Chapter

LAGRANGIAN METHODS FOR $O(1/t)$ CONVERGENCE IN CONSTRAINED CONVEX PROGRAMS

Michael J. Neely , Hao Yu

Reference info: M. J. Neely and H. Yu, "Lagrangian Methods for O(1/t) Convergence in Constrained Convex Programs." Convex Optimization: Theory, Methods, and Applications, edited by Arto Ruud, Nova Publishers, Jan. 2019.

Abstract

This chapter considers Lagrangian methods for numerical solutions to constrained convex programs. The dual subgradient algorithm is shown to achieve an ϵ-approximate solution with convergence time $O(1/\epsilon^2)$. An enhanced algorithm is shown to provide an improved $O(1/\epsilon)$ convergence time. Both algorithms turn the constrained minimization problem into a sequence of unconstrained minimizations. For the dual subgradient algorithm, it is shown that a Lagrange multiplier update resembles a queueing equation. Max-Weight and Lyapunov drift methods for queues are used to provide a simple performance analysis. For the enhanced algorithm, the traditional Lagrange multiplier update is modified to take a soft reflection across the zero boundary. This, together with a modified drift expression, is shown to yield improved performance with error that decays like $O(1/t)$, where t is the number of iterations.

1. Introduction

This chapter considers Lagrangian or *dual based* methods for computing an approximate solution to a general convex program. Fix n and k as positive integers. The problem of interest is to find a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ that solves:

$$\text{Minimize:} \quad f(x) \tag{1}$$

$$\text{Subject to:} \quad g_i(x) \leq 0 \quad \forall i \in \{1, \ldots, k\} \tag{2}$$

$$x \in \mathcal{X} \tag{3}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a given convex set; $f : \mathcal{X} \to \mathbb{R}$ is a given continuous and convex function (called the *objective function*); $g_i : \mathcal{X} \to \mathbb{R}$ for $i \in \{1, \ldots, k\}$ are given continuous and convex functions (called *constraint functions*). In this chapter, the functions f, $g_1, \ldots, g_k$ are not required to be smooth or differentiable.

The main idea is reduce the *constrained problem* (1)-(3) to a sequence of *unconstrained problems*. The complexity of the algorithm depends on the number of iterations in the sequence that are needed to produce an *approximate solution* within desired error bounds. The complexity also depends on the amount of computation required to solve the unconstrained problem on each iteration. Lagrangian algorithms are of interest because the computational complexity of each unconstrained problem is typically low. Two Lagrangian algorithms are considered in this chapter. The first is the well known *dual subgradient algorithm*. The second algorithm is an enhancement developed in [14] that has a similar per-iteration complexity as the dual subgradient algorithm, but uses a smaller number of iterations.

1.1 Definition of ϵ-optimal solution

It is assumed throughout this chapter that the problem (1)-(3) is *feasible*, meaning that there is at least one vector $x \in \mathbb{R}^n$ that satisfies the constraints (2)-(3). A vector $x^* \in \mathbb{R}^n$ is called an *optimal solution* to problem (1)-(3) if it satisfies the constraints (2)-(3) (so that $x^* \in \mathcal{X}$ and $g_i(x^*) \leq 0$ for all $i \in \{1, \ldots, k\}$), and if the inequality $f(x^*) \leq f(x)$ holds for all other vectors $x \in \mathbb{R}^n$ that satisfy the constraints (2)-(3). It is assumed that problem (1)-(3) has at least one optimal solution x^*.

Fix $\epsilon > 0$. A vector $x \in \mathcal{X}$ is called an *ϵ-optimal solution* to problem (1)-(3) if it satisfies

$$f(x) \leq f(x^*) + \epsilon \tag{4}$$

$$g_i(x) \leq \epsilon \quad \forall i \in \{1, \ldots, k\} \tag{5}$$

where x^* is an optimal solution to (1)-(3). Hence, an ϵ-optimal solution is a vector x in the set $\mathcal{X}$ that violates the inequality constraints by at most ϵ and that has an objective value $f(x)$ that is at most ϵ larger than the optimal objective value $f(x^*)$. A vector $x \in \mathcal{X}$ is said to be an $O(\epsilon)$-optimal solution if it satisfies (4)-(5) with the exception that the "ϵ" on each right-hand-side is replaced by some constant multiple of ϵ.

Section 2.4 describes the dual subgradient algorithm and shows that it finds an $O(\epsilon)$-optimal solution after $O(1/\epsilon^2)$ iterations. Hence, the *convergence time* is $O(1/\epsilon^2)$. Section 3.2 describes an enhanced algorithm with an improved convergence time of $O(1/\epsilon)$.

1.2 Introductory exercises

Jensen's inequality for a convex function $f : \mathcal{X} \rightarrow \mathbb{R}$ implies that

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x(t)) \geq f\left(\tfrac{1}{T}\textstyle\sum_{t=0}^{T-1} x(t)\right)$$

for all integers $T > 0$ and all vector sequences $\{x(0), x(1), \ldots x(T-1)\}$ that satisfy $x(t) \in \mathcal{X}$ for all t. The following exercises require only Jensen's inequality with the definition of *ϵ-optimal solution*.

Exercise 1. *(Convergence time of dual subgradient) Fix $\epsilon > 0$. Suppose we have an algorithm that produces a sequence of vectors $\{x(0), x(1), x(2), \ldots\}$, with $x(t) \in \mathcal{X}$ for all $t \in \{0, 1, 2, \ldots\}$, that satisfies the following for all positive integers T:*

$$\sum_{t=0}^{T-1} f(x(t)) \leq Tf(x^*) + T\epsilon$$

$$\sum_{t=0}^{T-1} g_i(x(t)) \leq \frac{1}{\epsilon} + \sqrt{T} \quad \forall i \in \{1, \ldots, k\}$$

Define $\overline{x}(T) = \frac{1}{T}\sum_{t=0}^{T-1} x(t)$. Choose a positive integer T for which $\overline{x}(T)$ is an ϵ-optimal solution. Hint: Consider T of the form a/ϵ^2 for some constant a. This value T is the convergence time. Theorem 2 in Section 2.6 shows the dual subgradient algorithm yields inequalities that are structurally similar to these.

Exercise 2. *(Convergence time of enhanced algorithm) Suppose we have an algorithm that produces a sequence of vectors $\{x(0), x(1), x(2), \ldots\}$, with $x(t) \in \mathcal{X}$ for all $t \in \{0, 1, 2, \ldots\}$, that satisfies the following for all positive integers T:*

$$\sum_{t=0}^{T-1} f(x(t)) \leq Tf(x^*) + c$$

$$\sum_{t=0}^{T-1} g_i(x(t)) \leq c \quad \forall i \in \{1, \ldots, k\}$$

for some constant $c > 0$. Fix $\epsilon > 0$. Define $\overline{x}(T) = \frac{1}{T}\sum_{t=0}^{T-1} x(t)$. Choose a positive integer T for which $\overline{x}(T)$ is an ϵ-optimal solution. Your answer should have an asymptotically better convergence time compared with that of Exercise 1. Section 3.2 presents an algorithm that yields inequalities similar to these.

2. Dual subgradient algorithm

This section develops and analyzes the dual subgradient algorithm as a numerical solution to problem (1)-(3). The dual subgradient algorithm received its name because it was originally developed using duality concepts. The traditional duality analysis requires additional assumptions such as *strict convexity* (see, for example, [1] [2]). Later, the dual subgradient algorithm was shown to be a special case of a more general *drift-plus-penalty* algorithm for stochastic problems [9] [8]. Rather than motivated by duality concepts, the drift-plus-penalty analysis is motivated by *queueing theory concepts* and introduces a simple averaging step that removes the need for strict convexity. This section shall use the queueing theory development. This is arguably simpler and provides physical intuition that shall be useful in the enhanced algorithm of Section 3.2. For completeness, a brief discussion of the traditional duality motivations are given in Section 2.8.

2.1 Compact set assumption

It is convenient to impose the additional assumption that the convex set $\mathcal{X}$ is also a compact set. That is, the set $\mathcal{X}$ is assumed to be convex, closed, and bounded. Since the functions $f, g_1, \ldots, g_k$ are continuous over the domain $\mathcal{X}$, the assumption that $\mathcal{X}$ is a compact set ensures that these functions are bounded over $\mathcal{X}$. This compactness assumption is used for the dual subgradient algorithm, but is not required for the enhanced algorithm of Section 3.2.

