Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 12, No. 2, pp. 289–305, April 2016

A FAST DUAL GRADIENT METHOD FOR SEPARABLE CONVEX OPTIMIZATION VIA SMOOTHING*

Jueyou Li, Zhiyou Wu, Changzhi Wu†, Qiang Long, Xiangyu Wang, Jae-Myung Lee and Kwang-Hyo Jung

Abstract: This paper considers a class of separable convex optimization problems with linear coupled constraints arising in many applications. Based on a novel smoothing technique, a simple fast dual gradient method is presented to solve this class of problems. Then the convergence of the proposed method is proved and the computational complexity bound of the method for achieving an approximately optimal solution is given explicitly. An improved iteration complexity bound is obtained when the objective function of the problem is strongly convex. Our algorithm is simple and fast, which can be implemented in a parallel fashion. Numerical experiments on a network utility maximization problem are presented to illustrate the effectiveness of the proposed algorithm.

Key words: *convex optimization, dual decomposition, smoothing technique, fast gradient method, parallel computation.*

Mathematics Subject Classification: *90C25, 90C06, 90C90, 65Y05.*

1 Introduction

There is a trend focusing on solving large-scale convex optimization problems in recent years [1–3, 10, 11, 20–22, 28, 29]. This class of optimization problems has extensive practical applications [12, 14–16, 18, 26, 27]. However, methods and algorithms for solving large-scale convex optimization problems are still limited [5, 8].

Convex optimization problems with a separable objective function subject to linear coupled constraints are encountered in many disciplines, e.g., network utility maximum [17]. They are known as separable convex minimization problems, which can be written as:

$$\begin{aligned} \min_{x} \quad & g(x) := \sum_{i=1}^{N} g_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{N} A_i x_i = b,\ x_i \in X_i,\ i = 1, \ldots, N, \end{aligned} \tag{1.1}$$

*This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) through GCRC-SOP (No. 2011-0030013). This research was partially supported by the Australia Research Council LP130100451, the Natural Science Foundation of China (11501070, 11501474, 61473326 and 11471062), the Natural Science Foundation of Chongqing (cstc2015jcyjA0382, cstc2013jjB00001 and cstc2013jcyjA00029), Chongqing Municipal Education Commission under Grant KJ1500301 and the Chongqing Normal University Research Foundation (15XLB005).

†Corresponding author.

where for $i = 1, \ldots, N$, $g_i : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}$ is convex, $X_i \subset \mathbb{R}^{n_i}$ is a nonempty compact convex set, $A_i \in \mathbb{R}^{m \times n_i}$, and $b \in \mathbb{R}^m$. $x = (x_1^\top, \cdots, x_N^\top)^\top$ with $x_i \in \mathbb{R}^{n_i}, i = 1, \ldots, N$ and $n_1 + n_2 + \cdots + n_N = n$.

In literature there are several approaches being proposed to solve problem (1.1), such as (augmented) Lagrangian relaxation and subgradient methods of multipliers [5, 22, 27], alternating direction methods [6], proximal point methods [9,19], and interior point methods [11, 30]. However, most of these methods cannot be implemented in a parallel fashion for solving problem (1.1). To exploit the separable structure of problem (1.1), the Lagrangian duality is often introduced, where the main idea is to solve the primal problem via solving its dual problem. But the dual problem is, in general, nonsmooth. Thus, the use of subgradient-based methods is inevitable. However, it is well-known that the subgradient-based methods usually suffer from the slow convergence [23]. To overcome this drawback, the augmented Lagrangian is introduced. However, it cannot be used here because the quadratic penalty term of the augmented Lagrangian may destroy the separability of problem (1.1).

In [24], Nesterov proposed a novel smoothing technique for solving nonsmooth convex optimization problems appeared in applications, such as networks and system identification, image processing and compressed sensing (see, e.g., [7, 13]). In [22], Nesterov's smoothing technique is applied to the dual problem under the framework of Lagrangian dual decomposition for solving separable convex optimization problems, where it is shown that the iteration complexity of the algorithm for achieving an ϵ-optimal solution is $\mathcal{O}(1/\epsilon)$. It is much superior to $\mathcal{O}(1/\epsilon^2)$ achieved by the subgradient methods for solving the dual problem [25] (ϵ is a specified and desired accuracy). We note that the proximal center algorithm proposed in [22] requires two maximizations where the gradient information in all the previous iterations is needed. This algorithm may need high computational cost for large-scale optimization problems. Recently, a double smoothing technique introduced in [10] for solving large-scale optimization problems is considered as a generalization of Nesterov's smoothing technique, in which a fast gradient scheme [23] is used. The complexity bound of the algorithm obtained in [10] for achieving an ϵ-optimal solution is $\mathcal{O}((1/\epsilon)\ln(1/\epsilon))$.

In this paper we propose a simple fast dual gradient algorithm for solving problem (1.1) motivated by the novel smoothing technique [10, 24] and a simple fast gradient scheme [23]. The proposed algorithm is fast and highly parallelizable, which allows us to obtain the dual and primal approximate solutions simultaneously. The explicitly computational complexity is established. This complexity bound on the number of iterations for achieving an ϵ-optimal solution is $\mathcal{O}((1/\epsilon)\ln(1/\epsilon))$, which is better than $\mathcal{O}(1/\epsilon^2)$ obtained by subgradient-based methods [23]. Although the convergence rate of our algorithm is slightly slower than that of the proximal center algorithm [22], our algorithm is simpler and computationally inexpensive.

The rest of this paper is organized as follows. In Section 2, we recall some concepts and Lagrangian dual decomposition method. In Section 3, a smoothing technique is introduced and a simple fast gradient method for solving the smoothed dual problem is described. In Section 4, the convergence of the algorithm is proved. An application to network utility maximization is presented in Section 5. Some conclusions are given in Section 6.

2 Preliminaries

In this section we introduce some concepts and recall briefly the Lagrangian dual decomposition method for a convex optimization problem with linear constraints.

The standard inner product of two vectors $x, y \in \mathbb{R}^n$ is denoted as $\langle x, y\rangle = x^\top y$, where the subscript "$\top$" denotes the transpose. For $x \in \mathbb{R}^n$, its Euclidean norm is $||x|| = \sqrt{\langle x, x\rangle}$.

Denote l_∞ norm by $||x||_\infty = \max_i |x_i|$. Let $x = (x_1^\top, \cdots, x_N^\top)^\top$ represent a column vector in $\mathbb{R}^n$, where x_i is a subvector in $\mathbb{R}^{n_i}$, $i = 1, \ldots, N$, and $n_1 + \cdots + n_N = n$. For a matrix $A \in \mathbb{R}^{m\times n}$, $||A||$ denotes the 2-norm.

For a function $f : \mathbb{R}^n \to \mathbb{R}$, if there exists a constant $L > 0$ such that $||f(x) - f(y)|| \leq L||x - y||$ for all $x, y \in \mathbb{R}^n$, then we say that f is L-Lipschitz continuous on $\mathbb{R}^n$ with respect to $||\cdot||$. If there exists a constant $\sigma > 0$ such that $f(x) \geq f(y) + d^\top(x - y) + \frac{\sigma}{2}||x - y||^2$ for all $x, y \in \mathbb{R}^n$ and all $d \in \partial f(x)$, then we say that f is σ-strongly convex on $\mathbb{R}^n$ with respect to $||\cdot||$.

Denote by $X = \prod_{i=1}^N X_i$, X^* and g^* the feasible set, the optimal solution set and the optimal value of problem (1.1), respectively. Problem (1.1) is said to satisfy the *Slater's condition* if $ri(X) \cap \{\bar{x} | \sum_{i=1}^N A_i\bar{x}_i = b\} \neq \varnothing$, where $ri(X)$ is the relative interior of the set X. A function p_X is called a proximity function (prox-function, for short) of a given nonempty, closed and convex set $X \subset \mathbb{R}^n$ if p_X is continuous, strongly convex with a convexity parameter $\sigma_X > 0$ and $X \subseteq \mathrm{dom}(p_X)$ [24]. Let x_c be the prox-center of X which is defined as $x_c = \arg\min_{x\in X} p_X(x)$. In this paper, we make the following assumptions.

Assumption 2.1. The optimal solution set X^* is nonempty and the Slater's condition of problem (1.1) holds. For each $i = 1, \ldots, N$, the function g_i is proper, lower semicontinuous and convex (not necessarily differentiable) in $\mathbb{R}^{n_i}$.

Assumption 2.2. Each feasible set X_i is equipped a prox-function p_i with the convexity parameter $\sigma_i > 0$. Moreover, $0 \leq D_i := \max_{x_i\in X_i} p_i(x_i) < \infty$, for $i = 1, \ldots, N$.

The Lagrangian function for problem (1.1) is

$$\mathcal{L}(x, \lambda) = \sum_{i=1}^N g_i(x_i) + \lambda^\top(\sum_{i=1}^N A_i x_i - b),$$

where $\lambda \in \mathbb{R}^m$ is the Lagrangian multiplier. Then, the dual problem can be written as

$$d^* = \max_{\lambda\in\mathbb{R}^m} d(\lambda), \tag{2.1}$$

where

$$d(\lambda) = \min_{x\in X} \mathcal{L}(x, \lambda) \tag{2.2}$$

is the dual function. Note that the dual function $d(\lambda)$ can be computed in a separable fashion as

$$d(\lambda) = \sum_{i=1}^N d_i(\lambda),$$

where $d_i(\lambda) = \min_{x_i\in X_i}\{g_i(x_i) + \lambda^\top A_i x_i\} - b^\top\lambda/N,\ i = 1, \ldots, N$. It is obvious that the dual function d is concave but non-differentiable in general. By Assumption 2.1, the *strong duality* holds [6], that is, $d^* = \max_{\lambda\in\mathbb{R}^m} d(\lambda) = \min_{x\in X}\{g(x) | \sum_{i=1}^N A_i x_i = b\} = g^*$.

3 Smoothing and Fast Dual Gradient Method

According to (2.2), let

$$\phi(\lambda) = \max_{x\in X}\{-\mathcal{L}(x, \lambda)\} = \max_{x\in X}\Big\{ -\sum_{i=1}^N g_i(x_i) - \lambda^\top\Big(\sum_{i=1}^N A_i x_i - b\Big)\Big\}. \tag{3.1}$$

Hence, $d(\lambda) = -\phi(\lambda)$ and the dual problem (2.1) can be equivalently written as

$$d^* = \max_{\lambda \in \mathbb{R}^m} d(\lambda) = -\min_{\lambda \in \mathbb{R}^m} \phi(\lambda) = -\phi^*.$$

Now we consider the problem below

$$\phi^* = \min_{\lambda \in \mathbb{R}^m} \phi(\lambda). \tag{3.2}$$

As shown above, the function ϕ is, in general, non-differentiable and not strongly convex. However, the properties of the differentiability and strong convexity can be ensured by smoothing. The goal of the first smoothing is to obtain an objective function with Lipschitz-continuous gradient, for which we can apply more efficient algorithms for smooth convex optimization. The second smoothing is to obtain a strongly convex dual objective, which is necessary to allow us to reconstruct efficiently a near feasible and optimal primal solution from a near optimal dual solution. To achieve this goal, we follow the argument used in [10, 22].

For any parameter $u > 0$, we smooth the dual objective ϕ as follows:

$$\phi_u(\lambda) \quad = \quad \sum_{i=1}^{N} \max_{x_i \in X_i} \{-g_i(x_i) - \lambda^\top (A_i x_i - b/N) - u p_i(x_i)\}. \tag{3.3}$$

It is clear that the objective function ϕ_u defined in (3.3) is separable in $x_i, i = 1, \ldots, N$. Denote $x_i(\lambda)$ by the optimal solutions of the maximization problem (3.3) in $x_i, i = 1, \ldots, N$. For notational simplicity, we set $D = \sum_{i=1}^N D_i$ and $E = (\sum_{i=1}^N D_i)(\sum_{i=1}^N ||A_i||^2/\sigma_i)$. The following lemma shows the main properties of $\phi_u(\lambda)$.

Lemma 3.1 ([24])**.** (i) *$\phi_u(\lambda)$ is convex and continuously differentiable on $\lambda \in \mathbb{R}^m$;*
(ii) *The gradient $\nabla \phi_u(\lambda) = -[\sum_{i=1}^N A_i x_i(\lambda) - b]$ is Lipschitz continuous with a constant $L_u = \frac{1}{u} \sum_{i=1}^N \frac{||A_i||^2}{\sigma_i}$;*
(iii)

$$\phi_u(\lambda) \le \phi(\lambda) \le \phi_u(\lambda) + uD, \ \forall \lambda \in \mathbb{R}^m. \tag{3.4}$$

Next the second smoothing is applied to the dual objective $\phi_u(\cdot)$, making it strongly convex. We simply add a strongly convex function $\frac{v}{2}||\cdot||^2$ to $\phi_u(\cdot)$ for a parameter $v > 0$, which is a special prox-function. This gives rise to the following objective function:

$$\phi_{u,v}(\lambda) = \phi_u(\lambda) + \frac{v}{2}||\lambda||^2.$$

The new objective function $\phi_{u,v}(\lambda)$ has the following good properties.

Lemma 3.2. *For the function $\phi_{u,v}(\lambda)$, it holds that:*
(i) *$\phi_{u,v}(\lambda)$ is v-strongly convex and continuously differentiable on $\lambda \in \mathbb{R}^m$;*
(ii) *The gradient $\nabla \phi_{u,v}(\lambda) = \nabla \phi_u(\lambda) + v\lambda$ is Lipschitz continuous with Lipschitz constant $L_{u,v} = L_u + v$.*

We now focus on solving the optimization problem below:

$$\min_{\lambda \in \mathbb{R}^m} \phi_{u,v}(\lambda). \tag{3.5}$$

Based on Lemma 3.2, we utilize a simple fast gradient method (see Section 2.2 in [23]) to solve problem (3.5) as follows:

Algorithm 1. Fast dual gradient method for solving problem (3.5)

Initialization: Set $\lambda^0 = \mu^0 = 0 \in \mathbb{R}^m$.

Iteration: For $k \geq 0$, compute $\lambda^{k+1} = \mu^k - \frac{1}{L_{u,v}} \nabla \phi_{u,v}(\mu^k)$,

$$\text{update } \mu^{k+1} = \lambda^{k+1} + \frac{1-\sqrt{\frac{v}{L_{u,v}}}}{1+\sqrt{\frac{v}{L_{u,v}}}}(\lambda^{k+1} - \lambda^k).$$

4 Convergence Analysis

Let $\tilde{\lambda}^*$ be the unique optimal solution of problem (3.5) and let λ^* be an optimal solution of the dual problem (2.1). From Theorem 3.5 in [22], it follows that there exists a sufficiently large number $\Lambda > 0$ such that the set $\{\lambda \in \mathbb{R}^m : ||\lambda|| \leq \Lambda\}$ contains λ^*. This means that

$$||\lambda^*|| \leq \Lambda. \tag{4.1}$$

We assume that the bound (4.1) is already available.

From Theorem 2.2.3 in [23] and $\lambda^0 = 0$, we obtain the sequence $\{\lambda^k\}_{k\geq 0}$ satisfying

$$\begin{aligned} \phi_{u,v}(\lambda^k) - \phi_{u,v}(\tilde{\lambda}^*) &\leq (\phi_{u,v}(0) - \phi_{u,v}(\tilde{\lambda}^*) + \frac{v}{2}||\tilde{\lambda}^*||^2) e^{-k\sqrt{\frac{v}{L_{u,v}}}} \\ &= (\phi_u(0) - \phi_u(\tilde{\lambda}^*)) e^{-k\sqrt{\frac{v}{L_{u,v}}}}. \end{aligned} \tag{4.2}$$

Since $\tilde{\lambda}^*$ is the optimal solution of problem (3.5), we have $\nabla \phi_{u,v}(\tilde{\lambda}^*) = 0$. Therefore, by Theorem 2.1.5 in [23], we obtain

$$||\nabla \phi_{u,v}(\lambda^k)||^2 \leq 2L_{u,v} \left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right) e^{-k\sqrt{\frac{v}{L_{u,v}}}}. \tag{4.3}$$

Because of the strong convexity of $\phi_{u,v}$, it follows from Theorem 2.1.8 in [23] that

$$\begin{aligned} ||\lambda^k - \tilde{\lambda}^*||^2 &\leq \frac{2}{v}(\phi_{u,v}(\lambda^k) - \phi_{u,v}(\tilde{\lambda}^*)) \\ &\overset{(4.2)}{\leq} \frac{2}{v}(\phi_u(0) - \phi_u(\tilde{\lambda}^*)) e^{-k\sqrt{\frac{v}{L_{u,v}}}}. \end{aligned} \tag{4.4}$$

Using Theorem 2.1.8 in [23] again, we obtain

$$||\tilde{\lambda}^*||^2 \leq \frac{2}{v}(\phi_{u,v}(0) - \phi_{u,v}(\tilde{\lambda}^*)) = \frac{2}{v}(\phi_u(0) - \phi_u(\tilde{\lambda}^*) - \frac{v}{2}||\tilde{\lambda}^*||^2),$$

which implies that

$$||\tilde{\lambda}^*|| \leq \sqrt{\frac{1}{v}(\phi_u(0) - \phi_u(\tilde{\lambda}^*))}. \tag{4.5}$$

4.1 Convergence analysis

The goal is to compute an approximate optimal solution for the primal problem (1.1).

Definition 4.1. For any given target accuracy $\epsilon > 0$, if there exist nonnegative constants c_1, c_2 such that

$$\Big|\sum_{i=1}^{N} g_i(\hat{x}_i) - g^*\Big| \le c_1\epsilon \ \text{ and } \ \Big\|\sum_{i=1}^{N} A_i\hat{x}_i - b\Big\|_\infty \le c_2\epsilon.$$

We say that $\hat{x} = (\hat{x}_1^\top, \cdots, \hat{x}_N^\top)^\top \in \mathbb{R}^n$ is an ϵ-optimal feasible solution of the primal problem (1.1).

The definition bounds the distance of the corresponding primal cost from the optimal value and the maximum primal infeasibility for the primal suboptimal solution, respectively. The next result establishes an upper bound on the distance of the objective values $\phi(\lambda^k)$ from the optimal objective value $\phi(\lambda^*)$.

Proposition 4.2. *Let $\{\lambda^k\}_{k\ge 0}$ be the sequence of iterates generated by Algorithm 1. Then, for all $k \ge 0$, it holds that*

$$\phi(\lambda^k) - \phi(\lambda^*) \le (2+\sqrt{2})\,(\phi(0) - \phi(\lambda^*) + uD)\, e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + uD + \frac{v}{2}\Lambda^2.$$

Proof. From (3.4), we have $\phi_u(0) \le \phi(0)$ and $\phi(\lambda^*) - uD \le \phi(\tilde{\lambda}^*) - uD \le \phi_u(\tilde{\lambda}^*)$. Then,

$$\phi_u(0) - \phi_u(\tilde{\lambda}^*) \le \phi(0) - \phi(\lambda^*) + uD. \tag{4.6}$$

Since $\phi_u(\tilde{\lambda}^*) + \frac{v}{2}||\tilde{\lambda}^*||^2 \le \phi_u(\lambda^*) + \frac{v}{2}||\lambda^*||^2$, we have

$$\phi_u(\tilde{\lambda}^*) \le \phi_u(\lambda^*) + \frac{v}{2}||\lambda^*||^2 \overset{(3.4)}{\le} \phi(\lambda^*) + \frac{v}{2}||\lambda^*||^2.$$

Therefore,

$$\phi_u(\lambda^k) - \phi_u(\tilde{\lambda}^*) \overset{(3.4)}{\ge} \phi(\lambda^k) - uD - \phi(\lambda^*) - \frac{v}{2}||\lambda^*||^2.$$

It follows from the above inequality and (4.1) that

$$\phi(\lambda^k) - \phi(\lambda^*) \le \phi_u(\lambda^k) - \phi_u(\tilde{\lambda}^*) + uD + \frac{v}{2}\Lambda^2. \tag{4.7}$$

Since $\phi_{u,v}(\lambda) = \phi_u(\lambda) + \frac{v}{2}||\lambda||^2$, we have

$$\phi_u(\lambda^k) - \phi_u(\tilde{\lambda}^*) \overset{(4.2)}{\le} (\phi_u(0) - \phi_u(\tilde{\lambda}^*))e^{-k\sqrt{\frac{v}{L_{u,v}}}} + \frac{v}{2}(||\tilde{\lambda}^*||^2 - ||\lambda^k||^2). \tag{4.8}$$

Now we estimate $||\tilde{\lambda}^*||^2 - ||\lambda^k||^2$ as follows:

$$\begin{aligned} ||\tilde{\lambda}^*||^2 - ||\lambda^k||^2 &\le ||\tilde{\lambda}^* - \lambda^k||(||\tilde{\lambda}^* - \lambda^k|| + 2||\tilde{\lambda}^*||) \\ &\overset{(4.4)(4.5)}{\le} \frac{2+2\sqrt{2}}{v}(\phi_u(0) - \phi_u(\tilde{\lambda}^*))e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}}. \end{aligned} \tag{4.9}$$

Combining (4.6)-(4.9), the result of the theorem follows readily. The proof is completed. □

Theorem 4.3. *For any given accuracy $\epsilon > 0$, let $\{\lambda^k\}_{k\ge 0}$ be the sequence of dual iterates generated by Algorithm 1. Then, there exists a $k_1 = \mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\epsilon}) > 0$ such that for all $k \ge k_1$, $\phi(\lambda^k) - \phi(\lambda^*) \le \epsilon$.*

Proof. In order to achieve $\phi(\lambda^k) - \phi(\lambda^*) \leq \epsilon$, we require all the three terms in Proposition 4.2 to be less than or equal to $\epsilon/3$. Therefore, we choose the corresponding smoothing parameters in view of the given accuracy $\epsilon > 0$ to be

$$u = u(\epsilon) = \frac{\epsilon}{3D}, \quad v = v(\epsilon) = \frac{2\epsilon}{3\Lambda^2}. \tag{4.10}$$

Then, we have

$$\phi(\lambda^k) - \phi(\lambda^*) \quad \leq \quad (2+\sqrt{2})(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3})e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + \frac{2\epsilon}{3}. \tag{4.11}$$

By choosing

$$k_1 = 2\sqrt{\frac{L_{u,v}}{v}} \ln \frac{3(2+\sqrt{2})\left(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3}\right)}{\epsilon}, \tag{4.12}$$

we can verify easily that, for $k \geq k_1$

$$(2+\sqrt{2})(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3})e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} \leq \frac{\epsilon}{3}.$$

By the definitions of L_u, u, v and $L_{u,v}$, we have

$$\frac{L_{u,v}}{v} = \frac{1}{uv}\sum_{i=1}^{N}\frac{||A_i||^2}{\sigma_i} + 1 = \frac{9\Lambda^2 E}{2\epsilon^2} + 1. \tag{4.13}$$

Substituting (4.13) to (4.12) and taking into consideration of (4.11), we need at most $k_1 = \mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\epsilon})$ iterations such that $\phi(\lambda^k) - \phi(\lambda^*) \leq \epsilon$. □

In order to reconstruct a near optimal and feasible primal solution efficiently, we need to provide an upper bound on the norm of $\nabla\phi_u(\lambda^k)$.

Theorem 4.4. *For any given accuracy $\epsilon > 0$, let $\{\lambda^k\}_{k\geq 0}$ be the sequence of dual iterates generated by Algorithm 1. Suppose that (4.10) holds. Then, there exists a $k_2 = \mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\epsilon})$ such that for all $k \geq k_2$,*

$$||\nabla\phi_u(\lambda^k)|| \leq \frac{\epsilon}{\Lambda}. \tag{4.14}$$

Proof. From Lemma 3.2, it gives rise to

$$||\nabla\phi_u(\lambda^k)|| \quad = \quad ||\nabla\phi_{u,v}(\lambda^k) - v\lambda^k|| \leq ||\nabla\phi_{u,v}(\lambda^k)|| + v||\lambda^k||.$$

The first term on the right-hand side of the inequality above can be estimated as follows:

$$||\nabla\phi_{u,v}(\lambda^k)|| \overset{(4.3)}{\leq} \sqrt{2L_{u,v}\left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}}.$$

For the term $||\lambda^k||$, we have

$$\begin{aligned} ||\lambda^k|| \quad &\leq \quad ||\lambda^k - \tilde{\lambda}^*|| + ||\tilde{\lambda}^*|| \\ &\overset{(4.4)}{\leq} \quad \sqrt{\frac{2}{v}\left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + ||\tilde{\lambda}^*||. \end{aligned} \tag{4.15}$$

Moreover, we note that

$$\begin{aligned} \phi(\lambda^*) + \frac{v}{2}||\lambda^*||^2 \quad &\overset{(3.4)}{\geq} \quad \phi_u(\lambda^*) + \frac{v}{2}||\lambda^*||^2 \geq \phi_u(\tilde{\lambda}^*) + \frac{v}{2}||\tilde{\lambda}^*||^2 \\ &\geq \quad \phi(\lambda^*) - uD + \frac{v}{2}||\tilde{\lambda}^*||^2, \end{aligned}$$

which implies that $||\tilde{\lambda}^*||^2 \leq ||\lambda^*||^2 + \frac{2u}{v}D$. Hence,

$$||\tilde{\lambda}^*|| \leq \sqrt{||\lambda^*||^2 + \frac{2u}{v}D} \overset{(4.10)}{=} \sqrt{||\lambda^*||^2 + \Lambda^2} \leq \sqrt{2}\Lambda. \tag{4.16}$$

Combining the above estimates, we obtain

$$||\nabla\phi_u(\lambda^k)|| \overset{(4.6)(4.10)}{\leq} \left(\sqrt{L_{u,v}} + \sqrt{v}\right)\sqrt{2\left(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3}\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + \frac{2\sqrt{2}\epsilon}{3\Lambda}.$$

For $\epsilon > 0$ fixed, the first term of the above inequality decreases in terms of the iteration counter k. To achieve $||\nabla\phi_u(\lambda^k)|| \leq \frac{\epsilon}{\Lambda}$, we only need to choose

$$k_2 = 2\sqrt{\frac{L_{u,v}}{v}}\ln\frac{3\Lambda\left(\sqrt{L_{u,v}} + \sqrt{v}\right)\sqrt{2\left(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3}\right)}}{(3-2\sqrt{2})\epsilon}. \tag{4.17}$$

By (4.10), k_2 can be written as

$$\begin{aligned} k_2 &= 2\sqrt{\frac{L_{u,v}}{v}}\ln\frac{3\Lambda\left(\sqrt{\frac{3E}{\epsilon} + \frac{2\epsilon}{3\Lambda^2}} + \sqrt{\frac{2\epsilon}{3\Lambda^2}}\right)\sqrt{2\left(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3}\right)}}{(3-2\sqrt{2})\epsilon} \\ &= 2\sqrt{\frac{L_{u,v}}{v}}\left[\ln\left(\sqrt{3E + \frac{2\epsilon^2}{3\Lambda^2}} + \sqrt{\frac{2\epsilon^2}{3\Lambda^2}}\right) + \frac{1}{2}\ln\frac{1}{\epsilon} + \ln\frac{3\Lambda\sqrt{2\left(\phi(0) - \phi(\lambda^*) + \frac{\epsilon}{3}\right)}}{(3-2\sqrt{2})\epsilon}\right], \end{aligned}$$

which means that $k_2 = \mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\epsilon})$. This completes the proof. □

Next we show how an ϵ-optimal feasible solution of the primal problem (1.1) can be constructed from the dual sequence $\{\lambda^k\}_{k\geq 0}$ generated by Algorithm 1. For this purpose we consider the following sequences $\{x_i^k\}_{k\geq 0}, i = 1, \ldots, N$, which is the unique optimal solution of the maximization problem (3.3) for a given λ^k, i.e.,

$$x_i^k = \arg\max_{x_i \in X_i}\{-g_i(x_i) - (\lambda^k)^\top(A_i x_i - b/N) - up_i(x_i)\},\ i = 1, \ldots, N. \tag{4.18}$$

Theorem 4.5. *For any given $\epsilon > 0$, let the sequences $\{\lambda^k\}_{k\geq 0}$ and $\{x_i^k\}_{k\geq 0}, i = 1, \ldots, N$, be generated by Algorithm 1 and (4.18), respectively. Suppose that (4.10) holds. Then, for any $k \geq k_0 = \max\{k_1, k_2\}$, it holds that*

$$\left|\sum_{i=1}^N g_i(x_i^k) - g^*\right| \leq 5\epsilon \quad \textit{and} \quad \left\|\sum_{i=1}^N A_i x_i^k - b\right\|_\infty \leq \frac{\epsilon}{\Lambda}.$$

Proof. From (3.3) and (4.18), we have $\phi_u(\lambda^k) = -\sum_{i=1}^N g_i(x_i^k) - (\lambda^k)^\top\big(\sum_{i=1}^N A_i x_i^k - b\big) - u\sum_{i=1}^N p_i(x_i^k)$. Note that $g^* = d^* = -\phi(\lambda^*)$ and

$$\nabla\phi_u(\lambda^k) = -\Big(\sum_{i=1}^N A_i x_i^k - b\Big), \tag{4.19}$$

it follows that

$$\sum_{i=1}^N g_i(x_i^k) - g^* = (\lambda^k)^\top\nabla\phi_u(\lambda^k) - u\sum_{i=1}^N p_i(x_i^k) - \phi_u(\lambda^k) + \phi(\lambda^*).$$

Since $\phi_u(\lambda^k) - \phi(\lambda^*) \le \phi(\lambda^k) - \phi(\lambda^*) \le \epsilon$ and

$$\phi_u(\lambda^k) - \phi(\lambda^*) \overset{(3.4)}{\ge} \phi(\lambda^k) - uD - \phi(\lambda^*) \overset{(4.10)}{\ge} \phi(\lambda^k) - \phi(\lambda^*) - \frac{\epsilon}{3} \ge -\frac{\epsilon}{3},$$

we have $|\phi_u(\lambda^k) - \phi(\lambda^*)| \le \epsilon$. Therefore,

$$\left|\sum_{i=1}^N g_i(x_i^k) - g^*\right| \le \left\|\lambda^k||||\nabla\phi_u(\lambda^k)\right\| + uD + \epsilon \overset{(4.10)}{\le} ||\lambda^k||||\nabla\phi_u(\lambda^k)|| + 2\epsilon.$$

In light of (4.15) and (4.16), it holds that

$$\begin{aligned}\left\|\lambda^k\right\| &\le \sqrt{\frac{2}{v}\left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + \sqrt{2}\Lambda \\ &\overset{(4.10)}{=} \Lambda\sqrt{\frac{3}{\epsilon}\left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + \sqrt{2}\Lambda.\end{aligned}$$

Due to the choice of k_0 and the above estimates, we obtain

$$\begin{aligned}\left|\sum_{i=1}^N g_i(x_i^k) - g^*\right| &\le \sqrt{3\epsilon\left(\phi_u(0) - \phi_u(\tilde{\lambda}^*)\right)}e^{-\frac{k}{2}\sqrt{\frac{v}{L_{u,v}}}} + (2+\sqrt{2})\epsilon \\ &\overset{(4.17)}{\le} (3-2\sqrt{2})\epsilon + (2+\sqrt{2})\epsilon \le 5\epsilon.\end{aligned}$$

Finally, using the facts that $||\cdot||_\infty \le ||\cdot||$, we get from (4.19) that

$$\left|\sum_{i=1}^N A_i x_i^k - b\right|_\infty \le ||\nabla\phi_u(\lambda^k)|| \overset{(4.14)}{\le} \frac{\epsilon}{\Lambda}.$$

□

4.2 Improving iteration complexity under strong convexity

Now we show that an additional assumption on the primal objective function g can be used to improve the iteration complexity. More specifically, for each $i = 1, \ldots, N$, if the function g_i is σ_i-strongly convex, then the iteration complexity for achieving an ϵ-optimal solution can be reduced from $\mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\epsilon})$ to $\mathcal{O}(\frac{1}{\sqrt{\epsilon}}\ln\frac{1}{\epsilon})$.

Since for each $i = 1, \ldots, N$, g_i is σ_i-strongly convex, $\phi(\lambda)$ is already differentiable. Thus, the first smoothing of the dual problem can be omitted. Now Algorithm 1 is applied to the minimization problem:

$$\min_{\lambda\in\mathbb{R}^m} \phi_v(\lambda), \tag{4.20}$$

where $\phi_v(\lambda) = \phi(\lambda) + \frac{v}{2}||\lambda||^2$ with $v > 0$. Clearly, $\phi_v(\lambda)$ is a v-strongly convex and differentiable. The Lipschitz constant of its gradient is $L_v = \sum_{i=1}^N \frac{||A_i||^2}{\sigma_i} + v$.

Denote $\tilde{\lambda}^*$ as the unique optimal solution of problem (4.20). Algorithm 1 yields a sequence $\{\lambda^k\}_{k\ge0}$ satisfying

$$\phi_v(\lambda^k) - \phi_v(\tilde{\lambda}^*) \le (\phi(0) - \phi(\tilde{\lambda}^*))e^{-k\sqrt{\frac{v}{L_v}}}. \tag{4.21}$$

$$||\nabla\phi_v(\lambda^k)||^2 \le 2L_v(\phi(0) - \phi(\tilde{\lambda}^*))e^{-k\sqrt{\frac{v}{L_v}}}. \tag{4.22}$$

$$||\lambda^k - \tilde{\lambda}^*||^2 \le \frac{2}{v}(\phi(0) - \phi(\tilde{\lambda}^*))e^{-k\sqrt{\frac{v}{L_v}}}. \tag{4.23}$$

In addition, we have

$$||\tilde{\lambda}^*|| \leq \sqrt{\frac{1}{v}(\phi(0) - \phi(\tilde{\lambda}^*))}, \tag{4.24}$$

and

$$||\tilde{\lambda}^*||^2 - ||\lambda^k||^2 \overset{(4.23),(4.24)}{\leq} \frac{2+2\sqrt{2}}{v}(\phi(0) - \phi(\tilde{\lambda}^*))e^{-\frac{k}{2}\sqrt{\frac{v}{L_v}}}. \tag{4.25}$$

Combining with the relations obtained above, we have

$$\phi(\lambda^k) - \phi(\tilde{\lambda}^*) \overset{(4.21),(4.25)}{\leq} (2+\sqrt{2})(\phi(0) - \phi(\tilde{\lambda}^*))e^{-\frac{k}{2}\sqrt{\frac{v}{L_v}}}.$$

Let λ^* be an optimal solution to the dual optimization problem (3.2). Since $\phi(\tilde{\lambda}^*) \leq \phi_u(\tilde{\lambda}^*) \leq \phi_u(\lambda^*) = \phi(\lambda^*) + \frac{v}{2}||\lambda^*||^2$ and $\phi(\lambda^*) \leq \phi(\tilde{\lambda}^*)$, we obtain, for any $k \geq 0$,

$$\phi(\lambda^k) - \phi(\lambda^*) \leq (2+\sqrt{2})(\phi(0) - \phi(\lambda^*))e^{-\frac{k}{2}\sqrt{\frac{v}{L_v}}} + \frac{v}{2}\Lambda^2.$$

In order to guarantee ϵ-accuracy, we force both terms in the above estimate to be less than or equal to $\epsilon/2$. To achieve this task, we only need to take $v = v(\epsilon) = \epsilon/\Lambda^2$ and set

$$k = 2\sqrt{\frac{L_v}{v}} \ln \frac{2(2+\sqrt{2})(\phi(0) - \phi(\lambda^*))}{\epsilon},$$

i.e., $k = \mathcal{O}(\frac{1}{\sqrt{\epsilon}} \ln \frac{1}{\epsilon})$. Thus, after k iterations, we can obtain that $\phi(\lambda^k) - \phi(\lambda^*) \leq \epsilon$.

On the other hand, the relation $\phi(\lambda^*) + \frac{v}{2}||\tilde{\lambda}^*||^2 \leq \phi_u(\tilde{\lambda}^*) \leq \phi_u(\lambda^*) = \phi(\lambda^*) + \frac{v}{2}||\lambda^*||^2$ yields $||\tilde{\lambda}^*|| \leq ||\lambda^*|| \leq \Lambda$. Thus, we have

$$||\nabla\phi(\lambda^k)|| \overset{(4.22),(4.23)}{\leq} (\sqrt{L_v} + \sqrt{v})\sqrt{2(\phi(0) - \phi(\lambda^*))}e^{-\frac{k}{2}\sqrt{\frac{v}{L_v}}} + \frac{\epsilon}{\Lambda}.$$

Therefore, in order to guarantee $||\sum_{i=1}^N A_i x_i^k - b||_\infty \leq ||\nabla\phi_u(\lambda^k)|| \leq 2\epsilon/\Lambda$, we need to take

$$k = 2\sqrt{\frac{L_v}{v}} \ln \frac{\Lambda(\sqrt{L_v} + \sqrt{v})\sqrt{2(\phi(0) - \phi(\lambda^*))}}{\epsilon},$$

i.e., $k = \mathcal{O}(\frac{1}{\sqrt{\epsilon}} \ln \frac{1}{\epsilon})$, which has the same order as that of $\phi(\lambda^k)$ to $\phi(\lambda^*)$.

4.3 Fast dual gradient algorithm

Interestingly, while Algorithm 1 handles the smoothed dual problem (3.5), it directly yields the primal optimal solution of problem (1.1). Assuming that the minimizations over $x_i, i = 1, \ldots, N$ in (4.18) can be easily carried out, we rewrite Algorithm 1 as follow:

Algorithm 2: Fast dual gradient algorithm (FDGA)

Initialization: Given $u > 0, v > 0$ by (4.10) and $\alpha = (1 - \sqrt{\frac{v}{L_{u,v}}})/(1 + \sqrt{\frac{v}{L_{u,v}}})$. Set $\mu_j^0 = \lambda_j^0 = 0 \in \mathbb{R}, j = 1, \ldots, m$.

Iteration: For $k \geq 0$, execute

Step 1. Receive μ^k and update primal variables in *parallel*: for $i = 1, \ldots, N$

$$x_i^{k+1} = \arg\max_{x_i \in X_i} \{-g_i(x_i) - \mu^{k\top}(A_i x_i - b/N) - up_i(x_i))\}.$$

Step 2. Receive x^{k+1} and update dual variables in *parallel*: for $j = 1, \ldots, m$

$$\nabla_j \phi_{u,v}(\mu^k) = -\left(\sum_{i=1}^{N} A_i x_i^{k+1} - b\right)_j + v\mu_j^k, \tag{4.26}$$

$$\lambda_j^{k+1} = \mu_j^k - \frac{1}{L_{u,v}} \nabla_j \phi_{u,v}(\mu^k), \tag{4.27}$$

$$\mu_j^{k+1} = \lambda_j^{k+1} + \alpha(\lambda_j^{k+1} - \lambda_j^k). \tag{4.28}$$

Step 3. If a given stopping criterion is satisfied, then terminate.

Remark 4.6. (1) The parallel computation appearing in Steps 2 and 3 of Algorithm 2 is helpful for solving large-scale separable convex optimization. We can see this from numerical experiments given in Section 5.

(2) Based on Theorem 4.5, a terminating criterion that does not involve unknown quantities, such as the optimal value g^*, can be established. More specifically, since $g^* = -\phi^*$ and $0 \le \phi(\lambda^k) - \phi^* \le \epsilon$, it follows from Theorem 4.5 that $-5\epsilon \le \sum_{i=1}^{N} g_i(x_i^k) + \phi(\lambda^k) \le 6\epsilon$, where $d(\lambda^{k+1})$ can be computed during the course of the algorithm. Therefore, at each iteration, one can test ϵ-optimality and ϵ-feasibility by examining whether $-5\epsilon \le \sum_{i=1}^{N} g_i(x_i^k) + \phi(\lambda^k) \le 6\epsilon$ and $||[\sum_{i=1}^{N} A_i x_i^k - b]_+||_\infty \le \epsilon/\Lambda$ are satisfied or not. If both of them are satisfied, the algorithm terminates.

(3) In Algorithm 2, the intermediate variables μ_j^k are adopted so as to achieve fast convergence. Clearly, if $\mu_j^{k+1} = \lambda_j^{k+1}$ is taken to replace (4.28), then the expected step number to achieve an ϵ-optimal solution will increase to $\mathcal{O}(1/\epsilon^2)$. In addition, if the equality constraints are replaced by inequalities $\sum_{i=1}^{N} A_i x_i \le b$ in (1.1), we can still obtain the results through similar arguments as given above.

(4) In Algorithm 2, the main computational cost is consumed in Step 1 which is dependent on the choice of prox-function. The simplest prox-function is $p_i(x_i) = ||x_i - x_c^i||^2/2$ for a given proximal center $x_c^i \in X_i$, which is adopted from the one considered in [10]. However, in some applications, through choosing a customized prox-function for the given feasible set X_i, we can reduce the computational complexity of Step 1 in Algorithm 2 (see [24] for more details).

5 Application to Network Utility Maximum Optimization

The separable convex optimization problems with linear coupled constraints have an interesting application to network utility maximization (NUM) problem considered in [4,17]. More specifically, a network is modeled as a set of links $\mathcal{L}$ with finite capacities $C = (C_l, l \in \mathcal{L})$. They are shared by a set of sources $\mathcal{S}$ indexed by s. Each source s uses a set $\mathcal{L}(s) \subset \mathcal{L}$ of links. Let $\mathcal{S}(l) = \{s \in \mathcal{S} | l \in \mathcal{L}(s)\}$ be the set of sources using link l. The set $\{\mathcal{L}(s)\}$ defines an $|\mathcal{L}| \times |\mathcal{S}|$ routing matrix A with entries given by $A_{ls} = 1$ if $l \in \mathcal{L}(s)$, $A_{ls} = 0$ otherwise. Each source s is associated with a utility function $U_s : \mathbb{R}^+ \to \mathbb{R}$, i.e., source s gains a utility $U_s(x_s)$ when it sends data at rate x_s that satisfies $0 \le m_s \le x_s \le M_s$.

Let $I_s = [m_s, M_s]$. Notice that the *local* utility function U_s is private, only known by the source s. Since the aggregate source rate at any link utilizes the available link capacity, we assume that the aggregate source rate at each link is equal to the link capacity. Then, the NUM problem is to determine the source rates that minimize the sum of disutilities with link capacity constraints:

$$\begin{aligned} \text{(NUM)} \quad \min_{x_s \in I_s} \quad & g_N(x) = \sum_{s \in \mathcal{S}} -U_s(x_s) \\ \text{s.t.} \quad & Ax = C. \end{aligned}$$

For numerical simulation, the α-fair utility function is taken as $U_s(x_s) = w_s \log(x_s + 0.1)$ from [4]. Setting $C_l = 1$ for all $l \in \mathcal{L}$ and $w_s = 10, m_s = 0, M_s = 1, s \in \mathcal{S}$. Then, the disutility function $-10\log(x_s + 0.1)$ is σ_s-strongly convex on [0,1] with $\sigma_s = 10/(1+0.1)^2 = \sigma$. Therefore, the first smoothing is omitted and u is set as 0 in Algorithm 2. As it has been proven, the estimate of the number of iterations to achieve an ϵ-optimal solution is $\mathcal{O}((1/\sqrt{\epsilon})\ln(1/\epsilon))$.

We compare the performance of our proposed method with the dual gradient algorithm (DGA) in [17]. The following numerical procedure is motivated by [4], but the optimization method proposed in this paper is different from [4].

We first generate a random routing matrix A with elements 0 and 1, with the number of links $\mathcal{L} = 50$ and the number of resources $\mathcal{S} = 20$. Figure 1 (a) depicts the objective function values versus the number of iterations. Figure 1 (b) depicts the constraint violations in terms of the iterations. Figure 1 (a) shows that Alg. FDGA achieves a faster convergence than Alg. DGA. More specially, after 1000 iterations, the objective function values obtained by Alg. FDGA is very close to the optimal value of the primal problem. In order to achieve a similar accuracy, Alg. DGA requires at least 3000 steps. Figure 1 (b) shows similar results for algorithms FDGA and DGA. Furthermore, it also shows that the constraint violations always exist even after 10000 iterations for Alg. DGA. However, this problem seems to unhappen in Algorithm FDGA.

We further test and compare the performances of the two algorithms over general networks. We generated 50 random networks, with number of links being a random integer taking values between 20 to 50, and number of sources being another independent random integer taking values between 10 and 20. Each routing matrix A is a random matrix with elements either 0 or 1. Both algorithms are terminated when either $k \geq 10000$ or the following three conditions are satisfied: (1) $\max_{l \in \mathcal{L}} |\lambda_l^{k+1} - \lambda_l^k| \leq \epsilon$, (2) $\max_{l \in \mathcal{L}} [Ax^{k+1} - C]_l \leq \epsilon$, (3) $\max_{s \in \mathcal{S}} |[U_s(x_s^{k+1}) - U_s(x_s^k)]/U_s(x_s^k)| \leq \epsilon$, where $\epsilon = 0.01$. We record the number of iterations upon termination for both algorithms. The results are depicted in Figure 2 which clearly shows that the number of iterations of Alg. FDGA is much less than that required by Alg. DGA. The mean number of iterations to convergence from the 50 trials is 4826.4 for Alg. DGA and 2564.7 for Alg. FDGA. Thus, Alg. FDGA achieves a better performance than Alg. DGA.

We generate a set of 50 random trials, with the number of links $\mathcal{L} = 100$ and the number of resources $\mathcal{S} = 40$ for studying the scaling properties of both algorithms with respect to the network size. Under the same computational environment and parameters setting, the results obtained are reported in Figure 3. For all 50 trials, Alg. DGA cannot achieve the targeted accuracy $\epsilon = 0.01$ while Alg. FDGA are successful for all the trials. The average number of iterations to achieve convergence for Alg. FDGA is 6022.5. Compared Figure 2 with Figure 3, we observe that the number of iteration to achieve a given accuracy becomes larger as the size of network is increasing. But the increasing rate of Alg. FDGA is far less

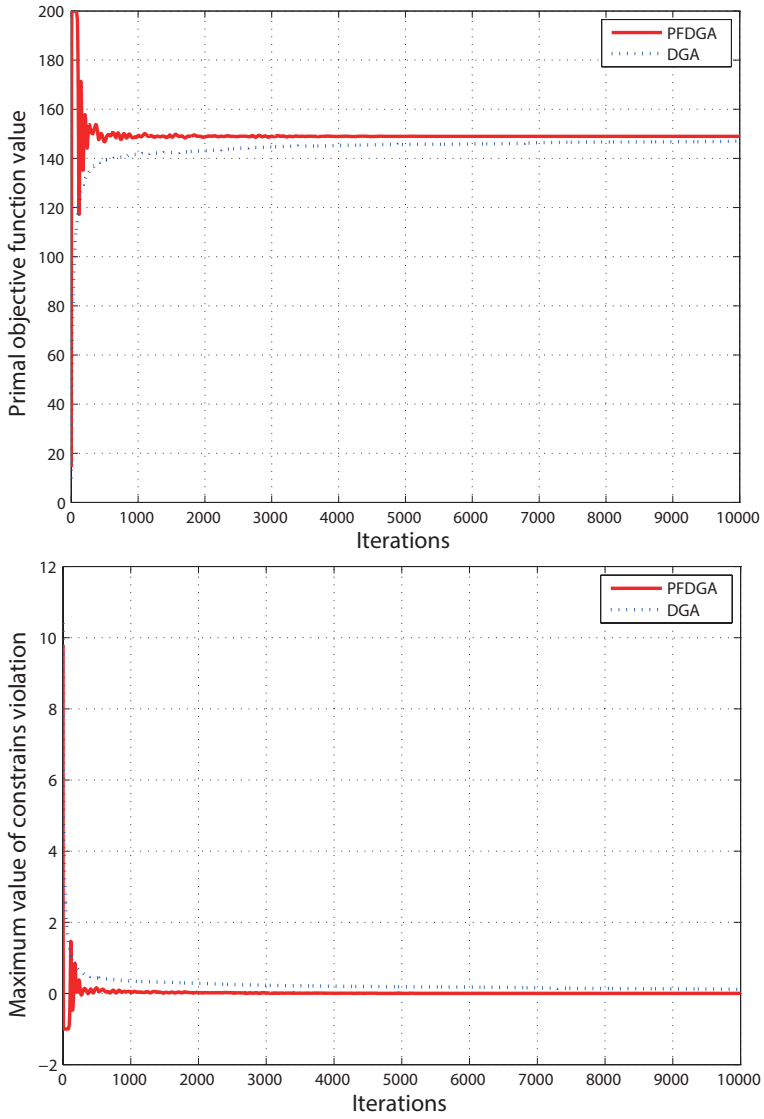

Figure 1: Objective value and maximum value of constraint violations versus number of iterations on a random network with $\mathcal{S} = 20$ and $\mathcal{L} = 50$.

than that of Alg. DGA. This is because Alg. FDGA is based on an accelerated gradient method without any stepsize tuning, while Alg. DGA is only based on (sub)gradient method.

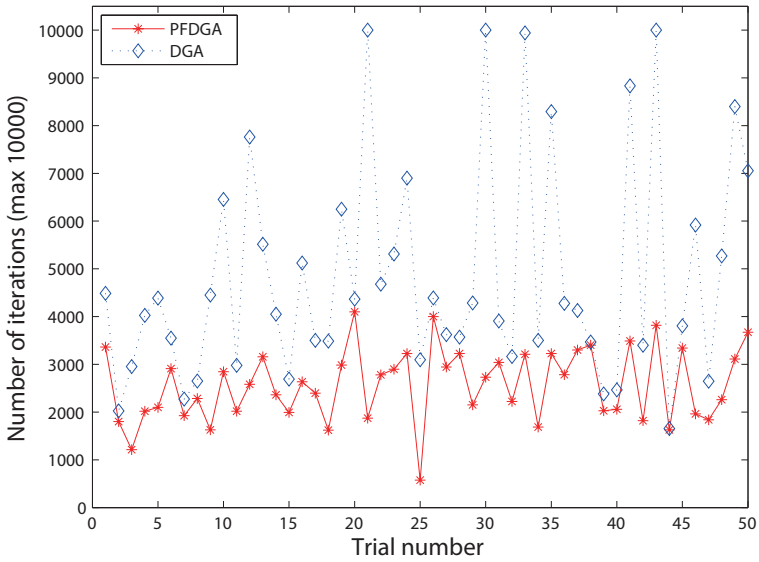

Figure 2: Number of iterations for both algorithms implemented over 50 randomly generated networks with $\mathcal{L} \in [20, 50]$ and $\mathcal{S} \in [10, 20]$.

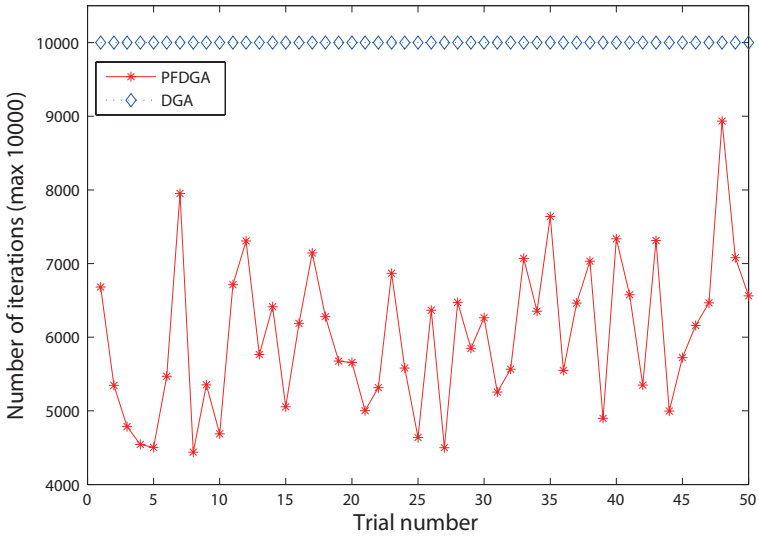

Figure 3: Number of iterations for both algorithms implemented over 50 randomly generated networks, each with 40 sources and 100 links.

We next test the problem NUM for larger sizes using our algorithm and report the average number of iterations and CPU time by solving 10 random NUM problems with different sizes. The results of Table 1 shows that Alg. FDGA has the potential to solve

the large-scale problem NUM. It can be observed from Table 1 that the average number of iterations and CPU time is increasing as the size of the problem NUM become larger.

Table 1: Average number of iterations and CPU time (s) with different sizes

	$(\mathcal{S},\mathcal{L})$	(100,200)	(200,400)	(300,600)	(400,800)	(500,1000)
Alg. FDGA:	*Iter*	7310	9157	10981	12701	14896
	CPU	0.9011	1.1458	1.2701	2.2791	3.5400

6 Conclusions

Based on a smoothing technique and a fast gradient algorithm, we proposed a simple dual fast dual gradient algorithm for solving separable convex optimization problems with linear coupled constraints. Our proposed algorithm can achieves a very fast convergence. The iteration complexity is established. The theoretical results established in the paper are validated by numerical experiments.

References

[1] B. Ling, N. Tian, C. Ho, W. Siu, K. Teo and Q. Dai, Maximally decimated paraunitary linear phase FIR filter bank design via iterative SVD approach, *IEEE Trans. Signal Proces.* 63 (2015) 466–481.

[2] B. Ling, C. Ho, W. Siu and Q. Dai, Best linear near unbiased estimation for nonlinear signal models via semi-infinite programming approach, *Comput. Stat. Data Anal.* 88 (2015) 111–118.

[3] B. Gao, W. Woo and B. Ling, Machine learning source separation using maximum a posteriori nonnegative matrix factorization, *IEEE T. Cybern.* 44 (2014) 1169–1179.

[4] A. Beck, A. Nedic, A. Ozdaglar and M. Teboulle, An O(1/k) Gradient Method for Network Resource Allocation Problems, *IEEE Trans. Control Net. Syst.* 1 (2014) 64–73.

[5] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, New York, 1989.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends@ in Machine Learning* 3 (2011) 1–122.

[7] R.I. Bot and C. Hendrich, A double smoothing technique for solving unconstrained nondifferentiable convex optimization problems. *Comput. Optim. Appl.* 54 (2013) 239–262.

[8] A.J. Connejo, R. Mínguez, E. Castillo and E. R. García-Bertrand, *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications.* Springer, Berlin, 2006.

[9] G. Chen and M. Teboulle, A proximal-based decomposition method for convex minimization problems. *Math. Program.* 64 (1994) 81–101.

[10] O. Devolder, F. Glineur and Y. Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization, *SIAM J. Optim.* 22 (2012) 702–727.

[11] Q. T. Dinh, I. Necoara, C. Savorgnan and M. Diehl, An inexact perturbed path-following method for lagrangian decomposition in large-scale separable convex optimization. *SIAM J. Optim.* 23(2013) 95-125.

[12] R. Ebrahimian and R. Baldick, State estimation distributed processing for power systems, *IEEE Trans. Power Syst.* 15 (2000) 1240–1246.

[13] D. Goldfarb and S. Ma, Fast multiple splitting algorithms for convex optimization. *SIAM J. Optim.* 22 (2012) 533–556.

[14] C. Li, W. Yu and T. Huang, Impulsive synchronization schemes of stochastic complex networks with switching topology: Average time approach. *Neural Networks* 54 (2014) 85–94.

[15] C. Li, C. Li, X. Liao and T. Huang, Impulsive effects on stability of high-order BAM neural networks with time delays. *Neurocomputing* 74 (2011) 1541–1550.

[16] C. Li, C. Li and T. Huang, Exponential stability of impulsive high-order Hopfield-type neural networks with delays and reaction-diffusion. *Int. J. Comput. Math.* 88 (2011) 3150–3162.

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

[18] J. Li, C. Li, Z. Wu and J. Huang, A feedback neural network for solving convex quadratic bi-level programming problems. *Neur. Comput. Appl.* 25 (2014) 603–611.

[19] J. Li, C. Wu, Z. Wu, Q. Long and X. Wang, Distributed proximal-gradient method for convex optimization with inequality constraints. *J. ANZIAM* 56 (2014) 160–178.

[20] J. Li, C. Wu, Z. Wu and Q. Long, Gradient-free method for nonsmooth distributed optimization. *J. Global Optim.* 61 (2015) 325–340.

[21] J. Li, C. Wu, Z. Wu, Q. Long, X. Wang, An Inexact Dual Fast Gradient-Projection Method for Separable Convex Optimization with Linear Coupled Constraints. *J. Optim. Theory Appl.* (2015) DOI 10.1007/s10957-015-0757-1.

[22] I. Necoara and J. Suykens, Application of a smoothing technique to decomposition in convex optimization. *IEEE Trans. Autom. Control* 53 (2008) 2674–2679.

[23] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, Boston, 2004.

[24] Y. Nesterov, Smooth minimization of non-smooth functions. *Math. Program.* 103 (2005) 127–152.

[25] A. Nedic and A. Ozdaglar, Subgradient methods for saddle-point problems, J. *Optim. Theory Appl.* 142 (2009) 205–228.

[26] L.B. Oliveira and E. Camponogara, Multi-agent model predictive control of signaling split in urban traffic networks, *Transportation Research Part C: Emerging Technologies* 18 (2010) 120-139.

[27] C.H. Rosa, and A. Ruszczyński, On augmented Lagrangian decomposition methods for multistage stochastic programs. *Ann. Oper. Res.* 64 (1996) 289–309.

[28] M. Tao, Some parallel splitting methods for separable convex programming with the O(1/t) convergence rate. *Pac. J. Optim.* 10 (2014) 201–213.

[29] Y. You, X. Fu and B. He, Lagrangian-PPA based contraction methods for linearly constrained convex optimization. *Pac. J. Optim.* 10 (2014) 359–384.

[30] G. Zhao, A Lagrangian dual method with self-concordant barriers for multistage stochastic convex programming. *Math. Program.* 102 (2005) 1–24.

Manuscript received 25 August 2014
revised 7 February 2015
accepted for publication 27 February 2015

Jueyou Li
School of Mathematical Sciences, Chongqing Normal University, Chongqing, 400047, China
E-mail address: lijueyou@163.com

Zhiyou Wu
School of Mathematical Sciences, Chongqing Normal University, Chongqing, 400047, China
E-mail address: zywu@cqnu.edu.cn

Changzhi Wu
Australasian Joint Research Centre for Building Information Modelling
School of Built Environment, Curtin University, Perth, WA 6845, Australia
E-mail address: c.wu@exchange.curtin.edu.au

Qiang Long
School of Science, Southwest University of Science and Technology
Mianyang, Sichuan, 621010, China
E-mail address: qianglong1985@qq.com

Xiangyu Wang
Australasian Joint Research Centre for Building Information Modelling
School of Built Environment, Curtin University, Perth, WA 6845, Australia
E-mail address: xiangyu.wang@.curtin.edu.au

Jae-Myung Lee
Department of Naval Architecture and Ocean Engineering
Pusan National University, Busan, Korea
Director, Integrative Graduate Program of Ship and
Offshore Plant Technology for Ocean Energy Resource, BK21Plus
Director, Cryogenic Material Research Institute
Pusan National University
E-mail address: jaemlee@pusan.ac.kr

Kwang-Hyo Jung
Department of Naval Architecture and Ocean Engineering
Pusan National University, Busan, Korea
E-mail address: kjung@pusan.ac.kr