Optimal Distributed Gradient Methods for Network Resource Allocation Problems

Amir Beck* Angelia Nedic† Asuman Ozdaglar‡ Marc Teboulle§

February 21, 2013

Abstract

We present an optimal distributed gradient method for the Network Utility Maximization (NUM) problem. Most existing works in the literature use (sub)gradient methods for solving the dual of this problem which can be implemented in a distributed manner. However, these (sub)gradient methods suffer from an $O(1/\sqrt{k})$ rate of convergence (where k is the number of iterations). In this paper, we assume that the utility functions are strongly concave, an assumption satisfied by most standard utility functions considered in the literature. We develop a completely distributed optimal gradient method for solving the dual of the NUM problem. We show that the generated primal sequences converge to the unique optimal solution of the NUM problem at rate $O(1/k)$.

1 Introduction

The unprecedented growth in the scale of communication networks has increased the importance and urgency of efficient scalable and decentralized algorithms for the allocation of resources in such networks. Optimization formulations of the corresponding resource allocation problem provide a powerful approach as exemplified by the canonical *Network Utility Maximization* (NUM) problem proposed in [7] (see also [11, 21] and [4]). NUM problems are characterized by a fixed network and a set of sources, which send information over the network along a predetermined set of links. Each source has a local utility function over the rate at which it sends information. The goal is to determine the source rates that maximize the sum of utilities subject to link capacity constraints.

Existing work has exploited the convexity of the NUM formulation, resulting from the concavity of the utility functions and the linearity of the capacity constraints, to derive a decentralized algorithm using a dual based (sub)gradient method with convergence rate of

*Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology.

†Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign.

‡Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

§School of Mathematical Sciences, Tel-Aviv University.

$O(1/\sqrt{k})$, where k is the number of iterations. Though this approach has proved to be rich and useful both in theory and practice, our starting point in this paper is that in most practically relevant cases, a significant improvement is possible. The reason for this is that in most such applications, utility functions are not just concave but also *strongly concave*. An important implication of this property is that the dual function is not only differentiable but also has a Lipschitz continuous gradient enabling the use of optimal gradient methods with much improved convergence rate.

In this paper, we derive a decentralized optimal dual gradient algorithm for the NUM problem and investigate its implications for the resulting generated primal solutions. Our analysis considers a more general convex optimization problem with linear constraints given by

$$\text{(P)} \qquad \begin{array}{rll} g_{\text{opt}} = & \max & g(\mathbf{x}) \\ & \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{c}, \\ & & \mathbf{x} \in X, \end{array}$$

where $\mathbf{A}$ is an $m \times n$ matrix, $X \subseteq \mathbb{R}^n$ is a closed convex set and g is a strongly concave function over X with parameter $\sigma > 0$. Under the assumption that each utility function u_i is strongly concave over a compact interval $I_i = [0, M_i]$ (where M_i is the maximum allowed rate for source i), the NUM problem is a special case of this problem with $g(\mathbf{x}) = \sum_{i \in \mathcal{S}} u_i(x_i)$, where $\mathcal{S}$ is the set of sources and $X = \Pi_{i \in \mathcal{S}} I_i$. Standard utility functions considered in the literature such as the α-fair utility functions (see [12]) satisfy the strong concavity assumption over the compact interval $I_i = [0, M_i]$.

Under a mild condition, i.e., Slater's condition, strong duality holds and we can solve problem (P) through the use of its dual. We first show that the dual problem of problem (P) can be expressed in terms of the conjugate function of the primal objective function $g(\mathbf{x})$. We then use an important equivalence relation between the differentiability of a convex function and the strong convexity of its conjugate. The equivalence relation enables us to establish that the gradient mapping of the dual function is Lipschitz continuous, thus allowing us to apply an optimal gradient method ([16]) with rate $O(1/k^2)$ to the dual problem. We show that the primal sequence generated by the method converge to the unique optimal solution of problem (P) at rate of $O(1/k)$. We also show that the primal infeasibility converges to 0, and that the objective function value converges to the optimal value at a rate of $O(1/k)$.

We demonstrate that a direct application of the optimal method to the NUM problem will require a centralized implementation since the stepsize needed to ensure convergence (which is a function of the Lipschitz constant of the gradient mapping of the dual function) relies on global information. We therefore develop a scaled version of the optimal gradient method in which each variable uses a different stepsize that depends on local information only, enabling the method to be implemented in a distributed manner while retaining the $O(1/k^2)$ rate of convergence of the dual sequence.

The paper is organized as follows. Section 2 contains the formulations of the NUM problem and its dual, and presents a dual-based gradient method for this problem. A fast gradient method is discussed in Section 3, together with its fully distributed implementation. Section 4 presents our simulation setting and reports our numerical results, while Section 5 provides some concluding remarks.

1.1 Notation, Terminology and Basics

We view a vector as a column vector, and we denote by $\mathbf{x}^T\mathbf{y}$ the inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$. We use $\|\mathbf{y}\|_2$ to denote the standard Euclidean norm (or l_2 norm), $\|\mathbf{y}\|_2 = \sqrt{\mathbf{y}^T\mathbf{y}}$ (we drop the subscript and refer to it as $\|\mathbf{y}\|$ whenever it is clear from the context). Occasionally, we also use the standard l_1 norm and l_∞ norm denoted respectively by $\|\mathbf{y}\|_1$ and $\|\mathbf{y}\|_\infty$, i.e., $\|\mathbf{y}\|_1 = \sum_i |y_i|$ and $\|\mathbf{y}\|_\infty = \max_i |y_i|$. For an $m \times n$ matrix $\mathbf{M}$, we use the following induced matrix norm: Given any vector norm $\|\cdot\|$, the corresponding induced matrix norm, also denoted by $\|\cdot\|$, is defined by

$$\|\mathbf{M}\| = \max\{\|\mathbf{M}\mathbf{x}\| : \|x\| = 1\|\}.$$

We next list some standard properties of the induced norm which will be used in our analysis (see [6], Section 5.6 for more details).

Lemma 1.1. *Given any vector norm $\|\cdot\|$ and the induced matrix norm, we have:*

(a) $\|\mathbf{M}\mathbf{x}\| \le \|\mathbf{M}\|\|\mathbf{x}\|$ for all $m \times n$ matrices $\mathbf{M}$ and all vectors $\mathbf{x} \in \mathbb{R}^n$, and $\|\mathbf{N}\mathbf{M}\| \le \|\mathbf{N}\|\|\mathbf{M}\|$ for all matrices $\mathbf{N}$ and $\mathbf{M}$ (with proper dimensions).

(b) $\rho(\mathbf{M}^T\mathbf{M}) \le \|\mathbf{M}^T\mathbf{M}\|$, where $\rho(\mathbf{M}^T\mathbf{M})$ is the spectral radius of matrix $\mathbf{M}^T\mathbf{M}$ (i.e., the maximum of the magnitudes of the eigenvalues of $\mathbf{M}^T\mathbf{M}$).
Moreover, $\|\mathbf{M}\|_2 = \|\mathbf{M}^T\|_2$, $\|\mathbf{M}\|_1 = \|\mathbf{M}^T\|_\infty$, and $\|\mathbf{M}\|_2^2 = \rho(\mathbf{M}^T\mathbf{M})$.

For a concave function $g : \mathbb{R}^n \to [-\infty, \infty)$, we denote the domain of g by $\mathrm{dom}(g)$, where

$$\mathrm{dom}(g) = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) > -\infty\}.$$

We say that $\mathbf{d} \in \mathbb{R}^n$ *is a subgradient of a concave function $g(\mathbf{x})$ at a given vector $\bar{\mathbf{x}} \in \mathrm{dom}(g)$* if the following relation holds:

$$g(\bar{\mathbf{x}}) + \mathbf{d}^T(\mathbf{x} - \bar{\mathbf{x}}) \ge g(\mathbf{x}) \qquad \text{for all } \mathbf{x} \in \mathrm{dom}(g). \tag{1.1}$$

The set of all subgradients of g at $\bar{\mathbf{x}}$ is denoted by $\partial g(\bar{\mathbf{x}})$.