2.2 Time averages and virtual queues

Consider the convex program (1)-(3). The problem shall be solved over a sequence of *time slots* $t \in \{0, 1, 2, \ldots\}$, where $x(t) \in \mathcal{X}$ is a *decision vector* that is computed on slot t. The particular decision vectors shall be determined later. This subsection demonstrates a time average property that holds for *every sequence* of vectors $x(t) \in \mathcal{X}$. Fix T as a positive integer and define $\overline{x}(T)$ as the time average of the $x(t)$ vectors over the first T slots:

$$\overline{x}(T) = \frac{1}{T}\sum_{t=0}^{T-1} x(t)$$

Notice that $\overline{x}(T)$ is a convex combination of points in the convex set $\mathcal{X}$, and so $\overline{x}(T) \in \mathcal{X}$. By *Jensen's inequality* for the convex functions $f, g_1, \ldots, g_k$ we have

$$\begin{aligned} f(\overline{x}(T)) &\leq \frac{1}{T}\sum_{t=0}^{T-1} f(x(t)) \\ g_i(\overline{x}(T)) &\leq \frac{1}{T}\sum_{t=0}^{T-1} g_i(x(t)) \quad \forall i \in \{1, \ldots, k\} \end{aligned}$$

These inequalities suggest that the convex program (1)-(3) can be solved by choosing decisions $x(t) \in \mathcal{X}$ over time that minimize the *time average* of $f(x(t))$ subject to *time averages* of $g_i(x(t))$ being less than or equal to zero. To enforce the desired inequality constraints $g_i(x) \leq 0$ in (2), for each $i \in \{1, \ldots, k\}$, define a sequence $Q_i(t)$ that evolves in discrete time

$t \in \{0, 1, 2, \ldots\}$ according to the update equation:

$$Q_i(t+1) = \max\{Q_i(t) + g_i(x(t)), 0\} \tag{6}$$

with initial condition $Q_i(0) = 0$. The sequence $Q_i(t)$ shall be called a *virtual queue process* for reasons explained after the following lemma.

Lemma 1. *(Virtual queues) Consider the update rule* (6) *under any sequence of vectors $x(t) \in \mathcal{X}$ for $t \in \{0, 1, 2, \ldots\}$. Assume the functions $g_i : \mathcal{X} \to \mathbb{R}$ are convex for each $i \in \{1, \ldots, k\}$. Then for all positive integers T and all $i \in \{1, \ldots, k\}$ we have*

$$\sum_{t=0}^{T-1} g_i(x(t)) \leq Q_i(T) \tag{7}$$

and so by Jensen's inequality

$$g_i(\overline{x}(T)) \leq \frac{Q_i(T)}{T} \tag{8}$$

Proof. Fix $i \in \{1, \ldots, k\}$ and $t \in \{0, 1, 2, \ldots\}$. We have from (6)

$$\begin{aligned} Q_i(t+1) &= \max\{Q_i(t) + g_i(x(t)), 0\} \\ &\geq Q_i(t) + g_i(x(t)) \end{aligned}$$

and so

$$Q_i(t+1) - Q_i(t) \geq g_i(x(t))$$

Let T be a positive integer. Summing the above over $t \in \{0, 1, \ldots, T-1\}$ gives

$$\underbrace{\sum_{t=0}^{T-1}[Q_i(t+1) - Q_i(t)]}_{Q_i(T) - Q_i(0)} \geq \sum_{t=0}^{T-1} g_i(x(t))$$

The summation $\sum_{t=0}^{T-1}[Q_i(t+1) - Q_i(t)]$ is called a *telescoping sum* because of its simple cancellations. Substituting $Q_i(0) = 0$ proves (7). Dividing by T and using Jensen's inequality proves (8). □

The above lemma shows that if $Q_i(T)/T$ is small, then the vector $\overline{x}(T)$ is close to satisfying the desired inequality constraint $g_i(x) \leq 0$. The lemma is motivated by the following physical intuition about queueing systems. Notice that we can write

$$g_i(x(t)) = g_i(x(t))^+ - g_i(x(t))^-$$

where $g_i(x(t))^+ = \max\{g_i(x(t)), 0\}$ and $g_i(x(t))^- = -\min\{g_i(x(t)), 0\}$ are defined as the positive and negative parts of the value $g_i(x(t))$. Thus, the update equation (6) is equivalent to:

$$Q_i(t+1) = \max\left\{ Q_i(t) \; + \; \underbrace{g_i(x(t))^+}_{\text{arrivals}} \; - \; \underbrace{g_i(x(t))^-}_{\text{service}}, 0 \right\}$$

This is a standard *discrete time queueing equation*, where $Q_i(t)$ can be viewed as the *queue backlog on slot* t, $g_i(x(t))^+$ can be viewed as the *new arrivals* on slot t, and $g_i(x(t))^-$ can be viewed as the *offered service* on slot t. It says that the queue backlog on slot $t+1$ is equal to the queue backlog on slot t, plus the new arrivals, minus the offered service. The equation takes a max with zero since queue backlog cannot be negative. The sequence $Q_i(t)$ is called a *virtual* queue process because the arrivals, service, and backlog are not objects in a physical system. Rather, they exist only as variables in a virtual system that is implemented in software. Lemma 1 transforms the problem of finding a vector $x \in \mathcal{X}$ to (approximately) satisfy $g_i(x) \leq 0$ into a queue control problem that seeks a *sequence* of vectors $x(t) \in \mathcal{X}$ that maintain small values of $Q_i(T)/T$.

2.3 Lyapunov drift

Motivated by the physical intuition of queueing systems described in the previous subsection, we employ a *drift-based algorithm* that stabilizes the queues while optimizing a performance objective. To this end, for each slot $t \in \{0, 1, 2, \ldots\}$ define the virtual queue vector $Q(t) = (Q_1(t), \ldots, Q_k(t))$ and define $L(t)$ by

$$L(t) = \frac{1}{2}||Q(t)||^2 = \frac{1}{2}\sum_{i=1}^{k} Q_i(t)^2$$

where, throughout this chapter, the norm $||\cdot||$ denotes the standard Euclidean norm (the square root of the sum of squares of the vector components). The value $L(t)$ is a scalar measure of the size of the queue vector on slot t and shall be called a *Lyapunov function*. Define $\Delta(t)$ as the change in $L(t)$ from slot t to slot $t+1$, which shall be called the *Lyapunov drift*:[1]

$$\Delta(t) = L(t+1) - L(t)$$

Lemma 2. *(Lyapunov drift) Consider the update rule* (6) *under any sequence of decision vectors $x(t) \in \mathcal{X}$ for $t \in \{0, 1, 2, \ldots\}$. Suppose that $\mathcal{X} \subseteq \mathbb{R}^n$ is a compact set. Then*

$$\Delta(t) \leq B + \sum_{i=1}^{k} Q_i(t) g_i(x(t)) \quad \forall t \in \{0, 1, 2, \ldots\}$$

where B is defined

$$B = \sup_{x \in \mathcal{X}} \frac{1}{2} \sum_{i=1}^{k} g_i(x)^2$$

The constant B is finite since the functions g_i are continuous over the compact domain $\mathcal{X}$.

Proof. Fix $i \in \{1, \ldots, k\}$ and $t \in \{0, 1, 2, \ldots\}$. By (6),

$$\begin{aligned} Q_i(t+1)^2 &= \max\{Q_i(t) + g_i(x(t)), 0\}^2 \\ &\overset{(a)}{\leq} [Q_i(t) + g_i(x(t))]^2 \\ &= Q_i(t)^2 + g_i(x(t))^2 + 2Q_i(t)g_i(t) \end{aligned}$$

[1] Traditionally, a Lyapunov function is defined on the state space of a system. Our function $L(t) = \frac{1}{2}||Q(t)||^2$ can indeed be viewed as a function of the current queue state vector $Q(t) = (Q_1(t), \ldots, Q_k(t))$. The Lyapunov drift is traditionally defined for stochastic problems as a conditional expected change in the Lyapunov function given the current state, namely, $\mathbb{E}\left[L(t+1) - L(t)|Q(t)\right]$. There is nothing stochastic in this chapter and so the conditional expectation can be removed. We still call the resulting quantity $L(t+1) - L(t)$ the *Lyapunov drift*. To emphasize dependence on the queue state vector $Q(t)$, one could use alternative notation $\tilde{L}(Q(t))$ and $\tilde{\Delta}(Q(t))$, where $L(t) = \tilde{L}(Q(t))$ and $\Delta(t) = \tilde{\Delta}(Q(t))$.

where (a) holds by the fact $(\max\{r,0\})^2 \leq r^2$ for all $r \in \mathbb{R}$. Summing over $i \in \{1,\ldots,k\}$ and dividing by 2 gives

$$\underbrace{\frac{1}{2}\sum_{i=1}^{k} Q_i(t+1)^2}_{L(t+1)} \leq \underbrace{\frac{1}{2}\sum_{i=1}^{k} Q_i(t)^2}_{L(t)} + \frac{1}{2}\sum_{i=1}^{k} g_i(x(t))^2 + \sum_{i=1}^{k} Q_i(t) g_i(x(t))$$

Observing that $\frac{1}{2}\sum_{i=1}^{k} g_i(x(t))^2 \leq B$ yields the result. □

2.4 The dual subgradient algorithm

Fix $\epsilon > 0$. The technique is to choose $x(t) \in \mathcal{X}$ on each slot t to minimize a bound on the following *drift-plus-penalty* expression:

$$\epsilon \underbrace{\Delta(t)}_{\text{drift}} + \underbrace{f(x(t))}_{\text{penalty}}$$

where ϵ is a weight that determines the relative importance of minimizing the drift component $\Delta(t)$ versus minimizing the "penalty" component $f(x(t))$. The parameter $\epsilon > 0$ is called the *stepsize* (for reasons explained in Section 2.8). It shall be shown that smaller values of ϵ place less emphasis on drift minimization and lead to larger virtual queue sizes, with the benefit of yielding solutions that are closer to minimizing the objective function f. By Lemma 2 we have the bound

$$\epsilon\Delta(t) + f(x(t)) \leq \epsilon B + \underbrace{f(x(t)) + \epsilon\sum_{i=1}^{k} Q_i(t) g_i(x(t))}_{\text{minimize every slot } t} \tag{9}$$

Every slot $t \in \{0,1,2,\ldots\}$ the algorithm observes the current queue vector $(Q_1(t),\ldots,Q_k(t))$ and chooses $x(t) \in \mathcal{X}$ to greedily minimize the expression marked by an underbrace in (9).

