term

$$\sum_{i=1}^S \rho_i \gamma_i \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) z_i^2,$$

we obtain

$$\Delta V \leq - \sum_{i=1}^S \gamma_i \left(1 - \frac{M}{2} r_i - \frac{\varepsilon_i}{2} \right) (z_i^2 - \rho_i \hat{z}_i^2) + \sum_{j=1}^L \left[(\alpha_j - \hat{\alpha}_j)^2 \sum_{i \in S_j} \frac{\gamma_i |\mathcal{L}_i|}{2\varepsilon_i} \right] - \sum_{j=1}^L \left[\sum_{i \in S_j} \frac{\rho_i \gamma_i}{|\mathcal{L}_i|} \left(1 - \frac{\varepsilon_i}{2} - \frac{M}{2} r_i \right) \hat{z}_i^2 \right].$$

This immediately suggests that $\Delta V(s) \leq 0$ is guaranteed for all k because the inequalities in (5) and (6) hold for any $i \in \mathcal{S}$ and $j \in \mathcal{L}$.

The only scenario that $\Delta V = 0$ can happen is

$$z_i = \hat{z}_i = 0, \quad \forall i \in \mathcal{S}, \quad \alpha_j = \hat{\alpha}_j = 0, \quad \forall j \in \mathcal{L},$$

which corresponds to $s^*(\mu, \lambda)$. As a result, the equilibrium $s^*(\mu, \lambda)$ is asymptotically stable. ■

B. Point-to-Point Communication Protocol

For this type of communication protocol, the communication between links and sources is point to point. Let $T_n^{L_j 2S_i}$, $n = 1, 2, \dots$, denote the time when the link j samples its link state α_j and transmits that state to source i . Therefore, the sampled link state is held constant at the source i ' node until a new message arrives in which

$$\hat{\alpha}_j^i(k) = \alpha_j(T_n^{L_j 2S_i}) \quad (10)$$

for any $k \in [T_n^{L_j 2S_i}, T_{n+1}^{L_j 2S_i})$. Define

$$z_i(k) = \nabla U_i(s_i(k)) + \frac{1}{\mu_i} \frac{1}{s_i(k)} - \sum_{j \in \mathcal{L}_i} \hat{\alpha}_j^i(k). \quad (11)$$

Let $T_n^{S_i 2L_j}$, $n = 1, 2, \dots$, denote the time when the source i samples its source state $z_i(k)$ and sends that state to link j . Then, the link holds the information until the state is updated in which

$$\hat{z}_i^j(k) = z_i(k) \quad (12)$$

for any $k \in [T_n^{S_i 2L_j}, T_{n+1}^{S_i 2L_j})$. The communication protocol for the source i and link j are determined by

$$z_i(k) \hat{z}_i^j(k) > 0, \quad (13)$$

and

$$\hat{z}_i^j(k) (a_j(k) - \hat{\alpha}_j^i(k)) < 0 \quad (14)$$

for all $i \in S_j$. When $\hat{z}_i^j(k) = 0$, then source i and link j do not need to exchange any information until $z_i(k) \neq 0$.

Remark: The event condition satisfies the general properties. When an event occurs, the event condition is satisfied immediately.

The state update scheme employed by the i th source is of the general form

$$s_i(k+1) = s_i(k) + \gamma_i z_i(k) \quad (15)$$

where γ_i is a constant positive step size.

Theorem 2: Assume that U_i is twice differentiable, strictly increasing, and strictly concave, and the routing matrix R is of full rank. Assume a fixed barrier vector $\lambda \succeq 0$ and $\mu \succeq 0$

and assume the initial source rates $s_i(0)$, $i \in \mathcal{S}$ lie in the feasible set \mathcal{F} . Consider the source event condition in (13) for each source $i \in \mathcal{S}$ and link event condition in (14) for each link $j \in \mathcal{L}$. For each source $i \in \mathcal{S}$, let the source rate, $s_i(k)$, satisfy (15) with sampled link states given by (10). For each $i \in \mathcal{S}$, let the source state, $z_i(k)$, satisfy (11), and assume link j 's measurement of the source state satisfies (12). Then the source rates $s(t)$ asymptotically converge to the unique minimizer of $L(s; \lambda, \mu)$.