Given a nonempty convex set $C \subseteq \mathbb{R}^n$, a function $g : C \to \mathbb{R}$ is said to be strongly concave over C with a parameter $\sigma > 0$ (in a norm $\|\cdot\|$) if for all $\mathbf{x}, \mathbf{y} \in C$ and all $\gamma \in [0, 1]$,

$$g(\gamma\mathbf{x} + (1-\gamma)\mathbf{y}) - \gamma(1-\gamma)\frac{\sigma}{2}\|\mathbf{x} - \mathbf{y}\|^2 \ge \gamma g(\mathbf{x}) + (1-\gamma)g(\mathbf{y}).$$

We will use the following equivalent characterization of a strongly concave function in our analysis: a function $g : C \to \mathbb{R}$ is strongly concave over C with parameter $\sigma > 0$ if and only if for all $\mathbf{x}, \mathbf{y} \in C$ and all $\mathbf{d} \in \partial g(\mathbf{y})$,

$$g(\mathbf{x}) \le g(\mathbf{y}) + \mathbf{d}^T(\mathbf{x} - \mathbf{y}) - \frac{\sigma}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{1.2}$$

Finally, for any function $h : \mathbb{R}^n \to \mathbb{R}$ which is continuously differentiable and with Lipschitz gradient L_h, we have the so-called descent Lemma (see e.g., [3]):

$$h(\mathbf{x}) \le h(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla h(\mathbf{y})\rangle + \frac{L_h}{2}\|\mathbf{x} - \mathbf{y}\|^2 \quad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{1.3}$$

2 The Network Utility Maximization Problem

Consider a network consisting of a finite set $\mathcal{S}$ of sources and a finite set $\mathcal{L}$ of undirected capacitated links, where a link l has capacity $c_l \geq 0$. Let $\mathcal{L}(i) \subseteq \mathcal{L}$ denote the set of links used by source i, and let $\mathcal{S}(l) = \{i \in \mathcal{S} \mid l \in \mathcal{L}(i)\}$ denote the set of sources that use link l.

Each source i is associated with a utility function $u_i : [0, \infty) \to [0, \infty)$, i.e., each source i gains a utility $u_i(x_i)$ when it sends data at rate x_i. We further assume that the rate x_i is constrained to lie in the interval $I_i = [0, M_i]$ for all $i \in \mathcal{S}$, where the scalar M_i denotes the maximum allowed rate for source i. We adopt the following assumption on the source utility functions.

Assumption 1. *For each i, the function $u_i : [0, \infty) \to [0, \infty)$ is continuous, increasing, and strongly concave over the interval $I_i = [0, M_i]$.*

Note that standard utility functions used in the literature (such as the α-fair utility functions; see [12]) satisfy the strong concavity assumption over the compact interval $I_i = [0, M_i]$.

The goal of the network utility maximization problem (abbreviated NUM), first proposed in [7] (see also [11, 21]), is to allocate the source rates as the optimal solution of the following problem:

$$\begin{array}{ll} \max & g_{\mathrm{N}}(\mathbf{x}) \equiv \sum_{i \in \mathcal{S}} u_i(x_i) \\ \text{s.t.} & \sum_{i \in \mathcal{S}(l)} x_i \leq c_l \qquad \text{for all } l \in \mathcal{L}, \\ & x_i \in I_i \qquad \text{for all } i \in \mathcal{S}. \end{array}$$

Let us consider the $|\mathcal{L}| \times |\mathcal{S}|$ network matrix $\mathbf{A}$ with entries given by

$$A_{li} = \begin{cases} 1 & l \in \mathcal{L}(i), \\ 0 & \text{else.} \end{cases} \tag{2.1}$$

Then, by letting $\mathbf{x} = (x_1, \ldots, x_{|\mathcal{S}|})^T$ and $\mathbf{c} = (c_1, \ldots, c_{|\mathcal{L}|})^T$, the problem can be compactly represented as

$$\begin{array}{lll} & \max & g_{\mathrm{N}}(\mathbf{x}) = \sum_{i \in \mathcal{S}} u_i(x_i) \\ \text{(N-P)} & \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{c}, \\ & & x_i \in I_i \qquad \text{for all } i \in \mathcal{S}. \end{array}$$

In our analysis we will also consider a more general model of a linearly constrained maximization problem:

$$\begin{array}{lll} g_{\mathrm{opt}} = & \max & g(\mathbf{x}) \\ \text{(P)} & \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{c}, \\ & & \mathbf{x} \in X, \end{array}$$

where $\mathbf{A}$ is an $m \times n$ matrix and

- the set $X \subseteq \mathbb{R}^n$ is closed and convex;
- the function g is a strongly concave over X with a parameter $\sigma > 0$ in the Euclidean norm.

Problem (P), as a problem of maximizing a concave function over a convex set, is a convex optimization problem. Moreover, by the strong concavity assumption on the function g, problem (P), whenever feasible, has a unique solution, denoted by $\mathbf{x}^*$. Problem (N-P) obviously fits into the general model (P) with $g(\mathbf{x}) = g_{\mathrm{N}}(\mathbf{x}) = \sum_{i\in\mathcal{S}} u_i(x_i)$ and $X = \Pi_{i\in\mathcal{S}} I_i$ and $g_{\mathrm{N}}(\mathbf{x})$ a strongly concave function over X with the constant $\sigma = \min_{i\in\mathcal{S}} \sigma_i$.

2.1 The Dual of (P) and its Properties

We will assume that Slater's condition is satisfied.

Assumption 2. *There exists a vector $\tilde{\mathbf{x}}$ in the relative interior of set X such that $\mathbf{A}\mathbf{x} \leq \mathbf{c}$.*

It is well known (see [17]) that, under Assumption 2, strong duality holds for problem (P). We let $\tilde{g}$ denote the extended-valued function associated with the objective function g and the set X, which is given by

$$\tilde{g}(\mathbf{x}) = \begin{cases} g(\mathbf{x}) & \mathbf{x} \in X, \\ \infty & \text{else.} \end{cases}$$

In what follows, we also use the notion of the conjugate of an extended-valued function h given by

$$h^*(\mathbf{y}) = \sup_{\mathbf{x}} \left\{ \mathbf{x}^T\mathbf{y} - h(\mathbf{x}) \right\}.$$

Equipped with the above notations, we can write the dual objective function of (P) as

$$\begin{aligned} q(\boldsymbol{\lambda}) &= \max_{x\in X}\{g(\mathbf{x}) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{c})\} \\ &= (-\tilde{g})^*(-\mathbf{A}^T\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T\mathbf{c} \end{aligned} \tag{2.2}$$

for every $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{L}|}$. Therefore, the dual problem is given by

$$\text{(D)} \quad q_{\text{opt}} = \begin{array}{ll} \min & (-\tilde{g})^*(-\mathbf{A}^T\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T\mathbf{c} \\ \text{s.t.} & \boldsymbol{\lambda} \geq \mathbf{0}. \end{array}$$

By the strong duality property that holds for the pair (P) and (D), we have $g_{\text{opt}} = q_{\text{opt}}$.

Recall also that by duality theory (see e.g., [3]), the dual objective function q is in fact differentiable (by the strong concavity of the primal) and its gradient is given by

$$\nabla q(\boldsymbol{\lambda}) = -\left(\mathbf{A}\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{c}\right), \tag{2.3}$$

where the unique maximizer $\mathbf{x}(\boldsymbol{\lambda})$ is given by

$$x(\boldsymbol{\lambda}) = \operatorname*{argmax}_{x\in X}\{g(\mathbf{x}) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{c})\}.$$

We will use the important equivalence between the differentiability of a convex function and the strong convexity of its conjugate, see e.g., [18, 12.60 Proposition, page 565].

Lemma 2.1. *Let $h : \mathbb{E} \to (-\infty, \infty]$ be a proper, lower semicontinuous and convex function, and let $\sigma > 0$. The following statements are equivalent:*

(a) The function h is differentiable and its gradient mapping ∇h is Lipschitz continuous in some norm $\|\cdot\|_{\mathbb{E}}$ with constant $\frac{1}{\sigma}$.

(b) The conjugate function $h^* : \mathbb{E}^* \to (-\infty, \infty]$ is σ-strongly convex with respect to the dual norm $\|\cdot\|_{\mathbb{E}}^*$.

In our setting, we work with the Euclidean norm, which coincides with its dual norm, and the function $-\tilde{g}$ is σ-strongly convex in this norm, since $-g$ is σ-strongly convex.

Coming back to the NUM problem (N-P), we can exploit the special structure of the objective function to obtain

$$(-\tilde{g}_{\mathrm{N}})^*(-\mathbf{A}^T\boldsymbol{\lambda}) = \sum_{i\in\mathcal{S}}(-\tilde{u}_i)^*(-(\mathbf{A}^T\boldsymbol{\lambda})_i) = \sum_{i\in\mathcal{S}}(-\tilde{u}_i)^*(-\pi_i(\boldsymbol{\lambda})),$$

where $\pi_i(\boldsymbol{\lambda}) = \sum_{l\in\mathcal{L}(i)}\lambda_l$ and $\tilde{u}_i$ is the extended valued function associated with the function u_i and set I_i given by

$$\tilde{u}_i(x) = \begin{cases} u_i(x) & x \in I_i, \\ \infty & \text{else.} \end{cases}$$

Consequently, the dual problem of the network utility maximization problem (N-P) is given by

$$\text{(N-D)} \qquad \begin{array}{ll} \min & q_{\mathrm{N}}(\boldsymbol{\lambda}) \equiv \sum_{i\in\mathcal{S}}(-\tilde{u}_i)^*(-(\mathbf{A}^T\boldsymbol{\lambda})_i) + \mathbf{c}^T\boldsymbol{\lambda} \\ \text{s.t.} & \boldsymbol{\lambda} \geq \mathbf{0}. \end{array}$$

Moreover, as recorded before, with this special choice of g, the resulting dual objective function q_{N} is differentiable and its gradient is given by

$$\nabla q_{\mathrm{N}}(\boldsymbol{\lambda}) = -\left(\mathbf{A}\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{c}\right),$$

where

$$x_i(\boldsymbol{\lambda}) = \underset{x_i\in I_i}{\operatorname{argmax}}\{u_i(x_i) - \pi_i(\boldsymbol{\lambda})x_i\} \qquad \text{for every } i \in \mathcal{S}.$$

Since we assume that the functions $u_i, i \in \mathcal{S}$, are not only strictly concave, but are in fact *strongly* concave over I_i the gradient of the objective function ∇q_{N} is Lipschitz continuous by Lemma 2.1.

2.2 A Dual-Based Gradient Method

One approach for constructing a solution method for (D) (and thus also for (P)) is to disregard the Lipschitz continuity of the gradient ∇q and employ a gradient projection method for solving the dual problem with a constant stepsize α_G. The method generates dual variables $\boldsymbol{\lambda}^k$ according to the following rule:

Gradient Method with Constant Stepsize
Step 0: Choose $\boldsymbol{\lambda}_0 \geq \mathbf{0}$.
Step k: (for $k \geq 1$)

$$\mathbf{x}^{k-1} = \operatorname*{argmax}_{\mathbf{x}\in X}\{g(\mathbf{x}) - (\boldsymbol{\lambda}^{k-1})^T(\mathbf{A}\mathbf{x} - \mathbf{c})\} \tag{2.4}$$

$$\boldsymbol{\lambda}^{k} = \left[\boldsymbol{\lambda}^{k-1} + \alpha_G(\mathbf{A}\mathbf{x}^{k-1} - \mathbf{c})\right]_+, \tag{2.5}$$

where $[\cdot]_+$ is the projection on the non-negative orthant in $\mathbb{R}^m$.

For the NUM problem (i.e., $g = g_{\rm N}$ and $q = q_{\rm N}$), the constraint set X and the objective function are separable in components of the variables vector $\mathbf{x}$ since $X = \Pi_{i\in\mathcal{S}} I_i$ and $g_N(\mathbf{x}) = \sum_{i\in S} u_i(x_i)$. This allows decoupling step (2.4) as

$$x_i^{k-1} = \operatorname*{argmax}_{x_i\in I_i}\left\{u_i(x_i) - \left(\sum_{l\in\mathcal{L}(i)}\lambda_l^{k-1}\right)x_i\right\} \qquad \text{for all } i \in \mathcal{S}, \tag{2.6}$$

Moreover, step (2.5) can be written as

$$\lambda_l^k = \left[\lambda_l^{k-1} + \alpha\left(\sum_{i\in\mathcal{S}(l)} x_i^{k-1} - c_l\right)\right]_+ \qquad \text{for all } l \in \mathcal{L}. \tag{2.7}$$

Hence, each link l can update its dual variable λ_l in step (2.7) by using the aggregated rates $\sum_{i\in\mathcal{S}(l)} x_i^{k-1}$ of users that utilize the link and its own link capacity value c_l. Moreover, each source i can update its rate in step (2.6) by using its own utility function u_i and the aggregated dual variables $\sum_{l\in\mathcal{L}(i)}\lambda_l^{k-1}$ for the links that serve the source. Hence, as long as there is a feedback mechanism that sends the aggregated information (along the links used by a source) back to the source (which is the case in practical flow control protocols), the preceding updates can be implemented using local information available to each source and destination.

The decomposition properties of these two steps have been observed in [7], which motivated interest in using dual decomposition and subgradient projection methods to solve network resource allocation problems (see e.g., [7, 11, 21, 19, 5]). To address the rate of convergence of such dual methods, a subgradient method with averaging has been considered in [13, 14], which is motivated by a primal-recovery approach proposed in [15], see also [20, 8, 9, 10]. The primal recovery approach constructs the primal sequence, denoted by $\{\hat{\mathbf{x}}^k\}$, as a running average of the iterate sequence $\{\mathbf{x}^k\}$, i.e.,

$$\hat{\mathbf{x}}^k = \frac{1}{k+1}\sum_{t=0}^{k}\mathbf{x}^t.$$

As shown in [13], the averages of the iterates generated by the method with a constant stepsize, do not necessarily converge. However, the function values approach the optimal value within an error depending on the stepsize value, while the feasibility violation diminishes at rate $O(1/k)$.

None of the aforementioned works makes use of the strong concavity of the utility functions and, thus, the results there remain within the domain of non-smooth convex optimization. The major disadvantages of such an approach are: (1) it suffers from the slow $O(1/\sqrt{k})$ rate of convergence of subgradient methods[1] and (2) the distributed implementation dictates a constant stepsize choice which essentially does not guarantee convergence to the optimal value but rather to a value in an interval surrounding the optimal value. In the next section we will show how to overcome the mentioned disadvantages by exploiting the Lipschitz continuity of the gradient of the dual objective function. To accommodate the proposed method, we need to provide the Lipschitz gradient constant, which is established in the following lemma together with some other properties of the dual objective function.

Lemma 2.2. *Consider the function $q(\boldsymbol{\lambda}) = (-\tilde{g})^*(-\mathbf{A}^T\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T\mathbf{c}$ defined in Eq. (2.2). The following statements hold for the function q.*

(a) The function q has a Lipschitz continuous gradient with constant $\frac{\rho(\mathbf{A}^T\mathbf{A})}{\sigma}$, where $\rho(\mathbf{A}^T\mathbf{A})$ is the spectral radius of the matrix $\mathbf{A}^T\mathbf{A}$.

(b) For any $\boldsymbol{\lambda} \geq \mathbf{0}$ we have

$$\frac{\sigma}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}^*\|^2 \leq q(\boldsymbol{\lambda}) - q(\boldsymbol{\lambda}^*) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}),$$

where

$$\mathbf{x}(\boldsymbol{\lambda}) = \operatorname*{argmax}_{\mathbf{x}\in X}\{g(\mathbf{x}) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{c})\},$$

$\mathbf{x}^*$ is the optimal solution of the primal problem (P), and $\boldsymbol{\lambda}^* \geq \mathbf{0}$ is an optimal solution of the dual problem (D).

Proof. (a) By the definition of q, we have

$$\nabla q(\boldsymbol{\lambda}) = -\mathbf{A}\nabla(-\tilde{g})^*(-\mathbf{A}^T\boldsymbol{\lambda}) + \mathbf{c}.$$

The function $-\tilde{g}$ is proper,convex lower-semicontinuous, and hence it conicides with its biconjugate (see e.g., [17, Theorem 12.2, page 104]). Since $\tilde{g}$ is also strongly convex with parameter σ, applying Lemma 2.1, with $h := (-\tilde{g})^*$, we have that $(-\tilde{g})^*$ has a Lipschitz continuous gradient with the constant $\frac{1}{\sigma}$. Combining this with the properties of induced norms given in Lemma 1.1, we obtain for all $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \mathbb{R}^{|\mathcal{L}|}$,

$$\begin{aligned}
\|\nabla q(\boldsymbol{\lambda}_1) - \nabla q(\boldsymbol{\lambda}_2))\| &\leq \frac{1}{\sigma}\|\mathbf{A}\|\|\mathbf{A}^T(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)\| \\
&\leq \frac{1}{\sigma}\|\mathbf{A}\|\|\mathbf{A}^T\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_)\| \\
&= \frac{\rho(\mathbf{A}^T\mathbf{A})}{\sigma}\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|,
\end{aligned}$$

proving the stated Lipschitz gradient property for q.

[1]Despite the fact that the dual objective function is differentiable, if we do not assume that it has a Lipschitz gradient, the convergence results are no better than those known for the nonsmooth case.

(b) We next consider $q(\boldsymbol{\lambda}) - q(\boldsymbol{\lambda}^*)$ for an arbitrary but fixed $\boldsymbol{\lambda} \geq \mathbf{0}$. Since g is a strongly concave function with parameter σ, so is the function $h(\mathbf{x}) \equiv g(\mathbf{x}) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{c})$. This, combined with the fact that $\mathbf{x}(\boldsymbol{\lambda})$ is the optimal solution of $\max_{\mathbf{x}\in X}\{g(\mathbf{x}) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{c})\}$, implies that

$$h(\mathbf{x}(\boldsymbol{\lambda})) - h(\mathbf{x}) \geq \frac{\sigma}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}\|^2 \qquad \text{for every } \mathbf{x} \in X.$$

In particular, using the above inequality with $\mathbf{x} = \mathbf{x}^*$, we have

$$h(\mathbf{x}(\boldsymbol{\lambda})) - h(\mathbf{x}^*) \geq \frac{\sigma}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}^*\|^2.$$

By the definition of h we have

$$\begin{aligned} h(\mathbf{x}(\boldsymbol{\lambda})) - h(\mathbf{x}^*) &= g(\mathbf{x}(\boldsymbol{\lambda})) - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{c}) - g(\mathbf{x}^*) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}) \\ &= q(\boldsymbol{\lambda}) - q(\boldsymbol{\lambda}^*) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}), \end{aligned}$$

where we use $g(\mathbf{x}^*) = q(\boldsymbol{\lambda}^*)$, which holds by strong duality for the primal problem (P). Therefore,

$$q(\boldsymbol{\lambda}) - q(\boldsymbol{\lambda}^*) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}) \geq \frac{\sigma}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}^*\|^2.$$

□

3 A Fast Gradient Method (FGM) for the Dual

Our approach is to utilize the Lipschitz gradient property of the dual objective function by applying an $O(1/k^2)$ optimal gradient method to the dual problem. We will show in Section 3.1, an $O(1/k)$ rate of convergence of the primal sequence can be established without the need of any primal averaging.

3.1 The Method

Since the objective function of the dual problem (D) has a Lipschitz gradient, in order to solve the problem (D), we can invoke an optimal gradient method, such as the one devised by Nesterov in 1983 [16] (see also [2]). At this point, we will not concern ourselves with the exact optimal gradient method that is, or can be used, and instead we will assume that there exists an algorithm that generates a sequence $\{\boldsymbol{\lambda}^k\}_{k=1}^{\infty}$ satisfying

$$q(\boldsymbol{\lambda}^k) - q^* \leq \frac{C}{k^2} \qquad \text{for } k = 1, 2, \ldots, \tag{3.1}$$

where $C > 0$ is some constant. The above inequality is quite often interpreted as follows: in order to obtain an ε-optimal solution of the dual problem (D), one requires at most $O(1/\sqrt{\varepsilon})$ iterations. Of course, we can also define a corresponding primal sequence by

$$\mathbf{x}^k = \operatorname*{argmax}_{\mathbf{x}\in X}\{g(\mathbf{x}) - (\boldsymbol{\lambda}^k)^T(\mathbf{A}\mathbf{x} - \mathbf{c})\} \qquad \text{for } k = 1, 2, \ldots. \tag{3.2}$$

The primal iterates $\mathbf{x}^k$ are not necessarily feasible (in fact, if $\mathbf{x}^k$ is feasible for some k, then it coincides with the *optimal* solution of (P)) and the natural question is whether the sequence $\mathbf{x}^k$ converges to the unique optimal solution $\mathbf{x}^*$ and, if so, at what rate? These questions are answered in the following theorem.

Theorem 3.1. *Suppose that* $\{\boldsymbol{\lambda}^k\} \subseteq \mathbb{R}^m_+$ *is a sequence satisfying (3.1) and let* $\{\mathbf{x}^k\}$ *be the sequence defined by (3.2). Then, for all* $k \geq 1$ *we have:*

(1) $\|\mathbf{x}^k - \mathbf{x}^*\| \leq \sqrt{\frac{2C}{\sigma}}\frac{1}{k}$.

(2) $[\mathbf{A}\mathbf{x}^k - \mathbf{c}]_+ \leq \left(\|\mathbf{A}\|_{2,\infty}\sqrt{\frac{2C}{\sigma}}\frac{1}{k}\right)\mathbf{e}_m$, *where for a matrix* $\mathbf{M}$, $\|\mathbf{M}\|_{2,\infty} \equiv \max\{\|\mathbf{M}\mathbf{x}\|_\infty : \|\mathbf{x}\|_2 = 1\}$ *and* $\mathbf{e}_m$ *is the* m*-dimensional vector with all entries equal to 1.*

(3) If g *is Lipschitz continuous over* X *with a constant* L_g*, then* $g_{\text{opt}} - g(\mathbf{x}^k) \leq L_g\sqrt{\frac{2C}{\sigma}}\frac{1}{k}$.

Proof. (1) By Lemma 2.2(b), with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^k$ and $\mathbf{x}(\boldsymbol{\lambda}) = \mathbf{x}^k$, we have

$$q(\boldsymbol{\lambda}^k) - q(\boldsymbol{\lambda}^*) + (\boldsymbol{\lambda}^k)^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}) \geq \frac{\sigma}{2}\|\mathbf{x}^k - \mathbf{x}^*\|^2. \tag{3.3}$$

On the other hand, since $\boldsymbol{\lambda}^*$ is a dual optimal solution , we obtain

$$q(\boldsymbol{\lambda}^k) - q(\boldsymbol{\lambda}^*) + (\boldsymbol{\lambda}^k)^T(\mathbf{A}\mathbf{x}^* - \mathbf{c}) \leq q(\boldsymbol{\lambda}^k) - q_{\text{opt}} \leq \frac{C}{k^2}, \tag{3.4}$$

where the first inequality follows from the two inequalities $\mathbf{A}\mathbf{x}^* \leq \mathbf{c}$ and $\boldsymbol{\lambda}^k \geq \mathbf{0}$. Combining (3.3) and (3.4) we conclude that

$$\|\mathbf{x}^k - \mathbf{x}^*\|^2 \leq \frac{2C}{\sigma k^2},$$

establishing the first part of the theorem.
(2) We have

$$\|\mathbf{A}\mathbf{x}^k - \mathbf{c} - (\mathbf{A}\mathbf{x}^* - \mathbf{c})\|_\infty = \|\mathbf{A}(\mathbf{x}^k - \mathbf{x}^*)\|_\infty \leq \|\mathbf{A}\|_{2,\infty}\|\mathbf{x}^k - \mathbf{x}^*\| \leq \|\mathbf{A}\|_{2,\infty}\sqrt{\frac{2C}{\sigma}}\frac{1}{k},$$

and in particular,

$$\mathbf{A}\mathbf{x}^k - \mathbf{c} - (\mathbf{A}\mathbf{x}^* - \mathbf{c}) \leq \left(\|\mathbf{A}\|_{2,\infty}\sqrt{\frac{2C}{\sigma}}\frac{1}{k}\right)\mathbf{e}_m.$$

Since $\mathbf{A}\mathbf{x}^* - \mathbf{c} \leq \mathbf{0}$, it follows that $-(\mathbf{A}\mathbf{x}^* - \mathbf{c}) \geq \mathbf{0}$, thus implying that

$$\mathbf{A}\mathbf{x}^k - \mathbf{c} \;\; \leq \;\; \left(\|\mathbf{A}\|_{2,\infty}\sqrt{\frac{2C}{\sigma}}\frac{1}{k}\right)\mathbf{e}_m,$$

and the desired relation for the feasibility violation follows.
(3) A direct consequence of part 1 of the theorem. □

We have thus shown that $\mathbf{x}^k \to \mathbf{x}^*$ with a rate of $O(1/k)$, and that the constraint violation measured by $[\mathbf{A}\mathbf{x}^k - \mathbf{c}]_+$ is also of the order $O(1/k)$. Therefore, we obtain the interesting fact that although the convergence rate of the sequence of dual objective functions is of the rate $O(1/k^2)$, the convergence rate of the primal sequence and its corresponding objective function values, is of the order $O(1/k)$. Next, we will show how such a dual-based method can be implemented for the network utility maximization problem.

As an example, in order to solve the NUM problem, we can use the optimal gradient method of Nesterov [16] (see also [2]) for solving problem (N-D) and obtain the following method:

Optimal Gradient Method
Input: $L_{\rm N}$ - a Lipschitz constant of ∇q_N.
Step 0. Take $\boldsymbol{\xi}^1 = \boldsymbol{\lambda}^0 \in \mathbb{R}^{|\mathcal{L}|}$, $t_1 = 1$.
Step k. ($k \geq 1$) Compute

$$\begin{aligned} \boldsymbol{\lambda}^k &= \left[\boldsymbol{\xi}^k - \frac{1}{L_{\rm N}}\nabla q_{\rm N}(\boldsymbol{\xi}^k)\right]_+ \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2}, \\ \boldsymbol{\xi}^{k+1} &= \boldsymbol{\lambda}^k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}). \end{aligned}$$

Then, the following convergence result holds, as proven in [16, 2].

Theorem 3.2. *Let $\{\boldsymbol{\lambda}^k\}$ be the sequence generated by the optimal gradient method. Then, for all $k \geq 0$,*

$$q_{\rm N}(\boldsymbol{\lambda}^k) - q_{\rm opt} \leq \frac{2L_{\rm N}\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2}{(k+1)^2}.$$

The main problem in applying such a scheme in a distributed way is that a backtracking procedure for determining the stepsize is not possible. It turns out that utilization of a constant stepsize that ensures convergence requires the knowledge of the Lipschitz constant of ∇q_N, which regretfully depends on the information from all the sources. An illustration of this fact is shown in the next lemma that derives a Lipschitz constant.

Lemma 3.1. *The following is a Lipschitz constant for the mapping ∇q_N:*

$$L_{\rm N} = \left(\max_{i\in\mathcal{S}} \frac{1}{\sigma_i}\right) \cdot \max_{i\in\mathcal{S}} |\mathcal{L}(i)| \cdot \max_{l\in\mathcal{L}} |\mathcal{S}(l)|. \tag{3.5}$$

Proof: First, let

$$h(\boldsymbol{\mu}) \equiv \sum_{i\in\mathcal{S}} (-\tilde{u}_i)^*(\mu_i), \quad \boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}|},$$

so that $q_N(\boldsymbol{\lambda}) = h(-\mathbf{A}^T\boldsymbol{\lambda}) + \mathbf{c}^T\boldsymbol{\lambda}$ and $\nabla q_N(\boldsymbol{\lambda}) = -\mathbf{A}\nabla h(-\mathbf{A}^T\boldsymbol{\lambda}) + \mathbf{c}$. Since for every $i \in \mathcal{S}$ the function u_i, and hence also $\tilde{u}_i$, is strongly concave with parameter $\sigma_i > 0$, it follows that $(-\tilde{u}_i)^*$ has a Lipschitz derivative with constant $\frac{1}{\sigma_i}$. Therefore,

$$\|\nabla h(\boldsymbol{\mu}_1) - \nabla h(\boldsymbol{\mu}_2)\| \leq \left(\max_{i\in\mathcal{S}} \frac{1}{\sigma_i}\right) \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|, \quad \text{for every } \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^{|\mathcal{S}|}.$$

Thus, using the properties of the induced norm given in Lemma 1.1, for every $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \mathbb{R}^{|\mathcal{L}|}$, we have

$$\begin{aligned}\|\nabla q_N(\boldsymbol{\lambda}_1) - \nabla q_N(\boldsymbol{\lambda}_2)\| &\leq \| - \mathbf{A}(\nabla h(-\mathbf{A}^T\boldsymbol{\lambda}_1) - \nabla h(-\mathbf{A}^T\boldsymbol{\lambda}_2))\| \\ &\leq \left(\max_{i\in\mathcal{S}} \frac{1}{\sigma_i}\right) \|\mathbf{A}\| \cdot \|\mathbf{A}^T(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)\| \\ &\leq \left(\max_{i\in\mathcal{S}} \frac{1}{\sigma_i}\right) \|\mathbf{A}\|\|\mathbf{A}^T\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\| \\ &= \left(\max_{i\in\mathcal{S}} \frac{1}{\sigma_i}\right) \|\mathbf{A}\|^2\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|.\end{aligned}$$

In addition,

$$\|\mathbf{A}\|^2 = \rho(\mathbf{A}^T\mathbf{A}) \leq \|\mathbf{A}^T\mathbf{A}\|_\infty \leq \|\mathbf{A}^T\|_\infty\|\mathbf{A}\|_\infty = \|\mathbf{A}\|_1\|\mathbf{A}\|_\infty.$$

With $\mathbf{A}$ defined in (2.1) we have

$$\|\mathbf{A}\|_1 = \max_{i\in\mathcal{S}} |\mathcal{L}(i)|, \quad \|\mathbf{A}\|_\infty = \max_{l\in\mathcal{L}} |\mathcal{S}(l)|,$$

establishing the desired result. □

Computation of a Lipschitz constant of ∇q_N, such as L_{N} given in (3.5), will require communication between all the sources in the network, and this is not possible when only local communication is permitted. We are therefore led to discuss scaled versions of optimal gradient methods in which each variable has its own stepsize which depends only on local information.

3.2 A Distributed Implementation of FGM for NUM

In this section we will show how to exploit the special structure of the dual problem (N-D) in order to establish a fully distributed optimal gradient method for solving it. For ease of notation, we will rewrite problem (N-D) as:

$$\begin{array}{ll}\min & \sum_{i\in\mathcal{S}} h_i\left(-\sum_{l\in\mathcal{L}(i)} \lambda_l\right) + \mathbf{c}^T\boldsymbol{\lambda} \\ \text{s.t.} & \boldsymbol{\lambda} \geq \mathbf{0},\end{array} \tag{3.6}$$

where $h_i(\lambda) \equiv (-\tilde{u}_i)^*(\lambda)$. For an index set I, the vector $\boldsymbol{\lambda}_I$ denotes the subvector of $\boldsymbol{\lambda}$ consisting of the variables $\lambda_j, j \in I$ (e.g., $\boldsymbol{\lambda}_{\{1,3,4\}} = (\lambda_1, \lambda_3, \lambda_4)^T$). We can thus also rewrite (3.6) as

$$\begin{array}{ll}\min & \sum_{i\in\mathcal{S}} H_i\left(-\boldsymbol{\lambda}_{\mathcal{L}(i)}\right) + \mathbf{c}^T\boldsymbol{\lambda} \\ \text{s.t.} & \boldsymbol{\lambda} \geq \mathbf{0},\end{array} \tag{3.7}$$

where for an index set I, $H_i(\boldsymbol{\lambda}_I) = h_i(\sum_{i\in I} \lambda_i)$. Recall that h_i has a Lipschitz derivative with constant $\frac{1}{\sigma_i}$. Therefore, from its definition, it follows that H_i has a Lipschitz gradient with constant $\frac{|\mathcal{L}(i)|}{\sigma_i}$.

Now, for every $i \in \mathcal{S}$ we can write the descent lemma for the function H_i:

$$H_i(\boldsymbol{\lambda}_{\mathcal{L}(i)}) \leq H_i(\boldsymbol{\mu}_{\mathcal{L}(i)}) + \langle \nabla H_i(\boldsymbol{\mu}_{\mathcal{L}(i)}), \boldsymbol{\lambda}_{\mathcal{L}(i)} - \boldsymbol{\mu}_{\mathcal{L}(i)}\rangle + \frac{|\mathcal{L}(i)|}{2\sigma_i}\|\boldsymbol{\lambda}_{\mathcal{L}(i)} - \boldsymbol{\mu}_{\mathcal{L}(i)}\|^2. \tag{3.8}$$

Summing the above inequality over $i \in \mathcal{S}$ and using the fact that $q_{\mathrm{N}}(\boldsymbol{\lambda}) \equiv \sum_{i\in\mathcal{S}} H_i(-\boldsymbol{\lambda}_{\mathcal{L}(i)}) + \mathbf{c}^T\boldsymbol{\lambda}$, we obtain that

$$q_{\mathrm{N}}(\boldsymbol{\lambda}) \leq q_{\mathrm{N}}(\boldsymbol{\mu}) + \langle \nabla q_{\mathrm{N}}(\boldsymbol{\mu}), \boldsymbol{\lambda} - \boldsymbol{\mu}\rangle + \frac{1}{2}(\boldsymbol{\lambda} - \boldsymbol{\mu})^T\mathbf{W}(\boldsymbol{\lambda} - \boldsymbol{\mu}), \tag{3.9}$$

where $\mathbf{W}$ is a positive definite diagonal matrix whose l-th diagonal element is given by

$$W_{ll} = \sum_{i\in\mathcal{S}(l)} \frac{|\mathcal{L}(i)|}{\sigma_i}.$$

It is well known that the key ingredient in proving convergence of gradient-type methods is the existence of a corresponding descent lemma. The weighted descent lemma given by (3.9) can also be used in order to prove the convergence of a corresponding scaled gradient projection method. Indeed, it is very easy to see that the analysis of [2] can be easily extended to the weighted case and the resulting optimal gradient projection method will have the following form:

Optimal Weighted Gradient Projection Method
Step 0. Initialize $\boldsymbol{\eta}^1 = \boldsymbol{\lambda}^0 \in \mathbb{R}_+^{|\mathcal{L}|}$, $t_1 = 1$.
Step k. ($k \geq 1$) Compute

$$\boldsymbol{\lambda}^k = \left[\boldsymbol{\eta}^k - \mathbf{W}^{-1}\nabla q_{\mathrm{N}}(\boldsymbol{\eta}^k)\right]_+, \tag{3.10}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{3.11}$$

$$\boldsymbol{\eta}^{k+1} = \boldsymbol{\lambda}^k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}). \tag{3.12}$$

As already mentioned, the convergence analysis of [2] can be easily extended to the weighted case when all the l_2 norms $\|\mathbf{x}\|$ are replaced by the weighted norm $\|\mathbf{x}\|_{\mathbf{W}}^2 = \sum_i W_{ii}x_i^2$ and the convergence result will be the following:

Theorem 3.3. *Let $\{\boldsymbol{\lambda}^k\}, \{\boldsymbol{\eta}^k\}$ be generated by the optimal weighted gradient projection method. Then for any $k \geq 1$*

$$q_{\mathrm{N}}(\boldsymbol{\lambda}^k) - q(\boldsymbol{\lambda}^*) \leq \frac{2\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_{\mathbf{W}}^2}{(k+1)^2}. \tag{3.13}$$

Note that even disregarding the issues of distributive optimization, the convergence result (3.13) is better than the one obtained when the Lipschitz constant L_{N} given in (3.5) since $\mathbf{W} \preceq L_{\mathrm{N}}\mathbf{I}$ implying that $\|\mathbf{x}\|_{\mathbf{W}}^2 \leq L_{\mathrm{N}}\|\mathbf{x}\|^2$. The additional attribute of this method is of course that it lends itself to a decentralized implementation. The method is described in details below.

Fast Dual-Based Method for Solving NUM

Initialization. For each link $l \in \mathcal{L}$, select λ_l^0 and set $\eta_l^1 = \lambda_l^0$. Let $t_1 = 1$ and

$$\alpha_l = \left(\sum_{i \in \mathcal{S}(l)} \frac{|\mathcal{L}(i)|}{\sigma_i} \right)^{-1}.$$

Step k. For $k \geq 1$, execute the following steps:

(A) **Source-Rate Update:**

$$x_i^{k-1} = \underset{x_i \in I_i}{\operatorname{argmax}} \left\{ u_i(x_i) - \left(\sum_{l \in \mathcal{L}(i)} \eta_l^{k-1} \right) x_i \right\} \qquad \text{for all } i \in \mathcal{S}.$$

(B) **Link-Price Update:**

$$\lambda_l^k = \left[\eta_l^{k-1} + \alpha_l \left(\sum_{i \in \mathcal{S}(l)} x_i^{k-1} - c_l \right) \right]_+ \qquad \text{for all } l \in \mathcal{L}.$$

(C) **Two-Step Network-Price Update:**

(C.1) $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,

(C.2) $\boldsymbol{\eta}^{k+1} = \boldsymbol{\lambda}^k + \left(\frac{t_k - 1}{t_{k+1}} \right) (\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1})$.

For each $l \in \mathcal{L}$, the step sizes α_l only depend on the sources i that use link l (that is, $\mathcal{S}(l)$) and it is assumed that at the beginning of the process each source i sends its strong convexity constant σ_i and the number of links it uses $|\mathcal{L}(i)|$ to all the links on its path (i.e., to all the links it uses). This is the only additional communication that is required for the optimal method. By Theorem 3.1 and Theorem 3.3 we have that the sequence of iterates $\mathbf{x}^k$ converges to the optimal allocation vector at the rate of $O(1/k)$.

4 Numerical Experiments

We compare the performance of the optimal weighted gradient method developed in Section 3.2 with two other distributed algorithms commonly used in the literature for solving the NUM problem: (dual) gradient method explained in Section 2.2 and Newton-type diagonally scaled (dual) gradient method introduced in [1]. We have implemented all three algorithms both on small deterministic networks and also on a randomly generated collection of networks. Our simulation results demonstrate that the proposed fast gradient method significantly outperforms the standard gradient methods in terms of the number of iterations.

We have assumed that all sources have identical utility functions given by $u_i(x_i) = 20 \log(x_i + 0.1)$, where we added the value 0.1 in the argument of the logarithmic function

to prevent numerical instability when x_i is close to 0. We have also assumed that all links have identical capacity given by 1, i.e., for these examples the NUM problem takes the form:

$$\begin{aligned} \max_x \sum_{i\in\mathcal{S}} & 20\log(x_i+0.1) \\ \text{s.t.} \sum_{i\in\mathcal{S}(l)} & x_i \le 1 \quad \text{for all } l\in\mathcal{L}, \\ & x_i \ge 0 \quad \text{for all } i\in\mathcal{S}. \end{aligned}$$

For all three algorithms, we used constant stepsize rules that can guarantee convergence. More specifically, in the price update (2.5) for the gradient method we used a stepsize α_G given by

$$\alpha_G = \frac{2\sigma}{N_p N_s},$$

where σ is a strong convexity constant for the utility functions (taken to be $\sigma = \frac{20}{(1+0.1)^2}$ for these experiments). The scalars N_p and N_s are defined, respectively, as the longest path length among all sources and the maximum number of sources sharing a particular link i.e.,

$$N_p = \max_{s\in\mathcal{S}} |\mathcal{L}(s)|, \qquad N_s = \max_{l\in\mathcal{L}} |\mathcal{S}(l)|.$$

Since in a distributed setting, we do not have information on N_p and N_s, we use the total number of links and sources, i.e., $|\mathcal{L}|$ and $|\mathcal{S}|$, as upper bounds on N_p and N_s, respectively. For the diagonally scaled gradient method, we used a stepsize α_{DS} that satisfies

$$\alpha_{DS} < \frac{2\epsilon\sigma}{N_p N_s},$$

where scalars N_p, N_s and σ are defined as above and ϵ is a positive scalar used to guarantee the Hessian approximation H is positive definite, i.e., if $H_{ll} < \epsilon$ then we use ϵ for that element to avoid singularity.[2] We set $\epsilon = 0.1$ in our experiments, which is the same value used in [1]. For the optimal weighted gradient method, we used the stepsize rule developed in Section 3.2 with $\sigma = \frac{20}{(1+0.1)^2}$.

In our first experiment, we considered the network shown in Figure 1 with two sources (and destinations determined by the set of links used by the sources). The links used by the sources are identified using the flows corresponding to each source. Figure 2 illustrates a sample evolution of the objective function value for each of the three algorithms. The iteration count on the horizontal axis is log-scaled. The blue dotted horizontal lines indicate $\pm 5\%$ interval around the optimal objective function value. Optimal weighted gradient method outperforms the standard gradient method. In this particular example, it also converges faster than the diagonally scaled gradient method.

[2]The Hessian approximation is given as

$$H_{ll} = -\frac{x_l^k - x_l^{k-1}}{\lambda_l^k - \lambda_l^{t-1}},$$

where x_l^k is the flow on link l and λ_l^k is the dual variable associated with link l at iteration k. Hence depending on the initial conditions, the approximated value of H_{ll} may be smaller than ϵ, even though the elements in the exact Hessian are lower bounded by the scalar α.

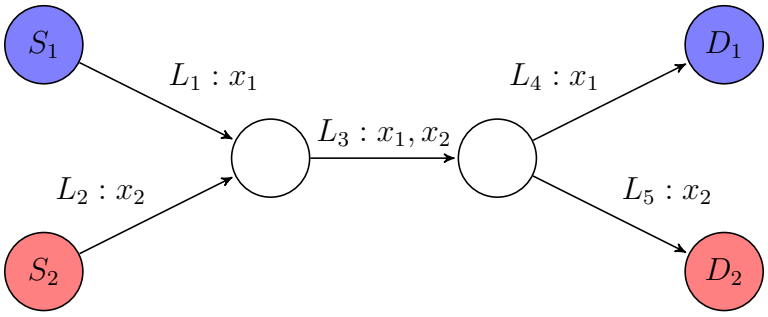

Figure 1: A sample network. Each source-destination pair is displayed with the same color. We use x_i to denote the flow corresponding to the i^{th} source and L_i to denote the i^{th} link.

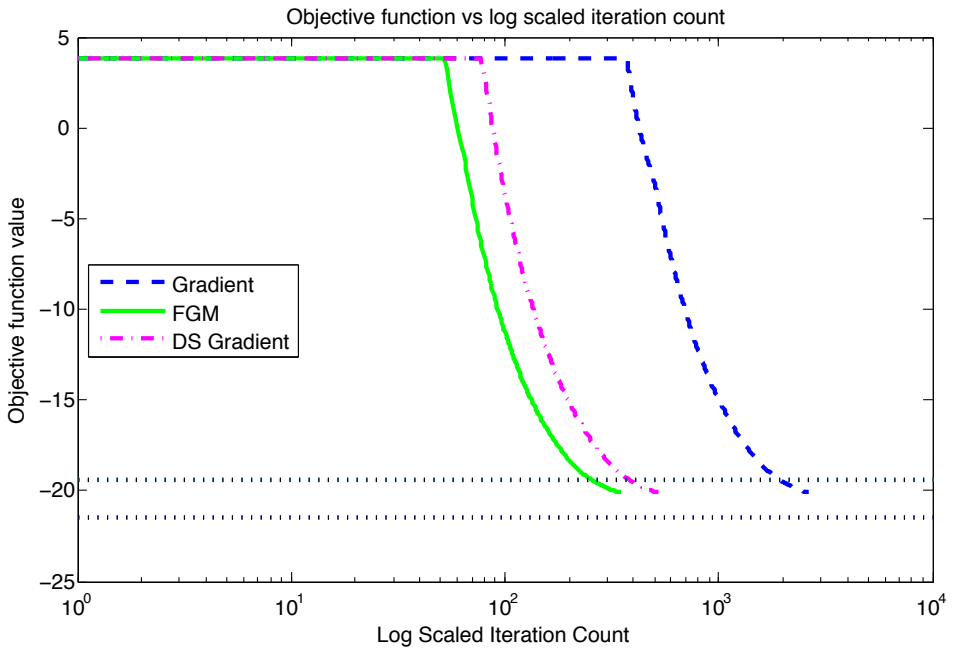

Figure 2: Sample objective function value of all three methods against log scaled iteration count for network in Figure 1. The dotted blue horizontal lines denote $\pm 5\%$ interval of the optimal objective function value.

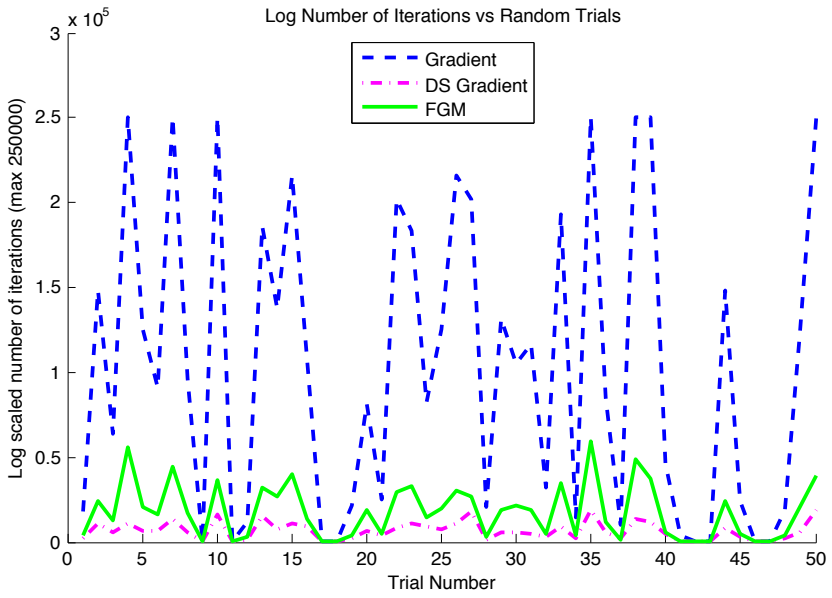

Figure 3: Log scaled iteration count for the three methods implemented over 50 randomly generated networks with random sizes.

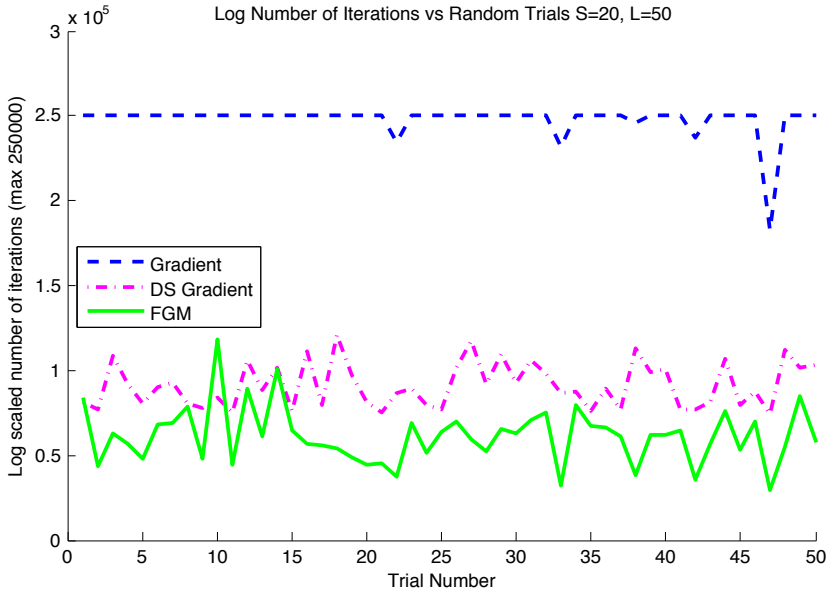

Figure 4: Log scaled iteration count for the three methods implemented over 50 randomly generated networks, each with 20 sources and 50 links.

To test the performance of the algorithms over general networks, we generated 50 random networks, with number of links being a random variable taking integer values in range $[1, 40]$ and number of sources being another independent random variable taking integer values in the interval $[1, 25]$. Each routing matrix consists of $|\mathcal{L}| \times |\mathcal{S}|$ Bernoulli random variables.[3] All three methods are implemented over the 50 networks. The methods were terminated when all of the following conditions are satisfied at an iteration k:

1) Primal objective function value satisfies $\left|\frac{f(x^{k+1})-f(x^k)}{f(x^k)}\right| \leq 0.01$;

2) Dual variable satisfies $\|\lambda^{k+1} - \lambda^k\|_\infty \leq 0.01$;

3) Primal feasibility satisfies $[c - Ax^k]_l \geq -0.01$ for all links l.

To be able to display results properly, we capped the number of iterations at 250000 (this cap was not exceeded in all the trials, except a few times with the gradient method). We record the number of iterations upon termination for all three methods and results are shown in Figure 3 on a log scale. The mean number of iterations to convergence from the 50 trials is 6584.2 for the diagonally scaling gradient method, 17871.6 for the optimal weighted gradient method and 103265.9 for the gradient method.

To study further, the scaling properties of the algorithm with respect to the network size, we generated another set of 50 random trials, with the number of sources $|\mathcal{S}| = 20$ and the number of links $|\mathcal{L}| = 50$. We repeated the same experiment as before and recorded the results in Figure 4. The average number of iterations is 91221 for the diagonally scaled gradient method, 61430 for the optimally weighted gradient method and 247628.6 for the gradient method. This set of results is qualitatively different from that of Figure 3, in the sense that the optimal weighted gradient method is faster than the diagonally scaled gradient method. This can be explained by the difference in stepsize rules used in the two methods. In the diagonally scaled gradient method, the stepsize is proportional to the global quantity $\frac{1}{|\mathcal{L}|\,|\mathcal{S}|}$, whereas in the optimal weighted gradient method, the stepsize is proportional to the local path lengths $\frac{1}{|\mathcal{L}(\rangle)|}$. The latter quantity in general results in larger stepsize choices. Thus for large networks, fast gradient method tends to converge faster than the diagonally scaled gradient method.

5 Conclusion

We have considered the NUM problem and proposed an optimal distributed dual-based gradient method for solving the problem. Our focus was on the NUM problem with strongly concave utility functions. We established the convergence rate of the order $1/k$ for the primal iterate sequences, which demonstrates the superiority of these methods over the standard dual-gradient methods with convergence rate of the order $1/\sqrt{k}$. Furthermore, we have proposed a fully distributed implementation of the optimal gradient method. Our numerical results indicate that the proposed method have significant advantages as compared not only

[3]When there exists a source that does not use any links or a link that is not used by any sources, we discard the routing matrix and generate another one.

to the standard gradient method, but also to the Newton-type diagonally scaled dual-gradient method of [1].

References

[1] S. Athuraliya and S. Low. Optimization flow control with Newton-like algorithm. *Journal of Telecommunication Systems*, 15:345–358, 2000.

[2] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. In D. Palomar and Y. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 139–162. Cambridge University Press, 2009.

[3] D. P. Bertsekas. *Nonlinear Programming*. Belmont MA: Athena Scientific, second edition, 1999.

[4] M. Chiang, S. H. Low, A. R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: a mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[5] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[6] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, NY, 1985.

[7] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *The Journal of the Operational Research Society*, 49(3):pp. 237–252, 1998.

[8] T. Larsson, M. Patriksson, and A. Strömberg. Ergodic results and bounds on the optimal value in subgradient optimization. In P. Kelinschmidt et al., editor, *Operations Research Proceedings*, pages 30–35. Springer, 1995.

[9] T. Larsson, M. Patriksson, and A. Strömberg. Ergodic convergence in subgradient optimization. *Optimization Methods and Software*, 9:93–120, 1998.

[10] T. Larsson, M. Patriksson, and A. Strömberg. Ergodic primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming*, 86:283–312, 1999.

[11] S. H. Low and D. E. Lapsley. Optimization flow control. i. basic algorithm and convergence. *Networking, IEEE/ACM Transactions on*, 7(6):861 –874, dec. 1999.

[12] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transaction on Networking*, 8(5):556–567, 2000.

[13] A. Nedić and A. Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. pages 4711–4716, 2007.

[14] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009.

[15] A.S. Nemirovskii and D.B. Yudin. Cezare convergence of gradient method approximation of saddle points for convex-concave functions. *Doklady Akademii Nauk SSSR*, 239:1056–1059, 1978.

[16] Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[17] R. T. Rockafellar. *Convex Analysis*. Princeton NJ: Princeton Univ. Press, 1970.

[18] R. T. Rockafellar and R. J. B Wets. *Variational analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1998.

[19] S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3):271–379, 2007.

[20] N. Z. Shor. *Minimization Methods for Nondifferentiable Functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer-Verlag, 1985.

[21] R. Srikant. *The mathematics of Internet congestion control*. Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA, 2004.