Specifically, the dual subgradient algorithm for problem (1)-(3) is as follows: Fix the stepsize $\epsilon > 0$, initialize $Q_i(0) = 0$ for all $i \in \{1,\ldots,k\}$, and formally define $\overline{x}(0) = 0$. Every slot $t \in \{0,1,2,\ldots\}$ do

- Choose $x(t) \in \mathcal{X}$ to minimize the expression:[2]

$$f(x(t)) + \epsilon \sum_{i=1}^{k} Q_i(t) g_i(x(t))$$

- Update virtual queues for each $i \in \{1, \ldots, k\}$ via:

$$Q_i(t+1) = \max\{Q_i(t) + g_i(x(t)), 0\}$$

- Update the time average vector $\overline{x}(t)$ via:

$$\overline{x}(t+1) = \left(\frac{t}{t+1}\right) \overline{x}(t) + \left(\frac{1}{t+1}\right) x(t)$$

2.5 Basic performance analysis for dual subgradient algorithm

Theorem 1. *(Basic performance) Suppose the set $\mathcal{X}$ is convex and compact and the convex program* (1)-(3) *has an optimal solution $x^* \in \mathcal{X}$. Fix $\epsilon > 0$ and implement the above dual subgradient algorithm using stepsize ϵ. Then for all $T \in \{1, 2, 3, \ldots\}$ we have $\overline{x}(T) \in \mathcal{X}$ and*

$$f\left(\overline{x}(T)\right) \leq f(x^*) + \epsilon B \tag{10}$$

$$g_i\left(\overline{x}(T)\right) \leq \sqrt{\frac{2(f(x^*) - f_{min})}{\epsilon T} + \frac{2B}{T}} \quad \forall i \in \{1, \ldots, k\} \tag{11}$$

where constants f_{min} and B are defined:[3]

$$B = \sup_{x \in \mathcal{X}} \frac{1}{2} \sum_{i=1}^{k} g_i(x)^2$$

$$f_{min} = \inf_{x \in \mathcal{X}} f(x)$$

In particular, for any desired $\epsilon > 0$, implementing the dual subgradient algorithm with stepsize ϵ produces an $O(\epsilon)$-optimal solution whenever the number of iterations satisfies $T \geq 1/\epsilon^3$.

[2] This seeks to minimize a continuous function over the compact set $\mathcal{X}$, and so at least one minimizer exists.

[3] The constants B, f_{min} are finite because functions f and g_i are continuous over the compact domain $\mathcal{X}$.

Proof. Fix $t \in \{0, 1, 2, \ldots\}$. By (9) we have

$$\begin{aligned}
\epsilon\Delta(t) + f(x(t)) &\leq \epsilon B + f(x(t)) + \epsilon\sum_{i=1}^{k} Q_i(t)g_i(x(t)) \\
&\overset{(a)}{\leq} \epsilon B + f(x^*) + \epsilon\sum_{i=1}^{k} Q_i(t)g_i(x^*) \\
&\overset{(b)}{\leq} \epsilon B + f(x^*)
\end{aligned}$$

where (a) holds because the algorithm chooses $x(t)$ to minimize $f(x) + \epsilon\sum_{i=1}^{k} Q_i(t)g_i(x)$ over all $x \in \mathcal{X}$, and x^* is just another vector in $\mathcal{X}$; (b) holds because $Q_i(t) \geq 0$ and $g_i(x^*) \leq 0$ for all $i \in \{1, \ldots, k\}$ (since x^* satisfies the constraints of problem (1)-(3)). Fix T as a positive integer. Substituting $\Delta(t) = L(t+1) - L(t)$ gives

$$\epsilon[L(t+1) - L(t)] + f(x(t)) \leq \epsilon B + f(x^*)$$

Summing over $t \in \{0, \ldots, T-1\}$ and observing the telescoping sum gives

$$\epsilon[L(T) - L(0)] + \sum_{t=0}^{T-1} f(x(t)) \leq \epsilon BT + Tf(x^*)$$

Substituting $L(0) = 0$, $L(T) = \frac{1}{2}||Q(T)||^2$, and dividing by T gives

$$\frac{\epsilon}{2T}||Q(T)||^2 + \frac{1}{T}\sum_{t=0}^{T-1} f(x(t)) \leq f(x^*) + \epsilon B \tag{12}$$

The inequality (10) follows from the above inequality by noting that $||Q(T)|| \geq 0$ and using Jensen's inequality for the convex function $f : \mathcal{X} \to \mathbb{R}$.

To prove (11), rearranging (12) gives

$$\begin{aligned}
\frac{||Q(T)||^2}{T^2} &\leq \frac{2}{\epsilon T^2}\sum_{t=0}^{T-1}[f(x^*) - f(x(t))] + \frac{2B}{T} \\
&\leq \frac{2(f(x^*) - f_{min})}{\epsilon T} + \frac{2B}{T}
\end{aligned}$$

Hence for each $i \in \{1, \ldots, k\}$

$$\frac{Q_i(T)}{T} \leq \frac{||Q(T)||}{T} \leq \sqrt{\frac{2(f(x^*) - f_{min})}{\epsilon T} + \frac{2B}{T}}$$

which yields (11) upon application of the virtual queue lemma (Lemma 1). □

2.6 Improved convergence with a Lagrange multiplier assumption

Theorem 1 ensures the dual subgradient algorithm produces an $O(\epsilon)$-optimal solution with convergence time $O(1/\epsilon^3)$. The convergence time bound can be improved to $O(1/\epsilon^2)$ under the following *Lagrange multiplier assumption*.

Assumption 1. *(Lagrange multiplier) Assume the problem* (1)-(3) *has a Lagrange multiplier vector* $\mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k$*, so that* $\mu_i \geq 0$ *for all* $i \in \{1, \ldots, k\}$ *and*

$$f(x) + \sum_{i=1}^{k} \mu_i g_i(x) \geq f(x^*) \quad \forall x \in \mathcal{X} \tag{13}$$

where x^* *is an optimal solution to* (1)-(3).

The Lagrange multiplier assumption (Assumption 1) is mild and holds in many cases, such as whenever a Slater condition holds (so that there is a vector $z \in \mathcal{X}$ and a real number $\delta > 0$ such that $g_i(z) \leq -\delta$ for all $i \in \{1, \ldots, k\}$) or when the set $\mathcal{X}$ is polyhedral and the constraint functions are affine (see, for example, [1] [2]).

Theorem 2. *(Performance with a Lagrange multiplier) Suppose the set* $\mathcal{X}$ *is convex and compact, the convex program* (1)-(3) *has an optimal solution* $x^* \in \mathcal{X}$*, and there exists a (nonnegative) Lagrange multiplier vector* μ *that satisfies* (13)*. Fix* $\epsilon > 0$ *and implement the dual subgradient algorithm using stepsize* ϵ*. Then for all* $T \in \{1, 2, 3, \ldots\}$ *we have* $\overline{x}(T) \in \mathcal{X}$ *and*

$$f(\overline{x}(T)) \leq f(x^*) + \epsilon B \tag{14}$$

$$g_i(\overline{x}(T)) \leq \frac{||\mu||}{T\epsilon} + \sqrt{\frac{||\mu||^2}{T^2\epsilon^2} + \frac{2B}{T}} \quad \forall i \in \{1, \ldots, k\} \tag{15}$$

where $||\mu|| = \sqrt{\sum_{i=1}^{k} \mu_i^2}$, and the constants f_{min} and B are as defined in Theorem 1. In particular, for any desired $\epsilon > 0$, implementing the dual subgradient algorithm with stepsize ϵ produces an $O(\epsilon)$-optimal solution whenever the number of iterations satisfies $T \geq 1/\epsilon^2$.

Proof. Inequality (14) has already been proven in Theorem 1. It suffices to prove (15). Rearranging (12) we have for all positive integers T:

$$\begin{aligned}
||Q(T)||^2 &\leq 2BT + \frac{2}{\epsilon}\sum_{t=0}^{T-1}[f(x^*) - f(x(t))] \\
&\overset{(a)}{\leq} 2BT + \frac{2}{\epsilon}\sum_{t=0}^{T-1}\sum_{i=1}^{k}\mu_i g_i(x(t)) \\
&= 2BT + \frac{2}{\epsilon}\sum_{i=1}^{k}\mu_i\sum_{t=0}^{T-1} g_i(x(t)) \\
&\overset{(b)}{\leq} 2BT + \frac{2}{\epsilon}\sum_{i=1}^{k}\mu_i Q_i(T) \\
&\overset{(c)}{\leq} 2BT + \frac{2}{\epsilon}||\mu|| \cdot ||Q(T)||
\end{aligned}$$

where (a) holds by the Lagrange multiplier assumption (13) and the fact that $x(t) \in \mathcal{X}$ for all t; (b) holds by the virtual queue lemma (Lemma 1); (c) holds by the Cauchy-Schwarz inequality. Define $y = ||Q(T)||$, $b = -\frac{2}{\epsilon}||\mu||$, $c = -2BT$. The above inequality reduces to the quadratic inequality $y^2 + by + c \leq 0$ and so

$$y \leq \frac{-b + \sqrt{b^2 - 4c}}{2} = \frac{||\mu||}{\epsilon} + \sqrt{\frac{||\mu||^2}{\epsilon^2} + 2BT}$$

Since $y = ||Q(T)||$, for each $i \in \{1, \ldots, k\}$ we have

$$\frac{Q_i(T)}{T} \leq \frac{||Q(T)||}{T} \leq \frac{||\mu||}{T\epsilon} + \sqrt{\frac{||\mu||^2}{T^2\epsilon^2} + \frac{2B}{T}}$$

The result of (15) follows by application of the virtual queue lemma (Lemma 1). □

2.7 An interpretation of steepest ascent over dual variables

The dual subgradient algorithm specified above can equivalently be described using *scaled values* $q_i(t) = \epsilon Q_i(t)$. Specifically, initialize $q_i(0) = 0$ for all $i \in \{1, \ldots, k\}$. Every slot $t \in \{0, 1, 2, \ldots\}$ do the following:

- Choose $x(t) \in \mathcal{X}$ to minimize the expression:

$$f(x(t)) + \sum_{i=1}^{k} q_i(t) g_i(x(t)) \tag{16}$$

- Update via

$$q_i(t+1) = \max\{q_i(t) + \epsilon g_i(x(t)), 0\} \tag{17}$$

- Update the average vector $\overline{x}(t)$ as before.

The traditional motivation for the dual subgradient algorithm comes from examining the *dual function* of the convex program (1)-(3). Define $\mathcal{D} \subseteq \mathbb{R}^k$ as the set of *dual variables*:

$$\mathcal{D} = \{(q_1, \ldots, q_k) \in \mathbb{R}^k : q_i \geq 0 \quad \forall i \in \{1, \ldots, k\}\}$$

The dual function $d : \mathcal{D} \to \mathbb{R}$ is defined:

$$d(q) = \inf_{x \in \mathcal{X}} \left[f(x) + \sum_{i=1}^{k} q_i g_i(x) \right] \tag{18}$$

where the infimum is achievable and finite because the functions $f, g_1, \ldots, g_k$ are continuous and $\mathcal{X}$ is compact. With this definition, it follows that if problem (1)-(3) has an optimal solution x^*, and if there exists a Lagrange multiplier vector $\mu \in \mathcal{D}$ that satisfies (13), then for all $q \in \mathcal{D}$ and all $x \in \mathcal{X}$ we have

$$d(q) \overset{(a)}{\leq} f(x^*) \overset{(b)}{\leq} f(x) + \sum_{i=1}^{k} \mu_i g_i(x) \tag{19}$$

where (a) holds by the infimum definition of $d(q)$ in (18) and the fact $x^* \in \mathcal{X}$ and $q_i g_i(x^*) \leq 0$ for all i; (b) holds by the Lagrange multiplier assumption (13).

Taking an infimum of both sides of this inequality over all $x \in \mathcal{X}$ and using the definition of $d(\mu)$ in (18) gives

$$d(q) \leq f(x^*) \leq d(\mu) \quad \forall q \in \mathcal{D} \tag{20}$$

In particular, the Lagrange multiplier vector $\mu \in \mathcal{D}$ maximizes $d(q)$ over all other vectors $q \in \mathcal{D}$. The maximum value is $d(\mu) = f(x^*)$. Furthermore, one can show (see, for example, [1]) that the dual function $d(q)$ is concave and has the following *subgradient* at each point $q \geq 0$:

$$d'(q) = (g_1(x_q), \ldots, g_k(x_q))$$

with $x_q \in \mathcal{X}$ defined as any vector that satisfies:

$$x_q \in \arg \inf_{x \in \mathcal{X}} \left[f(x) + \sum_{i=1}^{k} q_i g_i(x) \right]$$

With this interpretation, we can attempt to find the optimal Lagrange multiplier μ by attempting to maximizing the (concave) dual function $d(q)$ with a *steepest ascent algorithm* that sequentially computes guesses $q(t)$ and updates these guesses over time. Every step t, the new guess $q(t+1)$ is computed by starting with $q(t)$ and moving in the direction of the subgradient $d'(q(t))$ with a *stepsize* $\epsilon > 0$. Specifically:

- Initialize a Lagrange multiplier guess $q(0) = 0 \in \mathbb{R}^k$.
- For each step $t \in \{0, 1, 2, \ldots\}$, observe the current $q(t) \in \mathbb{R}^k$ and compute the subgradient $d'(q(t))$ via

$$x(t) \in \arg \inf_{x \in \mathcal{X}} \left[f(x) + \sum_{i=1}^{k} q_i(t) g_i(x) \right] \tag{21}$$

$$d'(q(t)) = [g_1(x(t)), g_2(x(t)), \ldots, g_k(x(t))] \tag{22}$$

  The $x(t)$ decision in (21) is identical to the decision of (16).

- Choose $q(t+1) \in \mathbb{R}^k$ by moving from $q(t)$ in the direction $d'(q(t))$ with stepsize ϵ:

$$q(t+1) = Proj_{\mathcal{D}}[q(t) + \epsilon d'(q(t))] \tag{23}$$

where $Proj_{\mathcal{D}}[\cdot]$ denotes a projection onto the set $\mathcal{D} \subseteq \mathbb{R}^k$ of nonnegative vectors. The projection takes the componentwise maximum with zero. Substituting (22) into (23) reveals that the update (23) is identical to the update (17).

The steps (21)-(23) of this steepest ascent procedure are identical to the steps (16)-(17) of the dual subgradient algorithm. It is remarkable that projecting onto the set of nonnegative dual variables is the same as the $\max\{\cdot, 0\}$ operation of a queueing equation. However, this steepest ascent procedure focuses on the *dual variables* $q(t)$ and does not compute a time average of the *primal variables* $x(t)$. Steepest ascent does not always produce $q(t)$ sequences that converge to the optimal value μ (see [12] [1] [2] for analysis of steepest descent for convex functions). Further, even if the optimal μ were known exactly, it is not always clear how to use μ to find the solution x^* to the desired convex program (1)-(3). It cannot always be done by solving the minimization problem $\inf_{x\in\mathcal{X}}[f(x) + \sum_{i=1}^{k} \mu_i g_i(x)]$ because, while every optimal solution x^* to the convex program is also a solution to this minimization problem, there may be infinitely many more solutions to this minimization problem, including ones that do not satisfy the inequality constraints (2).[4] This is where the time averaging step and the virtual queue analysis come to the rescue: The sequential procedure for computing the primal variables $x(t)$ leads to a time averaged vector $\overline{x}(t)$ that gets closer and closer to satisfying the desired inequality constraints (Theorems 1 and 2).

2.8 Dual subgradient algorithm exercises

Exercise 3. *(Approximate implementation) Suppose that an approximate version of the dual subgradient algorithm is implemented (with stepsize $\epsilon > 0$),*

[4] If the function $f : \mathcal{X} \to \mathbb{R}$ is *strictly convex* it can be shown that x^* is the *unique* solution to $\inf_{x\in\mathcal{X}}[f(x) + \sum_{i=1}^{k} \mu_i g_i(x)]$. Strict convexity fails when f is an affine function. For example, if $\mathcal{X}$ is the unit hypercube $[0,1]^n$ and $f, g_1, \ldots, g_k$ are affine functions, then $\inf_{x\in\mathcal{X}}[f(x) + \sum_{i=1}^{k} \mu_i g_i(x)]$ reduces to separately choosing each component $x_i \in [0,1]$ to minimize an affine function, and can be solved by choosing binary-valued components $x_i \in \{0,1\}$. However, the linear program of finding $x \in [0,1]^n$ to minimize $f(x)$ subject to $g_i(x) \leq 0$ for all $i \in \{1,\ldots,k\}$ may not have binary-valued solutions. Nevertheless, the *time average* of the binary-valued primal variables $x(t)$ chosen at each step of the dual subgradient algorithm can approach a non-binary solution to the linear program (as shown in Theorems 1 and 2).

with the only difference that every slot $t \in \{0, 1, 2, \ldots\}$ a vector $x(t) \in \mathcal{X}$ is chosen to satisfy

$$f(x(t)) + \epsilon \sum_{i=1}^{k} Q_i(t) g_i(x(t)) \leq \epsilon C + f(x) + \epsilon \sum_{i=1}^{k} Q_i(t) g_i(x) \quad \forall x \in \mathcal{X}$$

for some constant $C > 0$. Thus, $x(t)$ does not necessarily minimize the desired expression, but comes within ϵC of minimizing it (exact minimization holds when $C = 0$). Use (9) *to show that for all slots t we have:*

$$\epsilon \Delta(t) + f(x(t)) \leq \epsilon(B + C) + f(x(t)) + \epsilon \sum_{i=1}^{k} Q_i(t) g_i(x(t))$$

Follow the proofs of Theorems 1 and 2 to verify that, under the same assumptions as the theorems, the inequalities (10)-(11) *and* (14)-(15) *still hold when all instances of "B" are replaced by "$B + C$."*

Exercise 4. *(Separable problems) Consider* (1)-(3) *with $\mathcal{X} = [0, 1]^n$ and f, $g_1, \ldots, g_k$ defined as separable sums of single variable convex functions:*

$$\begin{aligned} f(x) &= \textstyle\sum_{j=1}^{n} f_j(x_j) \\ g_i(x) &= \textstyle\sum_{j=1}^{n} g_{ij}(x_j) \quad \forall i \in \{1, \ldots, k\} \end{aligned}$$

Show that the choice of $x(t) \in \mathcal{X}$ at every step of the dual subgradient algorithm (with stepsize $\epsilon > 0$) reduces to separately choosing each component $x_j(t)$ to minimize a single-variable convex function over the interval $[0, 1]$.

Exercise 5. *(Network flow control) Consider the 2-link network with 4 traffic flows shown in Fig. 1. The link capacities C_1, C_2 are given positive numbers. Let $x = (x_1, x_2, x_3, x_4)$ be the vector of flow rates. The problem is to maximize a concave network utility function subject to the link capacity constraints:*

$$\begin{aligned} \textit{Maximize:} \quad & \sum_{i=1}^{4} w_i \log(1 + x_i) \\ \textit{Subject to:} \quad & x_1 + x_2 + x_3 \leq C_1 \\ & x_2 + x_3 + x_4 \leq C_2 \\ & (x_1, x_2, x_3, x_4) \in \mathcal{X} \end{aligned}$$

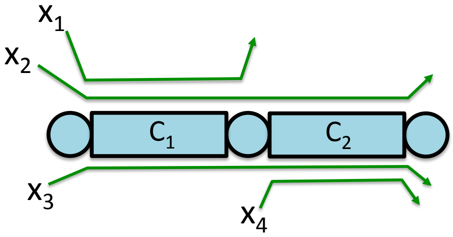

Figure 1. The 2-link network with 4 traffic flows for the network utility maximization problem of Exercise 5.

where $\mathcal{X} = [0,1]^4$ and $w_1, \ldots, w_4$ are given positive constants.[5]

a) Write the corresponding convex program in the form (1)-(3). *Hint: use $f(x) = -\sum_{i=1}^{4} w_i \log(1+x_i)$.*

b) Suppose we implement the dual subgradient algorithm with stepsize $\epsilon > 0$. Specify the update equations for the two virtual queues $Q_1(t)$ and $Q_2(t)$.

c) Continuing part (b), specify the (separable) decisions $x_i(t) \in [0,1]$ for each $t \in \{0,1,2,\ldots\}$ and each $i \in \{1,\ldots,4\}$. Be careful to distinguish cases when one or more queues are empty, and to ensure all decisions satisfy $0 \leq x_i(t) \leq 1$.

2.9 Notes on the dual subgradient algorithm

- The $O(1/\epsilon^2)$ convergence time result for the dual subgradient algorithm (with use of the averaged vector $\overline{x}(T)$) was proven under a more stringent Slater condition in [11], and independently in [6]. The $O(1/\epsilon^2)$ convergence time result under the weaker Lagrange multiplier assumption was proven in [10].

- The work [13] shows that, under certain piecewise linear assumptions, taking a time average over the *second half* interval $\{T/2, \ldots, T\}$ gives

[5] It can be shown that the convex program for the network flow problem of Exercise 5 is *strongly convex* and hence the dual subgradient algorithm enjoys a faster $O(1/\epsilon)$ convergence [5] [15]. This problem would not be strongly convex if it involved routing over multiple paths, since the constraints would then contain additional variables that do not appear in the objective function. In that case the convergence of dual subgradient reverts to the slower $O(1/\epsilon^2)$ time, and the enhanced algorithm of Section 3.2 would be needed to achieve the faster $O(1/\epsilon)$ convergence.

the dual subgradient algorithm an $O(1/\epsilon)$ convergence time.

- The dual subgradient algorithm is known to achieve an improved $O(1/\epsilon)$ convergence time if the objective function f is *strongly convex* [5] [15]. The convergence time is $O(\log(1/\epsilon))$ if f is strongly convex and the dual function has a locally quadratic property (such as being strongly concave) [15].
- Primal-dual algorithms that offer $O(1/\epsilon^2)$ convergence time while replacing the minimization step with a projection onto the set $\mathcal{X}$ are given in [7] [17].
- Stochastic generalizations of the dual subgradient algorithm are given in [9] [4] [8].

3. An enhanced Lagrangian algorithm

This section considers the convex program (1)-(3) and presents a Lagrangian algorithm from [14] that is faster than the dual subgradient algorithm.

3.1 Assumptions

Consider the convex program (1)-(3). Assume $\mathcal{X} \subseteq \mathbb{R}^n$ is a closed convex set (not necessarily compact). Assume $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g_i : \mathcal{X} \rightarrow \mathbb{R}$ for $i \in \{1, \ldots, k\}$ are continuous and convex functions. Define $g : \mathcal{X} \rightarrow \mathbb{R}^k$ by $g(x) = (g_1(x), \ldots, g_k(x))$.

Assumption 2. *(Lipschitz continuous constraint functions) Assume the function $g : \mathcal{X} \rightarrow \mathbb{R}^k$ is Lipschitz continuous with parameter $\beta > 0$, so that*

$$||g(x) - g(y)|| \leq \beta ||x - y|| \quad \forall x, y \in \mathcal{X}$$

where the norm $||\cdot||$ represents the standard Euclidean norm (square root of sum of squares).

Assume the convex program (1)-(3) has at least one optimal solution. Let x^* be a particular optimal solution. The Lagrange multiplier assumption (Assumption 1) is assumed to hold, so that there is a nonnegative vector $\mu = (\mu_1, \ldots, \mu_k)$

such that

$$f(x) + \mu^T g(x) \geq f(x^*) \quad \forall x \in \mathcal{X}$$

3.2 The enhanced algorithm

The enhanced algorithm for problem (1)-(3) is as follows: Fix an algorithm parameter $\alpha > 0$, initialize an arbitrary $x(-1) \in \mathcal{X}$, initialize $Q_i(0) = \max\{0, -g_i(x(-1))\}$ for all $i \in \{1, 2, \ldots, k\}$, and formally define $\overline{x}(0) = 0 \in \mathbb{R}^n$. Every slot $t \in \{0, 1, 2, \ldots\}$ do

- Choose $x(t) \in \mathcal{X}$ to minimize the expression:[6]

$$f(x(t)) + \sum_{i=1}^{k} \left(Q_i(t) + g_i(x(t-1)\right) g_i(x(t)) + \alpha||x(t) - x(t-1)||^2 \tag{24}$$

- Update virtual queues for each $i \in \{1, 2, \ldots, k\}$ via:

$$Q_i(t+1) = \max\{Q_i(t) + g_i(x(t)), -g_i(x(t))\} \tag{25}$$

- Update the time average vector $\overline{x}(t) \in \mathbb{R}^n$ via:

$$\overline{x}(t+1) = \left(\frac{t}{t+1}\right)\overline{x}(t) + \left(\frac{1}{t+1}\right)x(t)$$

Compared with the dual subgradient algorithm described in Section 2.4, the enhanced algorithm updates $x(t)$ by minimizing a modified expression that introduces an additional quadratic term $\alpha||x(t) - x(t-1)||^2$ and that changes the coefficient of each $g_i(x(t))$ from $Q_i(t)$ to $Q_i(t) + g_i(x(t-1))$. In addition, the enhanced algorithm modifies the virtual queue update equation by taking a max with $-g_i(x(t))$ instead of 0.[7]

[6] This seeks to minimize a continuous strongly convex function over the closed convex set $\mathcal{X}$, and so a unique minimizer exists.

[7] One immediate reason for this modification is to ensure the coefficient of each $g_i(x(t))$ in (24) is non-negative so that the algorithm chooses $x(t) \in \mathcal{X}$ to minimize a convex expression.

3.3 Virtual queue properties

The following lemmas specify basic properties of the modified virtual queue update equation (25). For simplicity, only the first lemma is proven (the remaining proofs are found in [14]).

Lemma 3. *(Modified virtual queue) Consider any sequence of decisions $x(t) \in \mathcal{X}$ for $t \in \{0, 1, 2, \ldots\}$. Under the modified virtual queue* (25) *we have for all positive integers T:*

$$\sum_{t=0}^{T-1} g_i(x(t)) \leq Q_i(T) - Q_i(0) \quad \forall i \in \{1, \ldots, k\}$$

and so by Jensen's inequality

$$g_i(\overline{x}(T)) \leq \frac{Q_i(T) - Q_i(0)}{T} \tag{26}$$

Proof. Fix $i \in \{1, \ldots, k\}$ and fix $t \in \{0, 1, 2, \ldots\}$. Then

$$\begin{aligned} Q_i(t+1) &= \max\{Q_i(t) + g_i(x(t)), -g_i(x(t-1))\} \\ &\geq Q_i(t) + g_i(x(t)) \end{aligned}$$

and so

$$Q_i(t+1) - Q_i(t) \geq g_i(x(t))$$

Summing over $t \in \{0, 1, \ldots, T-1\}$ gives the result. □

Lemma 4. *The enhanced algorithm ensures*

1. *$Q_i(t) \geq 0$ for all $i \in \{1, 2, \ldots, k\}$ and all $t \in \{0, 1, 2, \ldots\}$.*
2. *$Q_i(t) + g_i(x(t-1)) \geq 0$ for all $i \in \{1, 2 \ldots, k\}$ and all $t \in \{0, 1, 2, \ldots\}$.*
3. *$||Q(0)|| \leq ||g(x(-1))||$ and $||Q(t)|| \geq ||g(x(t-1))||$ for all $t \in \{1, 2, \ldots\}$.*

Proof. This lemma follows immediately from the virtual queue update equation. See [14]. □

Using the same Lyapunov drift $\Delta(t) = L(t+1) - L(t)$ with $L(t) = \frac{1}{2}||Q(t)||^2$ as before, we have the following new drift inequality under update rule (25):

Lemma 5. *(Drift inequality) Consider the update rule* (25) *under any sequence of decision vectors* $x(t) \in \mathcal{X}$ *for* $t \in \{0, 1, 2, \ldots\}$. *Then,*

$$\Delta(t) \leq ||g(x(t))||^2 + \sum_{i=1}^{k} Q_i(t) g_i(x(t)), \forall t \in \{0, 1, 2, \ldots\} \tag{27}$$

Proof. See [14]. □

At first sight, the above drift inequality under update equation (25) may look even looser when compared with the drift inequality under (6) shown in Lemma 2. Recall that the B constant in Lemma 2 is an upper bound of $\frac{1}{2}||g(x(t))||^2$ under the compactness assumption of set $\mathcal{X}$. However, the dual subgradient algorithm updates its primal variables $x(t)$ without taking the B term into consideration and hence suffers from a slow $O(1/\epsilon^2)$ convergence time. The enhanced algorithm modifies the $x(t)$ update and the $Q(t)$ update together such that they jointly cancel the effect of $||g(x(t))||^2$ to achieve a faster $O(1/\epsilon)$ convergence time. As a by-product, the enhanced algorithm does require boundedness of the set $\mathcal{X}$.

3.4 Quadratic pushback

The prox term of our enhanced algorithm turns our *ordinary* convex objective function $f(x)$ into a modified function $f(x) + \alpha||x - x(t-1)||^2$ that has a special property called *strong convexity*.

Definition 1. *Let* $\mathcal{X} \subseteq \mathbb{R}^n$ *be a convex set. Fix* $c > 0$. *A function* $h : \mathcal{X} \to \mathbb{R}$ *is a* ***c-strongly convex function*** *if the function* $h(x) - \frac{c}{2}||x||^2$ *is a convex function over* $\mathcal{X}$. *A function* h *is* ***strongly convex*** *if it is c-strongly convex for some* $c > 0$.

It can be shown that every strongly convex function is also a convex function. The functions $f, g_1, \ldots, g_k$ in our convex program (1)-(3) are convex but not necessarily strongly convex. However, it is easy to show that if $f : \mathcal{X} \to \mathbb{R}$ is

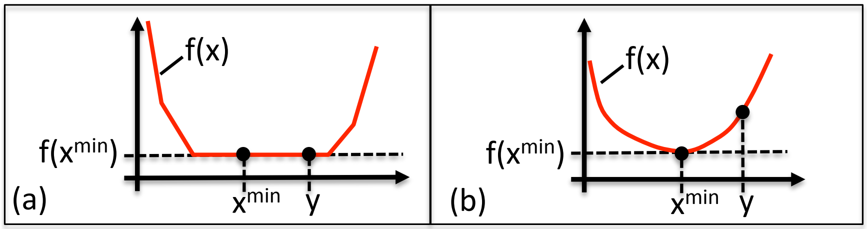

Figure 2. (a) A function with non-unique minima; (b) A strongly convex function with unique minimizer x^{min}. The value of $f(y)$ grows quadratically in the distance between y and x^{min}, which gives rise to the *quadratic pushback lemma* (Lemma 6).

any convex function and if $c > 0$, then for any $z \in \mathbb{R}^n$ the function $h : \mathcal{X} \to \mathbb{R}$ defined by

$$h(x) = f(x) + \frac{c}{2}||x - z||^2$$

is a c-strongly convex function. Indeed,

$$\begin{aligned} h(x) - \frac{c}{2}||x||^2 &= f(x) + \frac{c}{2}||x - z||^2 - \frac{c}{2}||x||^2 \\ &= f(x) + \underbrace{cx^T z + \frac{c}{2}||z||^2}_{\text{affine in } x} \end{aligned}$$

and the sum of a convex function with an affine function is convex.

Strongly convex functions are important because of their *quadratic pushback property*. Specifically, suppose x^{min} is a minimizer of a function $h : \mathcal{X} \to \mathbb{R}$. By definition of "minimizer" we have

$$h(x^{min}) \leq h(y) \quad \forall y \in \mathcal{X}$$

The next lemma shows that this inequality can be significantly improved when h is a strongly convex function (see Fig. 2).

Lemma 6. *(Quadratic pushback) Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set, let $h : \mathcal{X} \to \mathbb{R}$ be a c-strongly convex function for some $c > 0$, and let x^{min} be a minimizer of*

h over the set $\mathcal{X}$. Then

$$h(x^{min}) \leq h(y) - \underbrace{\frac{c}{2}||x^{min} - y||^2}_{pushback} \quad \forall y \in \mathcal{X}$$

Proof. Fix $y \in \mathcal{X}$. The function $h(x) - \frac{c}{2}||x||^2$ is convex over $x \in \mathcal{X}$, and so the function $r(x) = h(x) - \frac{c}{2}||x - x^{min}||^2$ is also convex over $x \in \mathcal{X}$. Thus, for any $p \in (0,1)$ we have

$$pr(y) + (1-p)r(x^{min}) \geq r(py + (1-p)x^{min})$$

Substituting the definition of $r(x)$ into this inequality gives

$$\begin{aligned}
&p[h(y) - \frac{c}{2}||y - x^{min}||^2] + (1-p)h(x^{min}) \\
&\geq h(py + (1-p)x^{min}) - \frac{c}{2}||py + (1-p)x^{min} - x^{min}||^2 \\
&= h(py + (1-p)x^{min}) - \frac{cp^2}{2}||y - x^{min}||^2 \\
&\overset{(a)}{\geq} h(x^{min}) - \frac{cp^2}{2}||y - x^{min}||^2
\end{aligned}$$

where (a) holds because $py + (1-p)x^{min} \in \mathcal{X}$ and x^{min} minimizes h over the set $\mathcal{X}$. Removing the common term $h(x^{min})$ from both sides of the above inequality and diving both sides by p gives

$$h(y) - \frac{c}{2}||y - x^{min}||^2 - h(x^{min}) \geq -\frac{cp}{2}||y - x^{min}||^2$$

The above inequality holds for all $p \in (0,1)$. Taking a limit as $p \to 0$ proves the result. □

3.5 DPP bound for the enhanced algorithm

Although the enhanced algorithm does not directly minimize an upper bound on the drift-plus-penalty expression as the dual subgradient algorithm does, the next lemma shows that the enhanced algorithm still provides an upper bound on the drift-plus-penalty expression.

Lemma 7. *Suppose convex program* (1)-(3) *satisfies Assumption 2 and has an optimal solution $x^* \in \mathcal{X}$. If $\alpha \geq \frac{1}{2}\beta^2$ in the enhanced algorithm, then for all $t \geq 0$, we have*

$$\begin{aligned}
&\Delta(t) + f(x(t)) \\
\leq & f(x^*) + \alpha||x^* - x(t-1)||^2 - \alpha||x^* - x(t)||^2 \\
&+ \frac{1}{2}||g(x(t))||^2 - \frac{1}{2}||g(x(t-1))||^2,
\end{aligned}$$

where β is defined in Assumption 2.

Proof. Fix $t \geq 0$. Note that Lemma 4 implies that $Q_i(t) + g_i(x(t-1))$ is nonnegative for all $i \in \{1, 2, \ldots, k\}$. Hence, the expression

$$f(x(t)) + \sum_{i=1}^{k} \left(Q_i(t) + g_i(x(t-1)\right) g_i(x(t)) + \alpha||x(t) - x(t-1)||^2$$

is (2α)-strongly convex with respect to $x(t)$. Since the enhanced algorithm chooses $x(t) \in \mathcal{X}$ to minimize the above strongly convex expression, by the quadratic pushback lemma (Lemma 6), we have

$$\begin{aligned}
&f(x(t)) + \sum_{i=1}^{k} \left(Q_i(t) + g_i(x(t-1))\right) g_i(x(t)) + \alpha||x(t) - x(t-1)||^2 \\
\leq & f(x^*) + \underbrace{\sum_{i=1}^{k} \left(Q_i(t) + g_i(x(t-1))\right) g_i(x^*)}_{\leq 0} + \alpha||x^* - x(t-1)||^2 - \alpha||x^* - x(t)||^2 \\
\overset{(a)}{\leq} & f(x^*) + \alpha||x^* - x(t-1)||^2 - \alpha||x^* - x(t)||^2,
\end{aligned} \tag{28}$$

where (a) follows by using the fact that $g_i(\mathbf{x}^*) \leq 0$ for all $i \in \{1, 2, \ldots, k\}$ and $Q_i(t) + g_i(\mathbf{x}(t-1)) \geq 0$ (i.e., part 2 in Lemma 4) to eliminate the term marked by an underbrace.

Recall that $\sum_{i=1}^{k} u_i v_i = \frac{1}{2}||u||^2 + \frac{1}{2}||v||^2 - \frac{1}{2}||u - v||^2$ for any two vectors u, v of the same length. We have

$$\sum_{i=1}^{k} g_i(x(t-1))g_i(x(t)) = \frac{1}{2}||g(x(t-1))||^2 + \frac{1}{2}||g(x(t))||^2 - \frac{1}{2}||g(x(t)) - g(x(t-1))||^2. \tag{29}$$

Substituting (29) into (28) and rearranging terms yields

$$
\begin{aligned}
&f(x(t))+\sum_{i=1}^{k}Q_i(t)g_i(x(t))\\
\leq& f(x^*)+\alpha||x^*-x(t-1)||^2-\alpha||x^*-x(t)||^2-\alpha||x(t)-x(t-1)||^2\\
&+\frac{1}{2}||g(x(t-1))-g(x(t))||^2-\frac{1}{2}||g(x(t-1))||^2-\frac{1}{2}||g(x(t))||^2\\
\overset{(a)}{\leq}& f(x^*)+\alpha||x^*-x(t-1)||^2-\alpha||x^*-x(t)||^2+(\frac{1}{2}\beta^2-\alpha)||x(t)-x(t-1)||^2\\
&-\frac{1}{2}||g(x(t-1))||^2-\frac{1}{2}||g(x(t))||^2\\
\overset{(b)}{\leq}& f(x^*)+\alpha||x^*-x(t-1)||^2-\alpha||x^*-x(t)||^2-\frac{1}{2}||g(x(t-1))||^2-\frac{1}{2}||g(x(t))||^2,
\end{aligned}
$$

where (a) follows from the fact that $||g(x(t-1))-g(x(t))|| \leq \beta||x(t)-x(t-1)||$, which further follows from the assumption that $g(x)$ is Lipschitz continuous with parameter β; and (b) follows from the fact $\alpha \geq \frac{1}{2}\beta^2$.

Summing (27) with the above inequality yields

$$
\begin{aligned}
&\Delta(t)+f(x(t))\\
\leq& f(x^*)+\alpha||x^*-x(t-1)||^2-\alpha||x^*-x(t)||^2+\frac{1}{2}||g(x(t))||^2-\frac{1}{2}||g(x(t-1))||^2.
\end{aligned}
$$

□

3.6 Performance theorem

Theorem 3. *Suppose convex program* (1)-(3) *satisfies Assumptions 1-2 and has an optimal solution $x^* \in \mathcal{X}$. If the enhanced algorithm uses $\alpha > \frac{1}{2}\beta^2$, then for all $T \in \{1,2,3,\ldots\}$ we have $\overline{x}(T) \in \mathcal{X}$ and*

$$
f\left(\overline{x}(T)\right) \leq f(x^*)+\frac{\alpha}{T}||x^*-x(-1)||^2 \tag{30}
$$

$$
g_i\left(\overline{x}(T)\right) \leq \frac{1}{T}||\mu||+\frac{1}{T}\sqrt{||\mu||^2+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2} \quad ,\forall i \in \{1,\ldots,k\} \tag{31}
$$

where μ is defined in Assumption 1, $||\mu|| = \sqrt{\sum_{i=1}^{k}\mu_i^2}$, and β is defined in Assumption 2. In particular, deviation from optimality decays like $O(1/T)$. For any desired $\epsilon > 0$, the enhanced algorithm produces an $O(\epsilon)$-optimal solution whenever the number of iterations satisfies $T \geq 1/\epsilon$.

Proof. (Theorem 3, inequality (30)) Fix T as a positive integer. By Lemma 7, for all $t \in \{0, 1, 2, \ldots\}$ we have

$$\begin{aligned}&\Delta(t) + f(x(t))\\ \leq& f(x^*) + \alpha||x^* - x(t-1)||^2 - \alpha||x^* - x(t)||^2 + \frac{1}{2}||g(x(t))||^2 - \frac{1}{2}||g(x(t-1))||^2.\end{aligned}$$

Substituting $\Delta(t) = L(t+1) - L(t)$ and summing over $t \in \{0, 1, \ldots, T-1\}$ yields

$$\begin{aligned}&L(T) - L(0) + \sum_{t=0}^{T-1} f(x(t))\\ \leq& Tf(x^*) + \alpha||x^* - x(-1)||^2 - \alpha||x^* - x(T-1)||^2 + \frac{1}{2}||g(x(T-1))||^2 - \frac{1}{2}||g(x(-1))||^2\end{aligned}$$

Substituting $L(T) = \frac{1}{2}||Q(T)||^2$, moving $L(0)$ to the right side, and noting that $L(0) = \frac{1}{2}||Q(0)||^2 \leq \frac{1}{2}||g(x(-1))||^2$ by Lemma 4 gives:

$$\frac{1}{2}||Q(T)||^2 + \sum_{t=0}^{T-1} f(x(t)) \leq Tf(x^*) + \alpha||x^* - x(-1)||^2 - \alpha||x^* - x(T-1)||^2 + \frac{1}{2}||g(x(T-1))||^2 \tag{32}$$

To prove (30), rearranging (32) and dividing by T gives

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1} f(x(t)) \leq& f(\mathbf{x}^*) + \frac{1}{T}\left(\alpha||x^* - x(-1)||^2 - \alpha||x^* - x(T-1)||^2 + \frac{1}{2}||g(x(T-1))||^2 - \frac{1}{2}||Q(T)||^2\right)\\ \leq& f(\mathbf{x}^*) + \frac{1}{T}\alpha||x^* - x(-1)||^2\end{aligned}$$

where the last inequality follows because $||Q(T)|| \geq ||g(x(T-1))||$ for positive T by Lemma 4. Then, (30) follows by applying Jensen's equality for the convex function f. □

Proof. (Theorem 3, inequality (31)) To prove (31), rearranging (32) gives

$$\begin{aligned}||Q(T)||^2 \leq& 2\sum_{t=0}^{T-1}\left(f(x^*) - f(x(t))\right) + 2\alpha||x^* - x(-1)||^2 \underbrace{-2\alpha||x^* - x(T-1)||^2 + ||g(x(T-1))||^2}_{\leq \frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2}\\ \leq& 2\sum_{t=0}^{T-1}\left(f(x^*) - f(x(t))\right) + 2\alpha||x^* - x(-1)||^2 + \frac{2\alpha}{2\alpha - \beta^2}||g(x^*)||^2 \end{aligned} \tag{33}$$

To see the term marked by an underbrace in the first step is less than or equal to $\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2$, we note that

$$\begin{aligned}
&-2\alpha||x^*-x(T-1)||^2+||g(x(T-1))||^2\\
=&-2\alpha||x^*-x(T-1)||^2+||g(x(T-1))-g(x^*)+g(x^*)||^2\\
\leq&-2\alpha||x^*-x(T-1)||^2+||g(x(T-1))-g(x^*)||^2+2||g(x^*)||\cdot||g(x(T-1))-g(x^*)||+||g(x^*)||^2\\
\overset{(a)}{\leq}&-2\alpha||x^*-x(T-1)||^2+\beta^2||x^*-x(T-1)||^2+2\beta||g(x^*)||\cdot||x^*-x(T-1)||+||g(x^*)||^2\\
=&-(2\alpha-\beta^2)\Big(||x^*-x(T-1)||-\frac{\beta}{2\alpha-\beta^2}||g(x^*)||\Big)^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2\\
\overset{(b)}{\leq}&\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2
\end{aligned}$$

where (a) follows because $g(x)$ is Lipschitz continuous with parameter β by Assumption 2; and (b) follows by $\alpha>\frac{1}{2}\beta^2$.

Since $x(t)\in\mathcal{X}$ for all t, by (13) from Assumption 1, we have

$$\sum_{t=0}^{T-1}(f(x^*)-f(x(t)))\leq\sum_{t=0}^{T-1}\sum_{i=1}^{k}\mu_i g_i(x(t)) \tag{34}$$

Substituting (34) into (33) gives

$$\begin{aligned}
||Q(T)||^2\leq&2\sum_{t=0}^{T-1}\sum_{i=1}^{k}\mu_i g_i(x(t))+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2\\
=&2\sum_{i=1}^{k}\mu_i\sum_{t=0}^{T-1}g_i(x(t))+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2\\
\overset{(a)}{\leq}&2\sum_{i=1}^{k}\mu_i(Q_i(T)-Q_i(0))+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2\\
\overset{(b)}{\leq}&2\sum_{i=1}^{k}\mu_i Q_i(T)+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2\\
\overset{(c)}{\leq}&2||\mu||\cdot||Q(T)||+2\alpha||x^*-x(-1)||^2+\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2
\end{aligned}$$

where (a) follows from the modified virtual queue lemma (Lemma 3) and the fact that $\mu_i\geq 0$ for all i; (b) follows from $Q_i(0)\geq 0$ for all i by Lemma 4 and the fact that $\mu_i\geq 0$ for all i; and (c) follows from Cauchy-Schwarz inequality.

Now we again obtain a quadratic inequality in terms of $||Q(T)||$ as we do in the proof of Theorem 2. Define $y=||Q(T)||$, $b=-2||\mu||$, $c=-2\alpha||x^*-x(-1)||^2-\frac{2\alpha}{2\alpha-\beta^2}||g(x^*)||^2$. The above inequality reduces to the quadratic inequality $y^2 + by + c \leq 0$ and so

$$y \leq \frac{-b + \sqrt{b^2 - 4c}}{2} = ||\mu|| + \sqrt{||\mu||^2 + 2\alpha||x^* - x(-1)||^2 + \frac{2\alpha}{2\alpha - \beta^2}||g(x^*)||^2}$$

Since $y = ||Q(T)||$, for each $i \in \{1, \dots, k\}$ we have

$$\frac{Q_i(T)}{T} \leq \frac{||Q(T)||}{T} \leq \frac{1}{T}||\mu|| + \frac{1}{T}\sqrt{||\mu||^2 + 2\alpha||x^* - x(-1)||^2 + \frac{2\alpha}{2\alpha - \beta^2}||g(x^*)||^2}$$

The result of (31) follows by application of the modified virtual queue lemma (Lemma 3) and noting that $Q_i(0) \geq 0$ by Lemma 4. □

3.7 Enhanced algorithm exercises

Exercise 6. *(Separable problems with the enhanced algorithm) Redo Exercise 4 using the enhanced algorithm.*

Exercise 7. *(Enhanced flow control) Consider the 2-link network with 4 traffic flows shown in Fig. 1 and the corresponding convex program given in Exercise 5.*[8]

a) As in Exercise 5, write the corresponding convex program in the form (1)-(3). *Hint: use $f(x) = -\sum_{i=1}^{4} w_i \log(1 + x_i)$.*

b) Suppose we implement the enhanced algorithm with parameter $\alpha > 0$. Specify the update equations for the two virtual queues $Q_1(t)$ and $Q_2(t)$.

c) Continuing part (b), specify the (separable) decisions $x_i(t) \in [0, 1]$ for each $t \in \{0, 1, 2, \dots\}$ and each $i \in \{1, \dots, 4\}$.

Exercise 8. *(Multipath network flow control) Consider the 3-link network with 3 traffic flows shown in Fig. 3. Note that there are 2 parallel paths, with capacity C_1 and C_2, between the first and the second nodes. Since the second traffic flow has two parallel paths, its overall traffic flow rate is given by $x_{21} + x_{22}$ if x_{21} and x_{22} are respective path rates. The network utility maximization in such a*

[8] As discussed in the footnote to Exercise 5, the convex program for this network flow problem is already *strongly convex* and so the results of [5] [15] ensure that the basic dual subgradient algorithm also enjoys a fast $O(1/\epsilon)$ convergence in this case.

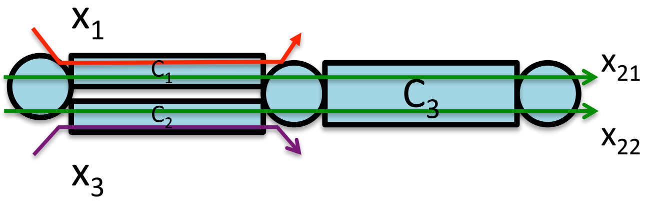

Figure 3. The 3-link network with 3 traffic flows, one of which has 2 paths, for the network utility maximization problem of Exercise 8.

multipath scenario is given by

$$
\begin{aligned}
\textit{Maximize:}\quad & w_1 \log(1+x_1) + w_2 \log(1+x_{21}+x_{22}) + w_3 \log(1+x_3) \\
\textit{Subject to:}\quad & x_1 + x_{21} \leq C_1 \\
& x_{22} + x_3 \leq C_2 \\
& x_{21} + x_{22} \leq C_3 \\
& (x_1, x_{21}, x_{22}, x_3) \in \mathcal{X}
\end{aligned}
$$

where $\mathcal{X} = [0,1]^4$ *and* $w_1, \ldots, w_3$ *are given positive constants.*[9]

a) As in Exercise 5, write the corresponding convex program in the form (1)-(3).

b) Suppose we implement the enhanced algorithm with parameter $\alpha > 0$. *Specify the update equations for the three virtual queues* $Q_1(t)$, $Q_2(t)$ *and* $Q_2(t)$.

c) Continuing part (b), specify the (separable) decisions $x_1(t), x_{21}(t), x_{22}(t), x_3(t) \in [0,1]$ *for each* $t \in \{0,1,2,\ldots\}$.

3.8 Notes

- The primal update for $x(t) \in \mathcal{X}$ requires to solve the set constrained minimization problem (24). If the $f, g_1, \ldots, g_k$ functions are smooth, the work [16] gives a primal-dual version of the enhanced algorithm by

[9] Note that $\log(1+x_{21}+x_{22})$ is not strongly concave with respect to the vector (x_{21}, x_{22}) even though $\log(1+x)$ is strongly concave with respect to x. Thus, the basic dual subgradient algorithm can only have a slow $O(1/\epsilon^2)$ convergence in this case.

replacing the set constrained minimization step with a projection onto $\mathcal{X}$ such that the fast $O(1/\epsilon)$ convergence time is preserved. The per-iteration complexity of this primal-dual version is similar to that of the classical primal-dual subgradient method studied in [7] which has slower $O(1/\epsilon^2)$ convergence.

- In the special case when all g_i functions in convex program (1)-(3) are linear (such as for the convex program of Exercises 5 and 7), we can change the virtual queue update equation (25) to $Q_i(t+1) = Q_i(t) + g_i(x(t))$ while keeping the $x(t)$ update unchanged. In [18], this variant is used to develop a new joint rate control and routing strategy for multi-hop data networks that yields queues with $O(1)$ size and with a utility optimality gap that decays like $O(1/t)$ (and so convergence time to an ϵ-optimal solution is $O(1/\epsilon)$), which improves upon the prior state-of-the-art for data networks. An interesting observation here is that such a variant is similar to the proximal Jacobian ADMM algorithm, also known as linearized ADMM, studied in [3]. The proximal Jacobian ADMM algorithm was previously shown to have a weak type of convergence in the sense that $||x(t+1) - x(t)||^2 = o(1/t)$, although this property in general says nothing about convergence towards a solution of the convex program or about optimality gaps for the objective and constraint functions (i.e., the f and g_i functions). The analysis of this section (developed in [14] and considered for the linear case in [18]) can prove the $O(1/t)$ convergence of the proximal Jacobian ADMM in terms of both objective optimality and constraint violations.

Acknowledgement

This work was supported in part by the NSF grant CCF-1718477.

References

[1] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, Convex analysis and optimization, Boston: Athena Scientific, 2003.

[2] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge University Press, 2004.

[3] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin, Parallel multi-block ADMM with $o(1/k)$ convergence, Journal of Scientific Computing **71** (2017), no. 2, 712–736.

[4] L. Georgiadis, M. J. Neely, and L. Tassiulas, Resource allocation and cross-layer control in wireless networks, Foundations and Trends in Networking **vol. 1, no. 1, pp. 1-149** (2006).

[5] 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).

[6] A. Nedic and A. Ozdaglar, Approximate primal solutions and rate analysis for dual subgradient methods, SIAM Journal on Optimization **vol. 19, no. 4** (2009).

[7] Angelia Nedić and Asuman Ozdaglar, Subgradient methods for saddle-point problems, Journal of Optimization Theory and Applications **142** (2009), no. 1, 205–228.

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

[9] ———, Stochastic network optimization with application to communication and queueing systems, Morgan & Claypool, 2010.

[10] ———, A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs, ArXiv technical report, arXiv:1412.0791v1 (Dec. 2014).

[11] ———, Distributed and secure computation of convex programs over a network of connected processors, DCDIS Conf., Guelph, Ontario (July 2005).

[12] Y. Nesterov, Introductory lectures on convex optimization: A basic course, Kluwer Academic Publishers, Boston, 2004.

[13] S. Supittayapornpong, L. Huang, and M. J. Neely, Time-average optimization with nonconvex decision set and its convergence, IEEE Transactions on Automatic Control **62** (2017), no. 8, 42024208.

[14] 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 **27** (2017), no. 2, 759–783.

[15] ———, On the convergence time of dual subgradient methods for strongly convex programs, IEEE Transactions on Automatic Control **63** (2018), no. 4, 1105–1112.

[16] ———, A primal-dual type algorithm with the $O(1/t)$ convergence rate for large scale constrained convex programs, Proc. IEEE Conference on Decision and Control (Las Vegas, USA), Dec. 2016.

[17] H. Yu, M. J. Neely, and X. Wei, Online convex optimization with stochastic constraints, Proc. 31st Conf. on Neural Information Processing Systems (NIPS) (2017).

[18] