IEEE TRANSACTIONS ON AUTOMATIC CONTROL, 63(4), PP. 1105–1112, APRIL, 2018

On the Convergence Time of Dual Subgradient Methods for Strongly Convex Programs

Hao Yu and Michael J. Neely
University of Southern California

***Abstract*—This paper studies the convergence time of dual gradient methods for general (possibly non-differentiable) strongly convex programs. For general convex programs, the convergence time of dual subgradient/gradient methods with simple running averages (running averages started from iteration 0) is known to be $O(\frac{1}{\epsilon^2})$. This paper shows that the convergence time for general strongly convex programs is $O(\frac{1}{\epsilon})$. This paper also considers a variation of the average scheme, called the sliding running averages, and shows that if the dual function of the strongly convex program is locally quadratic then the convergence time of the dual gradient method with sliding running averages is $O(\log(\frac{1}{\epsilon}))$. The convergence time analysis is further verified by numerical experiments.**

I. Introduction

Consider the following strongly convex program:

$$\min \quad f(\mathbf{x}) \tag{1}$$
$$\text{s.t.} \quad g_k(\mathbf{x}) \leq 0, \forall k \in \{1,2,\ldots,m\} \tag{2}$$
$$\mathbf{x} \in \mathcal{X} \tag{3}$$

where set $\mathcal{X} \subseteq R^n$ is closed and convex; function $f(\mathbf{x})$ is continuous and strongly convex on $\mathcal{X}$ (strong convexity is defined in Section II-A); functions $g_k(\mathbf{x}), \forall k \in \{1,2,\ldots,m\}$ are Lipschitz continuous and convex on $\mathcal{X}$. Note that the functions $f(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ are not necessarily differentiable. It is assumed throughout that problem (1)-(3) has an optimal solution. Strong convexity of f implies the optimum is unique.

Convex program (1)-(3) arises often in control applications such as model predictive control (MPC) [2], decentralized multi-agent control [3], and network flow control [4], [5]. More specifically, the model predictive control problem is to solve problem (1)-(3) where $f(\mathbf{x})$ is a quadratic function and each $g_k(\mathbf{x})$ is a linear function [2]. In decentralized multi-agent control [3], our goal is to develop distributive algorithms to solve problem (1)-(3) where $f(\mathbf{x})$ is the sum utility of individual agents and constraints $g_k(\mathbf{x}) \leq 0$ capture the communication or resource allocation constraints among individual agents. The network flow control and the transmission control protocol (TCP) in computer networks can be interpreted as the dual subgradient algorithm for solving a problem of the form (1)-(3) [4], [5]. In particular, Section V-B shows that the dual subgradient method based *online flow control* rapidly converges to optimality when utilities are strongly convex.

Hao Yu and Michael J. Neely are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA.

This work was presented in part at IEEE Conference on Decision and Control (CDC), Osaka, Japan, December, 2015 [1]. This work is supported in part by the NSF grant CCF-0747525.

A. The ϵ-Approximate Solution

Definition 1. *Let $\mathbf{x}^*$ be the optimal solution to problem (1)-(3). For any $\epsilon > 0$, a point $\mathbf{x}^\epsilon \in \mathcal{X}$ is said to be an ϵ-approximate solution*[1] *if $f(\mathbf{x}^\epsilon) \leq f(\mathbf{x}^*) + \epsilon$ and $g_k(\mathbf{x}^\epsilon) \leq \epsilon, \forall k \in \{1,\ldots,m\}$.*

Definition 2. *Let $\mathbf{x}(t), t \in \{1,2,\ldots\}$ be the solution sequence yielded by an iterative algorithm. The convergence time (to an ϵ-approximate solution) is the number of iterations required to achieve an ϵ-approximate solution. An algorithm is said to have convergence time $O(h(\epsilon))$ if $\{\mathbf{x}(t), t \geq O(h(\epsilon))\}$ is a sequence of ϵ-approximate solutions for some function $h(\epsilon)$.*

Note if $\mathbf{x}(t)$ satisfies $f(\mathbf{x}(t)) \leq f(\mathbf{x}^*) + \frac{1}{t}$ and $g_k(\mathbf{x}(t)) \leq \frac{1}{t}, \forall k \in \{1,\ldots,m\}$ for all $t \geq 1$, then error decays with time like $O(\frac{1}{t})$ and the convergence time is $O(\frac{1}{\epsilon})$.

B. The Dual Subgradient/Gradient Method

The dual subgradient method is a conventional method to solve (1)-(3) [6], [7]. It is an iterative algorithm that, every iteration, removes the inequality constraints (2) and chooses primal variables to minimize a function over the set $\mathcal{X}$. This can be decomposed into parallel smaller problems if the objective and constraint functions are separable. The dual subgradient method can be interpreted as a subgradient/gradient method applied to the Lagrangian dual function of convex program (1)-(3) and allows for many different step size rules [7]. This paper focuses on a constant step size due to its simplicity for practical implementations. Note that by Danskin's theorem (Proposition B.25(a) in [7]), the Lagrangian dual function of a strongly convex program is differentiable, thus the dual subgradient method for strongly convex program (1)-(3) is in fact a dual gradient method. The constant step size dual subgradient/gradient method solves problem (1)-(3) as follows:

Algorithm 1. *[The Dual Subgradient/Gradient Method] Let $c > 0$ be a constant step size. Let $\boldsymbol{\lambda}(0) \geq \mathbf{0}$, be a given constant vector. At each iteration t, update $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t+1)$ as follows:*

- $\mathbf{x}(t) = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} \left[f(\mathbf{x}) + \sum_{k=1}^m \lambda_k(t) g_k(\mathbf{x})\right]$.
- $\lambda_k(t+1) = \max\left\{\lambda_k(t) + c g_k(\mathbf{x}(t)), 0\right\}, \forall k$.

[1] If there exists $\mathbf{z} \in \mathcal{X}$ such that $g_k(\mathbf{z}) \leq -\delta, \forall k \in \{1,\ldots,m\}$ for some $\delta > 0$, one can convert an ϵ-approximate point $\mathbf{x}^\epsilon$ to another point $\mathbf{x} = \theta\mathbf{x}^\epsilon + (1-\theta)\mathbf{z}$, for $\theta = \frac{\delta}{\epsilon+\delta}$, which satisfies all desired constraints and has objective value within $O(\epsilon)$ of optimality.

Rather than using $\mathbf{x}(t)$ from Algorithm 1 as the primal solutions, the following running average schemes are considered:

1) **Simple Running Averages:** Use $\overline{\mathbf{x}}(t) = \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbf{x}(\tau)$ as the solution at each iteration $t \in \{1, 2, \ldots\}$.
2) **Sliding Running Averages:** Use $\widetilde{\mathbf{x}}(t) = \mathbf{x}(0)$ and

$$\widetilde{\mathbf{x}}(t) = \begin{cases} \frac{2}{t}\sum_{\tau=\frac{t}{2}}^{t-1}\mathbf{x}(\tau) & \text{if } t \text{ is even} \\ \widetilde{\mathbf{x}}(t-1) & \text{if } t \text{ is odd} \end{cases}$$

as the solution at each iteration $t \in \{1, 2, \ldots\}$.

The simple running average sequence $\overline{\mathbf{x}}(t)$ is also called the ergodic sequence in [8]. The idea of using the running average $\overline{\mathbf{x}}(t)$ as the solutions, rather than the original primal variables $\mathbf{x}(t)$, dates back to Shor [9] and has been further developed in [10] and [8]. The constant step size dual subgradient algorithm with simple running averages is also a special case of the drift-plus-penalty algorithm, which was originally developed to solve more general stochastic optimization [11] and used for deterministic convex programs in [12]. (See Section I.C in [1] for more discussions.) This paper proposes a new running average scheme, called sliding running averages. This paper shows that the sliding running averages can have a better convergence time when the dual function of the convex program satisfies additional assumptions.