Proof. In analogy to the proof of Theorem 1, we have

$$\nabla_i L[s(k); \lambda, \mu] = -z_i(k) + \sum_{j \in \mathcal{L}_i} [\alpha_j(k) - \hat{\alpha}_j^i(k)].$$

For all $k \geq 0$, we have

$$\begin{aligned} \Delta V &\leq \sum_{i=1}^S \gamma_i z_i \left[\sum_{j \in \mathcal{L}_i} (\alpha_j - \hat{\alpha}_j^i) - z_i \right] + \frac{M}{2} \gamma_i^2 z_i^2 \\ &= - \sum_{i=1}^S \gamma_i \left(1 - \frac{M}{2} \gamma_i \right) z_i^2 \\ &\quad + \sum_{i=1}^S \sum_{j \in \mathcal{L}_i} \gamma_i z_i (\alpha_j - \hat{\alpha}_j^i). \end{aligned}$$

Without loss of generality, suppose that $\hat{z}_i^j > 0$, then $\alpha_j - \hat{\alpha}_j^i < 0$ according to the event condition in (14). From (13), we also know that $z_i > 0$. Therefore, we have

$$z_i (\alpha_j - \hat{\alpha}_j^i) < 0.$$

When $\hat{z}_i^j(k) = 0$, it implies that $z_i(k) = 0$. Therefore, we have

$$\Delta V \leq - \sum_{i=1}^S \gamma_i \left(1 - \frac{M}{2} \gamma_i \right) z_i^2.$$

This immediately suggests that the equilibrium $s^*(\lambda, \mu)$ is asymptotically stable. ■

Remark 4: The event conditions in (13) and (14) are completely distributed event detection. In contrast to the broadcasting type event conditions, there are no shared predetermined parameters between the sources and links. The sources and links could have no idea about the network.

The state update scheme employed by the i th source is of the general form

$$s_i(k+1) = s_i(k) - \gamma_i \nabla_i \hat{L}(s(k); \lambda, \mu) \quad (16)$$

where γ_i is a constant positive step size.

Remark 5: From a system point of view, the state update scheme

$$\begin{aligned} s_i(k+1) &= s_i(k) + \underbrace{\gamma_i \nabla U_i(s_i(k))}_{f(s_i(k))} + \frac{\gamma_i}{\mu_i} \frac{1}{s_i(k)} \\ &\quad - \underbrace{\gamma_i \sum_{j \in \mathcal{L}_i} \hat{\alpha}_j(k)}_{u_i(k)}, \end{aligned}$$

can be regarded as a nonlinear discrete time system, and $u_i(k)$ is a piece-wise constant control input.

V. EVENT TRIGGERED NUM ALGORITHM IMPLEMENTATION

The initial condition can be chosen distributively. Assume that each link knows the number of sources that follow through, that is, $|\mathcal{S}_j|$, and then it sends the information

$$\frac{c_j}{|\mathcal{S}_j|}$$

to each source. Each source receives $|\mathcal{L}_i|$ such information. Then the source can choose the initial flow rate

$$s_i^0 = \theta_i \min \left\{ \frac{c_j}{|\mathcal{S}_j|}, j \in \mathcal{L}_i \right\}, \quad (17)$$

with $0 < \theta_i < 1$.

Suppose that $k_i \in \mathcal{L}_i$, and $\frac{c_{k_i}}{|\mathcal{S}_{k_i}|} = \min \left\{ \frac{c_j}{|\mathcal{S}_j|}, j \in \mathcal{L}_i \right\}$. It is easy to show that the link constraint is satisfied automatically

$$\sum_{i \in \mathcal{S}_j} s_i^0 = \sum_{i \in \mathcal{S}_j} \theta_i \frac{c_{k_i}}{|\mathcal{S}_{k_i}|} < \sum_{i \in \mathcal{S}_j} \frac{c_j}{|\mathcal{S}_j|} = c_j.$$

Algorithm 1: Source i 's Update Algorithm

1) Source i 's initialization

Initialize local tolerance $\epsilon_i > 0$, penalty factor $\mu_i \triangleq \mu_i^0 > 0$. Choose parameters $\sigma_i > 1$, $0 < \epsilon_i < 2$, $0 < \gamma_i < \frac{2-\epsilon_i}{M}$, and $0 < \rho_i < 1$. Receive information from links and derive an initial source rate s_i^0 . Send the information $\rho_i, \epsilon_i, \gamma_i$ and $|\mathcal{L}_i|$ to links $j \in \mathcal{L}_i$.

2) Local schedule update

a) State initialization: wait for all links $j \in \mathcal{L}_i$ to send their link states $\hat{\alpha}_j$ and set $\hat{\alpha}_j = \alpha_j$.

b) Update source rate:

$$z_i \triangleq \nabla U_i(s_i) + \frac{\gamma_i}{\mu_i} \frac{1}{s_i} - \gamma_i \sum_{j \in \mathcal{L}_i} \hat{\alpha}_j,$$

$$s_i \triangleq s_i + \gamma_i z_i.$$

c) Communication protocol: Transmit z_i to all links $j \in \mathcal{L}_i$ when the following condition is true

$$z_i^2 < \rho_i \hat{z}_i^2,$$

and set $\hat{z}_i = z_i$.

3) Increase μ_i . $\mu_i \triangleq \sigma_i \mu_i$ if

$$|z_i| \leq \epsilon_i,$$

until $\mu_i \delta_i > 2S$. Inform the links $j \in \mathcal{L}_i$ that source i performed a barrier update.

4) Repeat Step 2.

Algorithm 2: Link j 's Update Algorithm

1) State initialization: Initialize $\lambda_j \triangleq \lambda_j^0$, and choose parameter $\eta_j > 1$. Wait for users to return z_i , and $I_i = 0$ for all $i \in \mathcal{S}_j$ and set $\hat{z}_i = z_i$.

2) Link update: Monitor the link state

$$\alpha_j = \frac{1}{\lambda_j} \frac{1}{c_j - r_j^T s}.$$

3) Communication protocol: Transmit α_j to all sources in $i \in \mathcal{S}_j$ when the following condition is true

$$(\alpha_j(k) - \hat{\alpha}_j(k))^2 > \frac{\sum_{i \in \mathcal{S}_j} \frac{\rho_i \gamma_i}{|\mathcal{L}_i|} \left(1 - \frac{\epsilon_i}{2} - \frac{M}{2} \gamma_i \right) \hat{z}_i^2}{\sum_{i \in \mathcal{S}_j} \frac{\gamma_i |\mathcal{L}_i|}{2\epsilon_i}},$$

and set $\hat{\alpha}_i = \alpha_i$.

- 4) Barrier update notification: If link j receives a notice that source i performed a barrier update, set $I_i = 1$. If $I_i = 1$ for all $i \in \mathcal{S}_j$, then set $\lambda_j \triangleq \eta_j \lambda_j$, reset $I_i = 0$, and broadcast λ_j to all $i \in \mathcal{S}_j$ until $\lambda_j \delta_i > 2L$.
- 5) Go to Step 2.

Remark 6: Some comments are made on the accuracy of the solution to centering problems. Computing $s_i(k)$ exactly is not necessary since the central path has no significance beyond the fact that it leads to a solution to the original problem; inexact centering will still yield a sequence of points that converges to the optimal point. On the other hand, the cost of computing an extremely accurate minimizer as compared to the cost of computing a good minimizer is only marginally more. For this reason it is not unreasonable to assume exact centering.

Remark 7: There are a few parameters involved here. For clarification, the role of each parameter is explained as follows. The stopping criterion for inner iteration of source i is determined by ϵ_i , which is related to the decrement of the objective function involved in source i . The choice of the parameters σ_i and η_j involves a trade-off in the number of inner and outer iterations required. If σ_i and η_j are small, then at each outer iteration μ_i and λ_j increase by a small factor. Values from around 10 to 20 or so seem to work well. The parameter ϵ_i is used for Young's inequality; ρ_i is the parameter of source i 's event detector. Let $\delta = \max\{\delta_i\}$, then δ is a certificate for $f(s^*) - p \leq \delta$.

We will prove that the point $s^*(k)$ is δ -suboptimal, where δ is the desired accuracy of the difference between the objective function value at the approximate solution and the true optimal value.

Theorem 3: Under the assumptions of U_i , R and ρ in Theorem 1, the flow rates $s_i(k)$ generated by algorithms 1 and 2 converge asymptotically to the δ -suboptimal solution to the NUM problem.

Proof. By algorithms 1 and 2, all the parameters satisfy $\mu_i \geq \frac{2S}{\delta_i}$ for all $i \in \mathcal{S}$, and $\lambda_j \geq \frac{2L}{\delta_j}$ for all $j \in \mathcal{L}$. They are fixed after the conditions are satisfied. ■

VI. NUMERICAL EXAMPLES

To verify the proposed method, a simulation is demonstrated as follows. The utility functions are chosen as

$$U_i(s_i) = w_i \log s_i,$$

where w_i are random variables uniformly distributed on $[0.8, 1.2]$ to distinguish different sources. Functions $U_i(s_i)$ obey the assumptions of utility functions obviously. A network of 4 sources and 3 links is set up and shown in Figure 1. Link j is assigned a capacity c_j uniformly distributed on $[0.8, 1.2]$.

The settings for simulation are illustrated as follows. The initial condition is generated distributively according to (17) with $\theta_i = 0.95$, for all $i \in \mathcal{S}$, which is kept in the feasible set. The initial values for multipliers λ_j and μ_i are 1, which are increased by $\sigma_i = \eta_j = 10$ during each outer interaction. Parameters ρ_i of the triggering logic for all sources are chosen randomly from a uniform distribution on $[0, 1]$. Figure 2 plots the flow rate for each source over time. After a period of

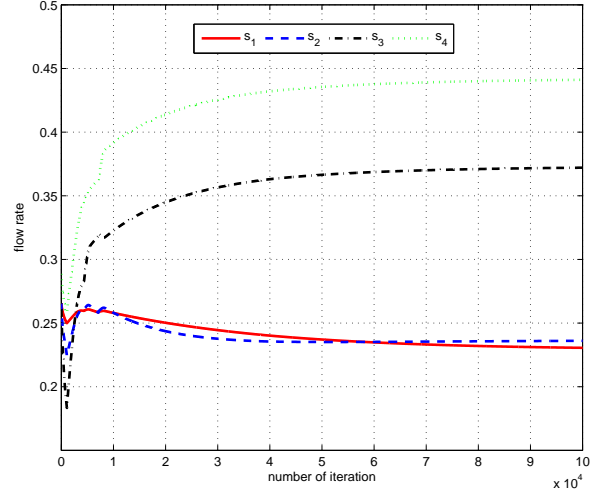


Fig. 2. Flow rate for each source

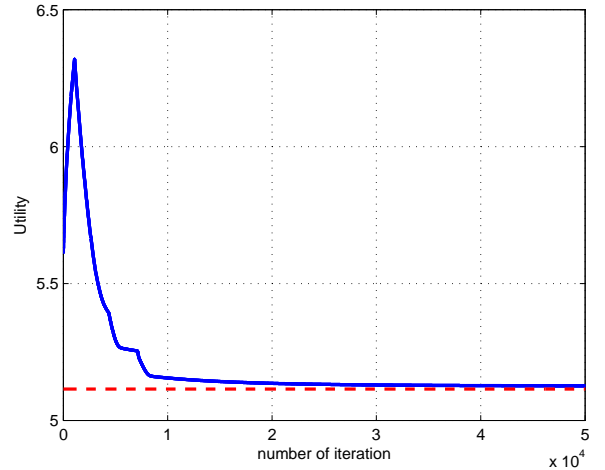


Fig. 3. Evolution of the opposite of the objective function

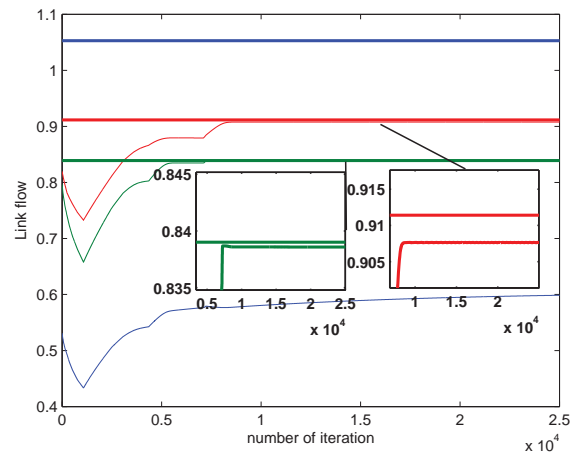


Fig. 4. Evolution of flow in each link

TABLE I
THE TRANSMISSION RATES OF SOURCES AND LINKS

	Links			Sources			
	L_1	L_2	L_3	S_1	S_2	S_3	S_4
No. Events	3	525	66	13	30	52	2
Transmission Rate %	0.0366	6.4017	0.8048	0.1585	0.3658	0.6341	0.0244

transit time, all flow rates tend to be a steady value. Figure 3 shows the opposite of the aggregate utility over time, that is, $-\sum_{i \in \mathcal{S}} U_i(s_i)$. In the figure, the red dashed line means the opposite of the maximum utility, where the optimal rate s^* and its corresponding utility U^* are calculated using a global optimization technique. The objective function does not show a monotonic behavior at the beginning. This is due to the Lagrangian parameters update. The objective function is monotonically decreasing only during each inner iteration, that is, when the Lagrangian parameters are fixed. The derived sub-optimal result is very close to the optimal utility value. For real-time implementation, it is very important to guarantee that the constraints are satisfied all the time. That is, the problem of overflow does not happen. Figure 4 shows the aggregate flow for each link over time. When flow rates are approaching the optimal values, the total flow through each link is close to the link capacity limit. There is still plenty of room in Link 1's capacity. However, since other links are already near their capacity limits, the flow in link 1 cannot be increased. This happens because the link capacity in the simulation is generated randomly. Actually, the link capacity should be well designed before installation.

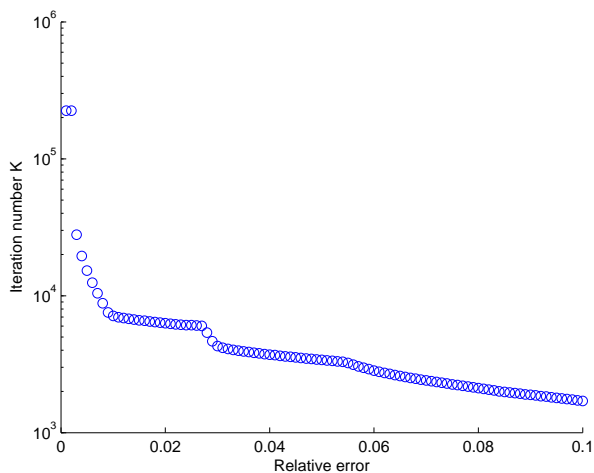


Fig. 5. Number of iterations with respect to relative errors

Define the relative error as $e(k) = \left| \frac{U(s_i(k)) - U^*}{U^*} \right|$, where $s(k)$ is the rate at time k , and $e(k)$ is the normalized derivation from the optimal point at the k th iteration. The number of iteration K is counted for $e(k)$ to decrease to and stay in the neighborhood $\{e(k) | e(k) \leq e_r\}$. For 1% relative error, the algorithm stops after 8201 iterations. Table I shows the number of information exchange and the corresponding transmission rate. It is easy to see that the communication

cost is reduced significantly.

To see the effect of e_r on the algorithm, the error e_r is varied from 0.1% to 10%, while keeping all other parameters unchanged. The resulting Figure 5 plots the iteration number K as a function of e_r . As e_r increases from 0.1% to 10%, the iteration number K decreases from 225977 to 1705. There is an underlying result that the number of iterations based on event triggered barrier methods increases dramatically for high precision solutions [12].

VII. CONCLUSION

We presented a barrier method for distributed optimization with event triggered communication. The proposed method is applicable to problems with a convex objective function constrained by a convex set. We showed the convergence of the proposed algorithm and gave an accurate estimate. Moreover, numerical results show that the application of the proposed algorithm to network utility maximization problems with local event triggered communication can significantly reduce information exchange between sources and links without the loss of accuracy.

REFERENCES

- [1] X. Meng and T. Chen, "Optimal sampling and performance comparison of periodic and event based impulse control," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3252–3259, 2012.
- [2] B. Wang, X. Meng, and T. Chen, "Event based pulse-modulated control of linear stochastic systems," *IEEE Trans. Autom. Control*, vol. 59, no. 8, pp. 2144–2150, 2014.
- [3] X. Meng and T. Chen, "Optimality and stability of event triggered consensus state estimation for wireless sensor networks," in *Proc. of American Control Conference (ACC)*, 2014, pp. 3565–3570.
- [4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.
- [5] J. Nocedal and S. Wright, *Numerical optimization*, 2nd ed. Springer, 2006.
- [6] X. Meng, "Distributed event triggered control, estimation, and optimization for cyber-physical systems," Ph.D. dissertation, University of Alberta, 2014.
- [7] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2123–2132, 2013.
- [8] M. Zhong and C. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2735–2750, 2010.
- [9] M. Meinel, M. Ulbrich, and S. Albrecht, "A class of distributed optimization methods with event-triggered communication," *Computational Optimization and Applications*, vol. 57, no. 3, pp. 517–553, 2014.
- [10] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2254–2269, 2007.
- [11] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization—i: Algorithm," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2162–2175, 2013.
- [12] M. Lemmon, *Networked Control Systems*, ser. Lecture Notes in Control and Information Sciences. Berlin/Heidelberg: Springer, 2011, vol. 406, ch. Event-triggered feedback in control, estimation, and optimization, pp. 293–358.
- [13] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. Springer, 2008.