C. Related Work

A lot of literature focuses on the convergence time of dual subgradient methods to an ϵ-approximate solution. For general convex programs in the form of (1)-(3), where the objective function $f(\mathbf{x})$ is convex but not necessarily strongly convex, the convergence time of the drift-plus-penalty algorithm is shown to be $O(\frac{1}{\epsilon^2})$ in [12], [13]. A similar $O(\frac{1}{\epsilon^2})$ convergence time of the dual subgradient algorithm with the averaged primal sequence is shown in [14]. A recent work [15] shows that the convergence time of the drift-plus-penalty algorithm is $O(\frac{1}{\epsilon})$ if the dual function is locally polyhedral and the convergence time is $O(\frac{1}{\epsilon^{3/2}})$ if the dual function is locally quadratic. For a special class of strongly convex programs in the form of (1)-(3), where $f(\mathbf{x})$ is second-order differentiable and strongly convex and $g_k(\mathbf{x}), \forall k \in \{1, 2, \ldots, m\}$ are second-order differentiable and have bounded Jacobians, the convergence time of the dual subgradient algorithm is shown to be $O(\frac{1}{\epsilon})$ in [2].

Note that convex program (1)-(3) with second order differentiable $f(\mathbf{x})$ and $g_k(\mathbf{x})$ in general can be solved via interior point methods with linear convergence time. However, to achieve fast convergence in practice, the barrier parameters must be scaled carefully and the computation complexity associated with each iteration is high. In contrast, the dual subgradient algorithm is a Lagrangian dual method and can yield distributed implementations with low computation complexity when the objective and constraint functions are separable.

This paper considers a class of strongly convex programs that is more general than those treated in [2].[2] Besides the strong convexity of $f(\mathbf{x})$, we only require the constraint functions $g_k(\mathbf{x})$ to be Lipschitz continuous. The functions $f(\mathbf{x})$ and $g_k(\mathbf{x})$ can even be non-differentiable. Thus, this paper can deal with non-smooth optimization. For example, the l_1 norm $\|\mathbf{x}\|_1$ is non-differentiable and often appears as part of the objective or constraint functions in machine learning, compressed sensing and image processing applications. This paper shows that the convergence time of the dual subgradient method with simple running averages for general strongly convex programs is $O(\frac{1}{\epsilon})$ and the convergence time can be improved to $O(\log(\frac{1}{\epsilon}))$ by using sliding running averages when the dual function is locally quadratic.

A closely related recent work is [16] that considers strongly convex programs with strongly convex and second order differentiable objective functions $f(\mathbf{x})$ and conic constraints in the form of $\mathbf{G}\mathbf{x} + h \in \mathcal{K}$, where $\mathcal{K}$ is a proper cone. The authors in [16] show that a hybrid algorithm using both dual subgradient and dual fast gradient methods can have convergence time $O(\frac{1}{\epsilon^{2/3}})$; and the dual subgradient method can have convergence time $O(\log(\frac{1}{\epsilon}))$ if the strongly convex program satisfies an error bound property. Results in the current paper are developed independently and consider general nonlinear convex constraint functions; and show that the dual subgradient/gradient method with a different averaging scheme has an $O(\log(\frac{1}{\epsilon}))$ convergence time when the dual function is locally quadratic. Another parallel work [17] considers strongly convex programs with strongly convex and smooth objective functions $f(\mathbf{x})$ and general constraint functions $\mathbf{g}(\mathbf{x})$ with bounded Jacobians. The authors in [17] show that the dual subgradient/gradient method with simple running averages has $O(\frac{1}{\epsilon})$ convergence.

This paper and independent parallel works [16], [17] obtain similar convergence times of the dual subgradient/gradient method with different averaging schemes for strongly convex programs under slightly different assumptions. However, the proof technique in this paper is fundamentally different from that used in [16] and [17]. Works [16], [17] and other previous works, e.g., [2], follow the classical optimization analysis approach based on the descent lemma, while this paper is based on the drift-plus-penalty analysis that was originally developed for stochastic optimization in dynamic queuing systems [18], [11]. Using the drift-plus-penalty technique, we further propose a new Lagrangian dual type algorithm with $O(\frac{1}{\epsilon})$ convergence for general convex programs (possibly without strong convexity) in a following work [19].

II. Preliminaries and Basic Analysis

A. Preliminaries and Assumptions

Definition 3 (Lipschitz Continuity). *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set. Function $h : \mathcal{X} \to \mathbb{R}^m$ is said to be Lipschitz continuous on $\mathcal{X}$ with modulus L if there exists $L > 0$ such that $\|h(\mathbf{y}) - h(\mathbf{x})\| \leq L\|\mathbf{y} - \mathbf{x}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Note that $\|\cdot\|$ in the above definition can be general norms. However, throughout this paper, we use $\|\cdot\|$ to denote the vector Euclidean norm.

Definition 4 (Strongly Convex Functions). *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set. Function h is said to be strongly convex on $\mathcal{X}$*

[2]Note that bounded Jacobians imply Lipschitz continuity. Work [2] also considers the effect of inaccurate solutions for the primal updates. The analysis in this paper can also deal with inaccurate updates. In this case, there will be an error term δ on the right of (6).

with modulus α if there exists a constant $\alpha > 0$ such that $h(\mathbf{x}) - \frac{1}{2}\alpha\|\mathbf{x}\|^2$ is convex on $\mathcal{X}$.

Lemma 1. *[See [20] or Corollary 1 in [19]] Let $h(\mathbf{x})$ be strongly convex on convex set $\mathcal{X}$ with modulus α. If $\mathbf{x}^{opt}$ is a global minimum, $h(\mathbf{x}^{opt}) \leq h(\mathbf{y}) - \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}^{opt}\|^2, \forall \mathbf{y} \in \mathcal{X}$.*

Denote the stacked vector of multiple functions $g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})$ as

$$\mathbf{g}(\mathbf{x}) = \big[g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})\big]^T.$$

Assumption 1. *In convex program* (1)-(3)*, function $f(\mathbf{x})$ is strongly convex on $\mathcal{X}$ with modulus α; and function $\mathbf{g}(\mathbf{x})$ is Lipschitz continuous on $\mathcal{X}$ with modulus β.*

Assumption 2. *There exists a Lagrange multiplier vector $\boldsymbol{\lambda}^* = [\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*]^T \geq \mathbf{0}$ such that*

$$q(\boldsymbol{\lambda}^*) = \min_{\mathbf{x}\in\mathcal{X}} \{f(\mathbf{x}) : g_k(\mathbf{x}) \leq 0, \forall k \in \{1, 2, \ldots, m\}\},$$

where $q(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \sum_{k=1}^m \lambda_k g_k(\mathbf{x})\}$ is the Lagrangian dual function of problem (1)-(3)*.*

B. Properties of the Lyapunov drift

Denote $\boldsymbol{\lambda}(t) = \big[\lambda_1(t), \ldots, \lambda_m(t)\big]^T$. Define Lyapunov function $L(t) = \frac{1}{2}\|\boldsymbol{\lambda}(t)\|^2$ and drift $\Delta(t) = L(t+1) - L(t)$.

Lemma 2. *At each iteration t in Algorithm 1,*

$$\frac{1}{c}\Delta(t) = \boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}(t)) - \frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 \quad (4)$$

Proof: The update equations $\lambda_k(t+1) = \max\{\lambda_k(t) + cg_k(\mathbf{x}(t)), 0\}, \forall k \in \{1, 2, \ldots, m\}$ can be rewritten as

$$\lambda_k(t+1) = \lambda_k(t) + c\tilde{g}_k(\mathbf{x}(t)), \forall k \in \{1, 2, \ldots, m\}, \quad (5)$$

where $\tilde{g}_k(\mathbf{x}(t)) = \begin{cases} g_k(\mathbf{x}(t)), & \text{if } \lambda_k(t) + cg_k(\mathbf{x}(t)) \geq 0 \\ -\frac{1}{c}\lambda_k(t), & \text{else} \end{cases}$, $\forall k \in \{1, 2, \ldots, m\}$. Fix $k \in \{1, 2, \ldots, m\}$. Squaring both sides of (5) and dividing by factor 2 yields:

$$\begin{aligned}
&\frac{1}{2}[\lambda_k(t+1)]^2 \\
=&\frac{1}{2}[\lambda_k(t)]^2 + \frac{c^2}{2}[\tilde{g}_k(\mathbf{x}(t))]^2 + c\lambda_k(t)\tilde{g}_k(\mathbf{x}(t)) \\
=&\frac{1}{2}[\lambda_k(t)]^2 + \frac{c^2}{2}[\tilde{g}_k(\mathbf{x}(t))]^2 + c\lambda_k(t)g_k(\mathbf{x}(t)) \\
&+ c\lambda_k(t)[\tilde{g}_k(\mathbf{x}(t)) - g_k(\mathbf{x}(t))] \\
\overset{(a)}{=}&\frac{1}{2}[\lambda_k(t)]^2 + \frac{c^2}{2}[\tilde{g}_k(\mathbf{x}(t))]^2 + c\lambda_k(t)g_k(\mathbf{x}(t)) \\
&- c^2\tilde{g}_k(\mathbf{x}(t))[\tilde{g}_k(\mathbf{x}(t)) - g_k(\mathbf{x}(t))] \\
=&\frac{1}{2}[\lambda_k(t)]^2 - \frac{c^2}{2}[\tilde{g}_k(\mathbf{x}(t))]^2 + c[\lambda_k(t) + c\tilde{g}_k(\mathbf{x}(t))]g_k(\mathbf{x}(t)) \\
\overset{(b)}{=}&\frac{1}{2}[\lambda_k(t)]^2 - \frac{1}{2}[\lambda_k(t+1) - \lambda_k(t)]^2 + c\lambda_k(t+1)g_k(\mathbf{x}(t))
\end{aligned}$$

where (a) follows from $\lambda_k(t)[\tilde{g}_k(\mathbf{x}(t)) - g_k(\mathbf{x}(t))] = -c\tilde{g}_k(\mathbf{x}(t))[\tilde{g}_k(\mathbf{x}(t)) - g_k(\mathbf{x}(t))]$, which can be shown by separately considering cases $\tilde{g}_k(\mathbf{x}(t)) = g_k(\mathbf{x}(t))$ and $\tilde{g}_k(\mathbf{x}(t)) \neq g_k(\mathbf{x}(t))$; and (b) follows from the fact that $\lambda_k(t+1) = \lambda_k(t) + c\tilde{g}_k(\mathbf{x}(t))$. Summing over $k \in \{1, 2, \ldots, m\}$ and dividing both sides by factor c yields the result. ■

III. Convergence Time Analysis

This section analyzes the convergence time of $\overline{\mathbf{x}}(t)$ for strongly convex program (1)-(3).

A. Objective Value Violations

Lemma 3. *Let $\mathbf{x}^*$ be the optimal solution to* (1)-(3)*. Assume $c \leq \alpha/\beta^2$. At each iteration t in Algorithm 1, we have*

$$\frac{1}{c}\Delta(t) + f(\mathbf{x}(t)) \leq f(\mathbf{x}^*) \quad , \forall t \geq 0. \quad (6)$$

Proof: Fix $t \geq 0$. Since $f(\mathbf{x})$ is strongly convex with modulus α and for all $k \in \{1, \ldots, m\}$ functions $g_k(\mathbf{x})$ are convex and scalars $\lambda_k(t)$ are non-negative, the function $f(\mathbf{x}) + \sum_{k=1}^m \lambda_k(t)g_k(\mathbf{x})$ is also strongly convex with modulus α. Note that $\mathbf{x}(t) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} [f(\mathbf{x}) + \sum_{k=1}^m \lambda_k(t)g_k(\mathbf{x})]$. By Lemma 1 with $\mathbf{x}^{opt} = \mathbf{x}(t)$ and $\mathbf{y} = \mathbf{x}^*$, we have

$$\begin{aligned}
&f(\mathbf{x}(t)) + \textstyle\sum_{k=1}^m \lambda_k(t)g_k(\mathbf{x}(t)) \\
\leq&\big[f(\mathbf{x}^*) + \textstyle\sum_{k=1}^m \lambda_k(t)g_k(\mathbf{x}^*)\big] - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2
\end{aligned}$$

Adding this to equation (4) yields $\frac{1}{c}\Delta(t) + f(\mathbf{x}(t)) \leq f(\mathbf{x}^*) + B(t)$, where

$$\begin{aligned}
B(t) =& -\frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
&+ \boldsymbol{\lambda}^T(t)[\mathbf{g}(\mathbf{x}^*) - \mathbf{g}(\mathbf{x}(t))] + \boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}(t)) \quad (7)
\end{aligned}$$

It remains to show that $B(t) \leq 0$. Since $\mathbf{x}^*$ is the optimal solution to problem (1)-(3), we have $g_k(\mathbf{x}^*) \leq 0$ for all k. Note that $\lambda_k(t+1) \geq 0$ for all k. Thus, $-\boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}^*) \geq 0$. Adding the nonnegative quantity $-\boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}^*)$ to the right-hand-side of (7) gives:

$$\begin{aligned}
B(t) \leq& -\frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
&+ \boldsymbol{\lambda}^T(t)[\mathbf{g}(\mathbf{x}^*) - \mathbf{g}(\mathbf{x}(t))] + \boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}(t)) \\
&- \boldsymbol{\lambda}^T(t+1)\mathbf{g}(\mathbf{x}^*) \\
=& -\frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
&+ [\boldsymbol{\lambda}^T(t) - \boldsymbol{\lambda}^T(t+1)][\mathbf{g}(\mathbf{x}^*) - \mathbf{g}(\mathbf{x}(t))] \\
\overset{(a)}{\leq}& -\frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
&+ \|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(t+1)\|\|\mathbf{g}(\mathbf{x}(t)) - \mathbf{g}(\mathbf{x}^*)\| \\
\overset{(b)}{\leq}& -\frac{1}{2c}\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|^2 - \frac{\alpha}{2}\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
&+ \beta\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(t+1)\|\|\mathbf{x}(t) - \mathbf{x}^*\| \\
=& -\frac{1}{2c}\big(\|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\| - c\beta\|\mathbf{x}(t) - \mathbf{x}^*\|\big)^2 \\
&- \frac{1}{2}(\alpha - c\beta^2)\|\mathbf{x}(t) - \mathbf{x}^*\|^2 \\
\overset{(c)}{\leq}& 0
\end{aligned}$$

where (a) follows from the Cauchy-Schwarz inequality; (b) follows from Assumption 1; and (c) follows from $c \leq \frac{\alpha}{\beta^2}$. ■

Theorem 1 (Objective Value Violations)**.** *Let $\mathbf{x}^* \in \mathcal{X}$ be the optimal solution to problem* (1)-(3)*. If $c \leq \frac{\alpha}{\beta^2}$ in Algorithm 1, then $f(\overline{\mathbf{x}}(t)) \leq f(\mathbf{x}^*) + \frac{\|\boldsymbol{\lambda}(0)\|^2}{2ct}, \forall t \geq 1$.*

Proof: Fix $t \geq 1$. Summing (6) over $\tau \in \{0, 1, \ldots, t-1\}$ yields $\frac{1}{c}\sum_{\tau=0}^{t-1}\Delta(\tau) + \sum_{\tau=0}^{t-1} f(\mathbf{x}(\tau)) \leq tf(\mathbf{x}^*)$. Dividing by factor t and rearranging terms yields

$$\begin{aligned}\tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1} f(\mathbf{x}(\tau)) &\leq f(\mathbf{x}^*) + \tfrac{L(0)-L(t)}{ct} \\ &= f(\mathbf{x}^*) + \tfrac{\|\boldsymbol{\lambda}(0)\|^2 - \|\boldsymbol{\lambda}(t)\|^2}{2ct}.\end{aligned}$$

Finally, $f(\overline{\mathbf{x}}(t)) \leq \frac{1}{t}\sum_{\tau=0}^{t-1} f(\mathbf{x}(\tau))$ by Jensen's inequality. ■

B. Constraint Violations

Lemma 4. *For any $t_2 > t_1 \geq 0$,*

$$\lambda_k(t_2) \geq \lambda_k(t_1) + c\sum_{\tau=t_1}^{t_2-1} g_k(\mathbf{x}(\tau)) \quad , \forall k \in \{1, 2, \ldots, m\}$$

In particular, $\lambda_k(t) \geq \lambda_k(0) + c\sum_{\tau=0}^{t-1} g_k(\mathbf{x}(\tau))$ for all $t \geq 1$ and $k \in \{1, 2, \ldots, m\}$.

Proof: Fix $k \in \{1, 2, \ldots, m\}$. Note that $\lambda_k(t_1+1) = \max\{\lambda_k(t_1) + cg_k(\mathbf{x}(t_1)), 0\} \geq \lambda_k(t_1) + cg_k(\mathbf{x}(t_1))$. By induction, this lemma follows. ■

Lemma 5. *Let $\boldsymbol{\lambda}^* \geq \mathbf{0}$ be given in Assumption 2. If $c \leq \frac{\alpha}{\beta^2}$ in Algorithm 1, then $\boldsymbol{\lambda}(t)$ satisfies*

$$\|\boldsymbol{\lambda}(t)\| \leq \sqrt{\|\boldsymbol{\lambda}(0)\|^2 + \|\boldsymbol{\lambda}^*\|^2} + \|\boldsymbol{\lambda}^*\|, \forall t \geq 1. \tag{8}$$

Proof: Let $\mathbf{x}^*$ be the optimal solution to problem (1)-(3). Assumption 2 implies that $f(\mathbf{x}^*) = q(\boldsymbol{\lambda}^*) \leq f(\mathbf{x}(\tau)) + \sum_{k=1}^m \lambda_k^* g_k(\mathbf{x}(\tau)), \forall \tau \in \{0, 1, \ldots\}$, where the inequality follows from the definition of $q(\boldsymbol{\lambda})$. Thus, we have $f(\mathbf{x}^*) - f(\mathbf{x}(\tau)) \leq \sum_{k=1}^m \lambda_k^* g_k(\mathbf{x}(\tau)), \forall \tau \in \{0, 1, \ldots\}$. Summing over $\tau \in \{0, 1, \ldots, t-1\}$ yields

$$\begin{aligned}&tf(\mathbf{x}^*) - \sum_{\tau=0}^{t-1} f(\mathbf{x}(\tau)) \leq \sum_{\tau=0}^{t-1}\sum_{k=1}^{m} \lambda_k^* g_k(\mathbf{x}(\tau)) \\ =&\sum_{k=1}^{m} \lambda_k^* \Big[\sum_{\tau=0}^{t-1} g_k(\mathbf{x}(\tau))\Big] \overset{(a)}{\leq} \frac{1}{c}\sum_{k=1}^{m} \lambda_k^*[\lambda_k(t) - \lambda_k(0)] \\ \leq&\frac{1}{c}\sum_{k=1}^{m} \lambda_k^*\lambda_k(t) \overset{(b)}{\leq} \frac{1}{c}\|\boldsymbol{\lambda}^*\|\|\boldsymbol{\lambda}(t)\| \end{aligned} \tag{9}$$

where (a) follows from Lemma 4 and (b) follows from the Cauchy-Schwarz inequality. On the other hand, summing (6) in Lemma 3 over $\tau \in \{0, 1, \ldots, t-1\}$ yields

$$tf(\mathbf{x}^*) - \sum_{\tau=0}^{t-1} f(\mathbf{x}(\tau)) \geq \frac{L(t) - L(0)}{c} = \frac{\|\boldsymbol{\lambda}(t)\|^2 - \|\boldsymbol{\lambda}(0)\|^2}{2c} \tag{10}$$

Combining (9) and (10) yields

$$\begin{aligned}&\frac{\|\boldsymbol{\lambda}(t)\|^2 - \|\boldsymbol{\lambda}(0)\|^2}{2c} \leq \frac{1}{c}\|\boldsymbol{\lambda}^*\|\|\boldsymbol{\lambda}(t)\| \\ \Rightarrow\quad &\left(\|\boldsymbol{\lambda}(t)\| - \|\boldsymbol{\lambda}^*\|\right)^2 \leq \|\boldsymbol{\lambda}(0)\|^2 + \|\boldsymbol{\lambda}^*\|^2 \\ \Rightarrow\quad &\|\boldsymbol{\lambda}(t)\| \leq \sqrt{\|\boldsymbol{\lambda}(0)\|^2 + \|\boldsymbol{\lambda}^*\|^2} + \|\boldsymbol{\lambda}^*\|\end{aligned}$$

■

Theorem 2 (Constraint Violations)**.** *Let $\boldsymbol{\lambda}^* \geq \mathbf{0}$ be defined in Assumption 2. If $c \leq \frac{\alpha}{\beta^2}$ in Algorithm 1, then the constraint functions satisfy for all $t \geq 1$:*

$$g_k(\overline{\mathbf{x}}(t)) \leq \frac{\sqrt{\|\boldsymbol{\lambda}(0)\|^2 + \|\boldsymbol{\lambda}^*\|^2} + \|\boldsymbol{\lambda}^*\|}{ct}, \forall k \in \{1, \ldots, m\}$$

Proof: Fix $t \geq 1$ and $k \in \{1, 2, \ldots, m\}$. Recall that $\overline{\mathbf{x}}(t) = \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbf{x}(\tau)$. Thus, $g_k(\overline{\mathbf{x}}(t)) \overset{(a)}{\leq} \frac{1}{t}\sum_{\tau=0}^{t-1} g_k(\mathbf{x}(\tau)) \overset{(b)}{\leq} \frac{\lambda_k(t)-\lambda_k(0)}{ct} \leq \frac{\lambda_k(t)}{ct} \leq \frac{\|\boldsymbol{\lambda}(t)\|}{ct} \overset{(c)}{\leq} \frac{\sqrt{\|\boldsymbol{\lambda}(0)\|^2 + \|\boldsymbol{\lambda}^*\|^2} + \|\boldsymbol{\lambda}^*\|}{ct}$, where (a) follows from the convexity of $g_k(\mathbf{x})$; (b) follows from Lemma 4; and (c) follows from Lemma 5. ■

Theorems 1-2 show that using the simple running average sequence $\overline{\mathbf{x}}(t)$ ensures the objective value and constraint error decay like $O(1/t)$. A lower bound of $f(\overline{\mathbf{x}}(t)) \geq f(\mathbf{x}^*) - O(\frac{1}{t})$ easily follows from strong duality and Theorem 2. See full version [20] for more discussions.

IV. Extensions

This section shows that the convergence time of sliding running averages $\widetilde{\mathbf{x}}(t)$ is $O(\log(\frac{1}{\epsilon}))$ when the dual function of problem (1)-(3) satisfies additional assumptions.

A. Smooth Dual Functions

Definition 5 (Smooth Functions)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^n$ and function $h(\mathbf{x})$ be continuously differentiable on $\mathcal{X}$. Function $h(\mathbf{x})$ is said to be smooth on $\mathcal{X}$ with modulus L if $\nabla_{\mathbf{x}} h(\mathbf{x})$ is Lipschitz continuous on $\mathcal{X}$ with modulus L.*

Define $q(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x})\}$ as the *dual function* of problem (1)-(3). Recall that $f(\mathbf{x})$ is strongly convex with modulus α by Assumption 1. For fixed $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, $f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x})$ is strongly convex with respect to $\mathbf{x} \in \mathcal{X}$ with modulus α. Define $\mathbf{x}(\boldsymbol{\lambda}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x})\}$. By Danskin's theorem (Proposition B.25 in [7]), $q(\boldsymbol{\lambda})$ is differentiable with gradient $\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}) = \mathbf{g}(\mathbf{x}(\boldsymbol{\lambda}))$.

Lemma 6 (Smooth Dual Functions)**.** *The dual function $q(\boldsymbol{\lambda})$ is smooth on $\mathbb{R}_+^m$ with modulus $\gamma = \frac{\beta^2}{\alpha}$.*

Proof: Fix $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}_+^m$. Let $\mathbf{x}(\boldsymbol{\lambda}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x})\}$ and $\mathbf{x}(\boldsymbol{\mu}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \boldsymbol{\mu}^T\mathbf{g}(\mathbf{x})\}$. By Lemma 1, we have $f(\mathbf{x}(\boldsymbol{\lambda})) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x}(\boldsymbol{\lambda})) \leq f(\mathbf{x}(\boldsymbol{\mu})) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x}(\boldsymbol{\mu})) - \frac{\alpha}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\|^2$ and $f(\mathbf{x}(\boldsymbol{\mu})) + \boldsymbol{\mu}^T\mathbf{g}(\mathbf{x}(\boldsymbol{\mu})) \leq f(\mathbf{x}(\boldsymbol{\lambda})) + \boldsymbol{\mu}^T\mathbf{g}(\mathbf{x}(\boldsymbol{\lambda})) - \frac{\alpha}{2}\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\|^2$. Summing the above two inequalities and simplifying gives

$$\begin{aligned}\alpha\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\|^2 &\leq [\boldsymbol{\mu} - \boldsymbol{\lambda}]^T[\mathbf{g}(\mathbf{x}(\boldsymbol{\lambda})) - \mathbf{g}(\mathbf{x}(\boldsymbol{\mu}))] \\ &\overset{(a)}{\leq} \|\boldsymbol{\mu} - \boldsymbol{\lambda}\|\|\mathbf{g}(\mathbf{x}(\boldsymbol{\lambda})) - \mathbf{g}(\mathbf{x}(\boldsymbol{\mu}))\| \\ &\overset{(b)}{\leq} \beta\|\boldsymbol{\mu} - \boldsymbol{\lambda}\|\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\|\end{aligned}$$

where (a) follows from the Cauchy-Schwarz inequality and (b) follows because $\mathbf{g}(\mathbf{x})$ is Lipschitz coninuous. This implies

$$\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\| \leq \frac{\beta}{\alpha}\|\boldsymbol{\lambda} - \boldsymbol{\mu}\| \tag{11}$$

Thus, we have $\|\nabla q(\boldsymbol{\lambda})-\nabla q(\boldsymbol{\mu})\| \overset{(a)}{=} \|\mathbf{g}(\mathbf{x}(\boldsymbol{\lambda}))-\mathbf{g}(\mathbf{x}(\boldsymbol{\mu}))\| \overset{(b)}{\leq} \beta\|\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{x}(\boldsymbol{\mu})\| \overset{(c)}{\leq} \frac{\beta^2}{\alpha}\|\boldsymbol{\lambda} - \boldsymbol{\mu}\|$ where (a) follows from $\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}) = \mathbf{g}(\mathbf{x}(\boldsymbol{\lambda}))$; (b) follows from the Lipschitz continuity of $\mathbf{g}(\mathbf{x})$; and (c) follows from (11).

Thus, $q(\boldsymbol{\lambda})$ is smooth on $\mathbb{R}^m_+$ with modulus $L = \frac{\beta^2}{\alpha}$. ■

Since $\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}(t)) = \mathbf{g}(\mathbf{x}(t))$, the dynamic of $\boldsymbol{\lambda}(t)$ can be interpreted as the projected gradient method with step size c to solve $\max_{\boldsymbol{\lambda}\in\mathbb{R}^m_+}\{q(\boldsymbol{\lambda})\}$ where $\mathbf{q}(\cdot)$ is a smooth function by Lemma 6. Thus, we have the next lemma.

Lemma 7. *Assume problem* (1)-(3) *satisfies Assumptions 1-2. If* $c \leq \frac{\alpha}{\beta^2}$*, then* $q(\boldsymbol{\lambda}^*) - q(\boldsymbol{\lambda}(t)) \leq \frac{1}{2ct}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2, \forall t \geq 1$.

Proof: Recall that a projected gradient algorithm with step size $c < \frac{1}{\gamma}$ maximizes a concave function with smooth modulus γ with the error decaying like $O(\frac{1}{t})$. Thus, this lemma follows. See [20] for the complete proof. ■

B. Problems with Locally Quadratic Dual Functions

In addition to Assumptions 1-2, this subsection further requires the next assumption.

Assumption 3 (Locally Quadratic Dual Functions)**.** *Let* $\boldsymbol{\lambda}^*$ *be a Lagrange multiplier of problem* (1)-(3) *defined in Assumption 2. There exists* $D_q > 0$ *and* $L_q > 0$*, where the subscript* q *denotes locally "quadratic", such that for all* $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda} \in \mathbb{R}^m_+ : \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\| \leq D_q\}$*, the dual function* $q(\boldsymbol{\lambda})$ *satisfies* $q(\boldsymbol{\lambda}^*) \geq q(\boldsymbol{\lambda}) + L_q\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\|^2$.

Lemma 8. *Suppose problem* (1)-(3) *satisfies Assumptions 1-3. Let* $q(\boldsymbol{\lambda}), \boldsymbol{\lambda}^*, D_q$ *and* L_q *be given in Assumption 3.*

1) *If* $\boldsymbol{\lambda} \in \mathbb{R}^m_+$ *and* $q(\boldsymbol{\lambda}^*) - q(\boldsymbol{\lambda}) \leq L_q D_q^2$*, then* $\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\| \leq D_q$.
2) *The* $\boldsymbol{\lambda}^*$ *defined in Assumption 2 is unique.*

Proof: See [20] for the proof. ■

Define

$$T_q = \frac{\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2}{2cL_qD_q^2}. \tag{12}$$

Lemma 9. *Assume problem* (1)-(3) *satisfies Assumptions 1-3. If* $c \leq \frac{\alpha}{\beta^2}$ *in Algorithm 1, then* $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq D_q$ *for all* $t \geq T_q$*, where* T_q *is defined in* (12)*.*

Proof: By Lemma 7 and Lemma 8, if $\frac{1}{2ct}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2 \leq L_qD_q^2$, then $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq D_q$. Note that $t \geq \frac{\|\boldsymbol{\lambda}(0)-\boldsymbol{\lambda}^*\|^2}{2cL_qD_q^2}$ implies that $\frac{1}{2ct}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2 \leq L_qD_q^2$. ■

Lemma 10. *Assume problem* (1)-(3) *satisfies Assumptions 1-3. If* $c \leq \frac{\alpha}{\beta^2}$ *in Algorithm 1, then*

1) $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq \frac{1}{\sqrt{t}}\frac{1}{\sqrt{2cL_q}}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|, \forall t \geq T_q$*, where* T_q *is defined in* (12)*.*
2) $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq \big(\sqrt{\frac{1}{1+2cL_q}}\big)^{t-T_q}\|\boldsymbol{\lambda}(T_q) - \boldsymbol{\lambda}^*\| \leq \big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}}$*,* $\forall t \geq T_q$ *, where* T_q *is defined in* (12)*.*

Proof:

1) By Lemma 7, $q(\boldsymbol{\lambda}^*) - q(\boldsymbol{\lambda}(t)) \leq \frac{1}{2ct}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2, \forall t \geq 1$. By Lemma 9 and Assumption 3, $q(\boldsymbol{\lambda}^*) - q(\boldsymbol{\lambda}(t)) \geq L_q\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\|^2, \forall t \geq T_q$. Thus, we have $L_q\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\|^2 \leq \frac{1}{2ct}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|^2, \forall t \geq T_q$, which implies that $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq \frac{1}{\sqrt{t}}\frac{1}{\sqrt{2cL_q}}\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\|, \forall t \geq T_q$.
2) By part (1), we know $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \leq D_q, \forall t \geq T_q$. The second part essentially follows from Theorem 12 in [21], which shows that the projected gradient method for set constrained smooth convex optimization converge geometrically if the objective function satisfies a quadratic growth condition. See [20] for the proof.

■

Corollary 1. *Assume problem* (1)-(3) *satisfies Assumptions 1-3. If* $c \leq \frac{\alpha}{\beta^2}$ *in Algorithm 1, then* $\|\boldsymbol{\lambda}(2t) - \boldsymbol{\lambda}(t)\| \leq 2\big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}}, \forall t \geq T_q$*, where* T_q *is defined in* (12)*.*

Proof:

$$\begin{aligned}
&\|\boldsymbol{\lambda}(2t) - \boldsymbol{\lambda}(t)\| \leq \|\boldsymbol{\lambda}(2t) - \boldsymbol{\lambda}^*\| + \|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*\| \\
&\overset{(a)}{\leq} \big(\frac{1}{\sqrt{1+2cL_q}}\big)^{2t} D_q(1+2cL_q)^{\frac{T_q}{2}} + \big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}} \\
&\overset{(b)}{\leq} 2\big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}},
\end{aligned}$$

where (a) follows from part (2) in Lemma 10; and (b) follows from $\frac{1}{\sqrt{1+2cLq}} < 1$. ■

Theorem 3. *Assume problem* (1)-(3) *satisfies Assumptions 1-3. Let* $\mathbf{x}^*$ *be the optimal solution and* $\boldsymbol{\lambda}^*$ *be defined in Assumption 3. If* $c \leq \frac{\alpha}{\beta^2}$ *in Algorithm 1, then* $f(\widetilde{\mathbf{x}}(2t)) \leq f(\mathbf{x}^*) + \frac{1}{t}\big(\frac{1}{\sqrt{1+2cL_q}}\big)^t\eta, \forall t \geq T_q$*, where* $\eta = \frac{2D_q^2(1+2cL_q)^{T_q} + 2D_q(1+2cL_q)^{\frac{T_q}{2}}(\sqrt{\|\boldsymbol{\lambda}(0)\|^2+\|\boldsymbol{\lambda}^*\|^2}+\|\boldsymbol{\lambda}^*\|)}{c}$ *and* T_q *is defined in* (12)*.*

Proof: Fix $t \geq T_q$. By Lemma 3, we have $\frac{1}{c}\Delta(\tau) + f(\mathbf{x}(\tau)) \leq f(\mathbf{x}^*)$ for all $\tau \in \{0, 1, \ldots\}$. Summing over $\tau \in \{t, t+1, \ldots, 2t-1\}$ yields $\frac{1}{c}\sum_{\tau=t}^{2t-1}\Delta(\tau) + \sum_{\tau=t}^{2t-1} f(\mathbf{x}(\tau)) \leq tf(\mathbf{x}^*)$. Dividing by factor t yields

$$\frac{1}{t}\sum_{\tau=t}^{2t-1} f(\mathbf{x}(\tau)) \leq f(\mathbf{x}^*) + \frac{L(t) - L(2t)}{ct} \tag{13}$$

Thus, we have

$$\begin{aligned}
f(\widetilde{\mathbf{x}}(2t)) &\overset{(a)}{\leq} \frac{1}{t}\sum_{\tau=t}^{2t-1} f(\mathbf{x}(\tau)) \overset{(b)}{\leq} f(\mathbf{x}^*) + \frac{L(t) - L(2t)}{ct} \\
&= f(\mathbf{x}^*) + \frac{\|\boldsymbol{\lambda}(t)\|^2 - \|\boldsymbol{\lambda}(2t)\|^2}{2ct} \\
&= f(\mathbf{x}^*) + \frac{\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(2t) + \boldsymbol{\lambda}(2t)\|^2 - \|\boldsymbol{\lambda}(2t)\|^2}{2ct} \\
&\overset{(c)}{\leq} f(\mathbf{x}^*) + \frac{\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(2t)\|^2 + 2\|\boldsymbol{\lambda}(2t)\|\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(2t)\|}{2ct} \\
&\overset{(d)}{\leq} f(\mathbf{x}^*) + \frac{\Big(2\big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}}\Big)^2}{2ct} \\
&\quad + \frac{4\big(\frac{1}{\sqrt{1+2cL_q}}\big)^t D_q(1+2cL_q)^{\frac{T_q}{2}}\|\boldsymbol{\lambda}(2t)\|}{2ct}
\end{aligned}$$

$$\overset{(e)}{\leq} f(\mathbf{x}^*) + \frac{1}{t}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t\Big(\frac{2D_q^2(1+2cL_q)^{T_q}}{c} + \frac{2D_q(1+2cL_q)^{\frac{T_q}{2}}\|\boldsymbol{\lambda}(2t)\|}{c}\Big)$$
$$\overset{(f)}{\leq} f(\mathbf{x}^*) + \frac{1}{t}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t\eta$$

where (a) follows from $\widetilde{\mathbf{x}}(2t) = \frac{1}{t}\sum_{\tau=t}^{2t-1}\mathbf{x}(\tau)$ and the convexity of $f(\mathbf{x})$; (b) follows from (13); (c) follows from the Cauchy-Schwarz inequality; (d) follows from Corollary 1; (e) follows from $\frac{1}{\sqrt{1+2cL_q}} < 1$; and (f) follows from (8) and the definition of η. ∎

Theorem 4. *Assume problem* (1)-(3) *satisfies Assumptions 1-3. If* $c \leq \frac{\alpha}{\beta^2}$ *in Algorithm 1, then* $g_k(\widetilde{\mathbf{x}}(2t)) \leq \frac{2D_q(1+2cL_q)^{\frac{T_q}{2}}}{ct}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t, \forall k \in \{1,2,\ldots,m\}, \forall t \geq T_q$, *where* T_q *is defined in* (12).

Proof: Fix $t \geq T_q$ and $k \in \{1,2,\ldots,m\}$. Thus, we have

$$g_k(\widetilde{\mathbf{x}}(2t)) \overset{(a)}{\leq} \frac{1}{t}\sum_{\tau=t}^{2t-1} g_k(\mathbf{x}(\tau)) \overset{(b)}{\leq} \frac{1}{ct}\big(\lambda_k(2t) - \lambda_k(t)\big)$$
$$\leq \frac{1}{ct}\|\boldsymbol{\lambda}(2t) - \boldsymbol{\lambda}(t)\|$$
$$\overset{(c)}{\leq} \frac{2D_q(1+2cL_q)^{\frac{T_q}{2}}}{ct}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t$$

where (a) follows from the convexity of $g_k(\mathbf{x})$; (b) follows from Lemma 4; and (c) follows from Corollary 1. ∎

Under Assumptions 1-3, Theorems 3 and 4 show that if $c \leq \frac{\alpha}{\beta^2}$, then $\widetilde{\mathbf{x}}(t)$ provides an ϵ-approximate solution with convergence time $O(\log(\frac{1}{\epsilon}))$.

C. Discussions

1) Practical Implementations: Assumption 3 in general is difficult to verify. However, we note that to ensure $\widetilde{\mathbf{x}}(t)$ provides the better $O(\log(\frac{1}{\epsilon}))$ convergence time, we only require $c \leq \frac{\alpha}{\beta^2}$, which is independent of the parameters in Assumptions 3. Namely, in practice, we can blindly apply Algorithm 1 to problem (1)-(3) with no need to verify Assumption 3. If problem (1)-(3) happens to satisfy Assumption 3, then $\widetilde{\mathbf{x}}(t)$ enjoys the faster convergence time $O(\log(\frac{1}{\epsilon}))$. If not, then $\widetilde{\mathbf{x}}(t)$ (or $\overline{\mathbf{x}}(t)$) at least has convergence time $O(\frac{1}{\epsilon})$.

2) Local Assumption and Local Geometric Convergence: Since Assumption 3 only requires the "quadratic" property to be satisfied in a local radius D_q around $\boldsymbol{\lambda}^*$, the error of Algorithm 1 starts to decay like $O\Big(\frac{1}{t}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t\Big)$ only after $\boldsymbol{\lambda}(t)$ arrives at the D_q local radius after T_q iterations, where T_q is independent of the approximation requirement ϵ and hence is order $O(1)$. Thus, Algorithm 1 provides an ϵ-approximate solution with convergence time $O(\log(\frac{1}{\epsilon}))$. However, it is possible that T_q is relatively large if D_q is small.

In fact, $T_q > 0$ because Assumption 3 only requires the dual function to have the "quadratic" property in a local radius. Thus, the theory developed in this section can deal with a large class of problems. On the other hand, if the dual function has the "quadratic" property globally, i.e., for all $\boldsymbol{\lambda} \geq \mathbf{0}$, then $T_q = 0$ and the error of Algorithm 1 decays like $O\Big(\frac{1}{t}\Big(\frac{1}{\sqrt{1+2cL_q}}\Big)^t\Big), \forall t \geq 1$.

3) Locally Strongly Concave Dual Functions: The following assumption is stronger than Assumption 3 but can be easier to verify in certain cases.

Assumption 4 (Locally Strongly Concave Dual Functions)**.** *Let* $\boldsymbol{\lambda}^*$ *be a Lagrange multiplier vector defined in Assumption 2. There exists* $D_c > 0$ *and* $L_c > 0$*, where the subscript* c *denotes locally strongly "concave", such that the dual function* $q(\boldsymbol{\lambda})$ *is strongly concave with modulus* L_c *over* $\{\boldsymbol{\lambda} \in \mathbb{R}_+^m : \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\| \leq D_c\}$.

In fact, Assumption 4 implies Assumption 3.

Lemma 11. *If problem* (1)-(3) *satisfies Assumption 4, then it also satisfies Assumption 3 with* $D_q = D_c$ *and* $L_q = \frac{L_c}{2}$.

Proof: See [20] for the proof. ∎

Since Assumption 4 implies Assumption 3, by the results from the previous subsection, $\widetilde{\mathbf{x}}(t)$ from Algorithm 1 provides an ϵ-approximate solution with convergence time $O(\log(\frac{1}{\epsilon}))$.

V. Applications

A. Problems with Non-Degenerate Constraint Qualifications

Theorem 5. *Consider strongly convex program* (1)-(3) *where* $f(\mathbf{x})$ *and* $g_k(\mathbf{x}), \forall k \in \{1,2,\ldots,m\}$ *are second-order continuously differentiable. Let* $\mathbf{x}^*$ *be the unique solution.*

1) *Let* $\mathcal{K} \subseteq \{1,2,\ldots,m\}$ *be the set of active constraints, i.e.,* $g_k(\mathbf{x}^*) = 0, \forall k \in \mathcal{K}$*, and denote the vector composed by* $g_k(\mathbf{x}), k \in \mathcal{K}$ *as* $\mathbf{g}_{\mathcal{K}}$*. If* $\mathbf{g}(\mathbf{x})$ *has a bounded Jacobian and rank*$(\nabla_{\mathbf{x}}\mathbf{g}_{\mathcal{K}}(\mathbf{x}^*)^T) = |\mathcal{K}|$*, then Assumptions 1-3 hold.*
2) *If* $\mathbf{g}(\mathbf{x})$ *has a bounded Jacobian and rank*$(\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^*)^T) = m$*, then Assumptions 1-4 hold.*

Proof: See [20] for the proof. ∎

Corollary 2. *Consider* $\min\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$*, where* $f(\mathbf{x})$ *is second-order continuously differentiable and strongly convex function; and* $\mathbf{A}$ *is an* $m \times n$ *matrix.*

1) *Let* $\mathbf{x}^*$ *be the optimal solution. Assume* $\mathbf{A}\mathbf{x}^* \leq \mathbf{b}$ *has* l *rows that hold with equality, and let* $\mathbf{A}'$ *be the* $l \times n$ *submatrix of* $\mathbf{A}$ *corresponding to these "active" rows. If rank*$(\mathbf{A}') = l$*, then Assumptions 1-3 hold.*
2) *If rank*$(\mathbf{A}) = m$*, then Assumptions 1-4 hold with* $D_c = \infty$.

B. Network Utility Maximization (NUM)

Consider a network with l links and n flow streams. Let $\{b_1, b_2, \ldots, b_l\}$ be the capacities of each link and $\{x_1, x_2, \ldots, x_n\}$ be the rates of each flow stream. Let $\mathcal{N}(k) \subseteq \{1,2,\ldots,n\}, 1 \leq k \leq l$ be the set of flow streams that use link k. This problem is to maximize the utility function $\sum_{i=1}^n w_i \log(x_i)$ with constants $w_i > 0, \forall 1 \leq i \leq n$, which represents a measure of network fairness [22], subject to the capacity constraint of each link. This problem is known as

the network utility maximization (NUM) problem and can be formulated as follows[3]:

$$\min \quad \sum_{i=1}^{n} -w_i \log(x_i) \tag{14}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \tag{15}$$

$$\mathbf{x} \geq \mathbf{0} \tag{16}$$

where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_n]$ is a 0-1 matrix of size $m \times n$ such that $a_{ij} = 1$ if and only if flow x_j uses link i and $\mathbf{b} > \mathbf{0}$.

Note that problem (14)-(16) satisfies Assumptions 1 and 2. By the results from Section III, $\overline{\mathbf{x}}(t)$ has convergence time $O(\frac{1}{\epsilon})$. The next theorem provides sufficient conditions such that $\widetilde{\mathbf{x}}(t)$ has better convergence time $O(\log(\frac{1}{\epsilon}))$.

Theorem 6. *The NUM problem* (14)-(16) *satisfies:*

1) *Let* $b^{\max} = \max_{1\leq i\leq n} b_i$ *and* $\mathbf{x}^{\max} > \mathbf{0}$ *such that* $x_i^{\max} > b^{\max}, \forall i \in \{1, \ldots, n\}$. *If we replace constraint* (16) *with* $\mathbf{0} \leq \mathbf{x} \leq \mathbf{x}^{\max}$ *in problem* (14)-(16), *then we obtain an equivalent problem.*
2) *Let* $\mathbf{x}^*$ *be the optimal solution. Assume* $\mathbf{A}\mathbf{x}^* \leq \mathbf{b}$ *has* m' *rows that hold with equality, and let* $\mathbf{A}'$ *be the* $m' \times n$ *submatrix of* $\mathbf{A}$ *corresponding to these "active" rows. If rank*$(\mathbf{A}') = m'$, *then Assumptions 1-3 hold for the above equivalent problem.*
3) *If rank*$(\mathbf{A}) = m$, *then Assumptions 1-4 hold for the above equivalent problem.*

Proof: See [20] for the proof. ■

VI. Numerical Results

A. Network Utility Maximization Problems

Consider the simple NUM problem described in Figure 1. Let x_1, x_2 and x_3 be the data rates of stream $1, 2$ and 3 and let the network utility be minimizing $-\log(x_1) - 2\log(x_2) - 3\log(x_3)$. It can be checked that capacity constraints other than $x_1 + x_2 + x_3 \leq 10, x_1 + x_2 \leq 8$, and $x_2 + x_3 \leq 8$ are redundant. By Theorem 6, the NUM problem can be formulated as follows:

$$\begin{aligned} \min \quad & -\log(x_1) - 2\log(x_2) - 3\log(x_3) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{x}^{\max} \end{aligned}$$

where $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 10 \\ 8 \\ 8 \end{bmatrix}$ and $\mathbf{x}^{\max} = [11, 11, 11]^T$. The optimal solution to this NUM problem is $x_1^* = 2, x_2^* = 3.2, x_3^* = 4.8$ and the optimal value is -7.7253.

Since the objective function is separable, the dual subgradient/gradient method can yield a distributed solution. This is why the dual subgradient/gradient method is widely used to solve NUM problems [4]. The objective function is strongly convex with modulus $\alpha = \frac{2}{121}$ on $\mathcal{X} = \{\mathbf{0} \leq \mathbf{x} \leq \mathbf{x}^{\max}\}$ and $\mathbf{g}(\cdot)$ is Lipschitz continuous with modulus $\beta \leq \sqrt{6}$ on $\mathcal{X}$. Figure 2 verifies the convergence of $\overline{\mathbf{x}}(t)$ with $c = \frac{\alpha}{\beta^2} = \frac{1}{363}$ and $\boldsymbol{\lambda}(0) = \mathbf{0}$. Since $\boldsymbol{\lambda}(0) = \mathbf{0}$, by Theorem 1, we have $f(\overline{\mathbf{x}}(t)) \leq f(\mathbf{x}^*), \forall t > 0$. To verify the convergence time of constraint violations, Figure 3 plots $g_1(\overline{\mathbf{x}}(t))$, $g_2(\overline{\mathbf{x}}(t))$, $g_3(\overline{\mathbf{x}}(t))$ and $1/t$ with both x-axis and y-axis in $\log_{10}$ scales. As observed in Figure 3, the curves of $g_1(\overline{\mathbf{x}}(t))$ and $g_3(\overline{\mathbf{x}}(t))$ are parallel to the curve of $1/t$ for large t. Note that $g_2(\overline{\mathbf{x}}(t)) \leq 0$ is satisfied early because this constraint is loose. Figure 3 shows that error decays like $O(\frac{1}{t})$ and suggests that the convergence time is actually $\Theta(\frac{1}{\epsilon})$ for this NUM problem.

[3]Without loss of optimality, we define $\log(0) = -\infty$ and hence $\log(\cdot)$ is defined over $\mathbb{R}_+$. Or alternatively, we can replace the non-negative rate constraints with $x_i \geq x_i^{\min}, \forall i \in \{1, 2, \ldots, n\}$ where $x_i^{\min}, \forall i \in \{1, 2, \ldots, n\}$ are sufficiently small positive numbers.

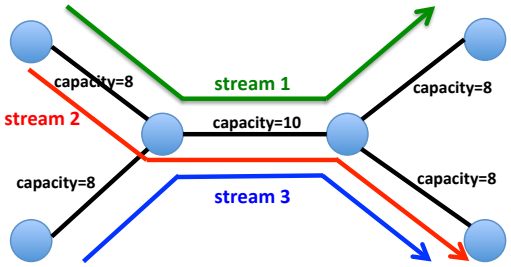

Fig. 1. A simple NUM problem with 3 flow streams

Note that $\text{rank}(\mathbf{A}) = 3$. By Theorem 6, this NUM problem satisfies Assumptions 1-4. Figure 4 verifies Theorem 6 that the convergence time of $\widetilde{\mathbf{x}}(t)$ is $O(\log(\frac{1}{\epsilon}))$ by showing that error decays like $O(\frac{1}{t}0.998^t)$ with $c = \frac{\alpha}{\beta^2} = \frac{1}{363}$.

B. Large Scale Quadratic Programs

Consider quadratic program $\min_{\mathbf{x}\in\mathbb{R}^N}\{\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{d}^T\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ where $\mathbf{Q}, \mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{d}, \mathbf{b} \in \mathbb{R}^N$. $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{U}^H \in \mathbb{R}^{N\times N}$ where $\mathbf{U}$ is the orthonormal basis for a random $N \times N$ zero mean and unit variance normal matrix and Σ is the diagonal matrix with entries from uniform $[1, 3]$. $\mathbf{A}$ is a random $N \times N$ zero mean and unit variance normal matrix. $\mathbf{d}$ and $\mathbf{b}$ are random vectors with entries from uniform $[0, 1]$. In a PC with a 4 core 2.7GHz Intel i7 Cpu and 16GB Memory, we run both Algorithm 1 and quadprog from Matlab, which by default is using the interior point method, over randomly generated large scale quadratic programs with $N = 400, 600, 800, 1000$ and 1200. For different problem size N, the running time is the average over 100 random quadratic programs and is plotted in Figure 5.

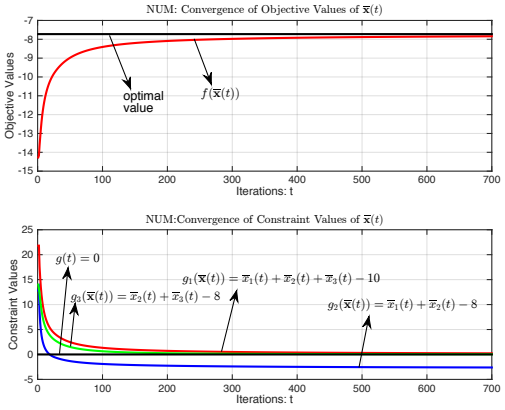

Fig. 2. The convergence of $\overline{\mathbf{x}}(t)$ for a NUM problem.

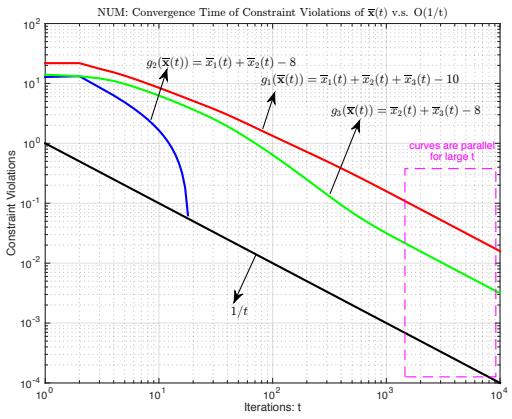

Fig. 3. The convergence time of $\overline{\mathbf{x}}(t)$ for a NUM problem.

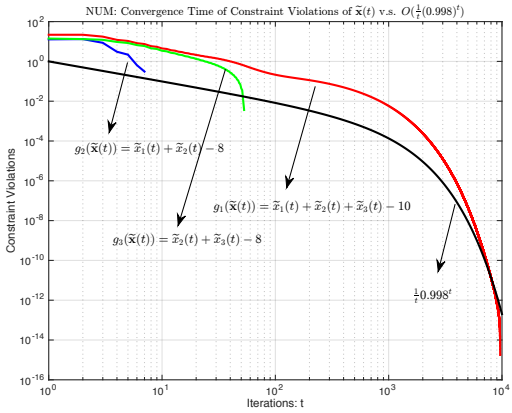

Fig. 4. The convergence time of $\widetilde{\mathbf{x}}(t)$ for a NUM problem.

VII. Conclusions

This paper studies the dual gradient method for strongly convex programs and shows that the convergence time of simple running averages is $O(\frac{1}{\epsilon})$. This paper also considers a variation of the primal averages, called the sliding running averages, and shows that if the dual function is locally quadratic then the convergence time is $O(\log(\frac{1}{\epsilon}))$.

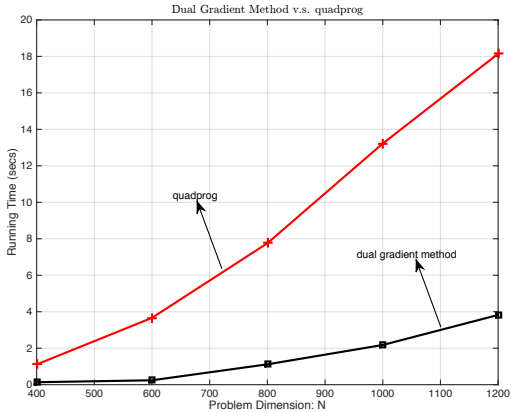

Fig. 5. The average running time for large scale quadratic programs.

VIII. Acknowledgement

We thank an anonymous reviewer for bringing to our attentions independent parallel works [16], [17] on similar topics and work [21] on linear convergence for unconstrained optimization. By using the results from [21], we improve the convergence time from $O(\frac{1}{\epsilon^{2/3}})$ in our conference version [1] to $O(\log(\frac{1}{\epsilon}))$ when the dual function is locally quadratic.

References

[1] H. Yu and M. J. Neely, "On the convergence time of the drift-plus-penalty algorithm for strongly convex programs," in *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2015.

[2] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232–1243, May 2014.

[3] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized multi-agent optimization via dual decomposition," in *IFAC World Congress*, 2011.

[4] S. H. Low and D. E. Lapsley, "Optimization flow control—I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[5] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, 2003.

[6] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. Wiley-Interscience, 2006.

[7] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[8] T. Larsson, M. Patriksson, and A.-B. Strömberg, "Ergodic, primal convergence in dual subgradient schemes for convex programming," *Mathematical programming*, vol. 86, no. 2, pp. 283–312, 1999.

[9] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, 1985.

[10] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs," *Operations Research Letters*, vol. 19, no. 3, pp. 105–113, 1996.

[11] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[12] ——, "Distributed and secure computation of convex programs over a network of connected processors," in *DCDIS International Conference on Engineering Applications and Computational Algorithms*, 2005.

[13] ——, "A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs," *arXiv:1412.0791*, 2014.

[14] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757–1780, 2009.

[15] S. Supittayapornpong, L. Huang, and M. J. Neely, "Time-average optimization with nonconvex decision set and its convergence," in *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2014.

[16] I. Necoara and A. Patrascu, "Iteration complexity analysis of dual first order methods for conic convex programming," *Optimization Method and Software*, vol. 31, no. 3, pp. 645–678, 2016.

[17] I. Necoara, A. Patrascu, and A. Nedić, "Complexity certifications of first-order inexact lagrangian methods for general convex programming: Application to real-time mpc," in *Developments in Model-Based Optimization and Control*. Springer, 2015, pp. 3–26.

[18] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[19] H. Yu and M. J. Neely, "A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 759–783, 2017.

[20] ——, "On the convergence time of dual subgradient methods for strongly convex programs," *arXiv:1503.06235*, 2015.

[21] I. Necoara, Y. Nesterov, and F. Glineur, "Linear convergence of first order methods for non-strongly convex optimization," *arXiv:1504.06298*, 2015.

[22] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.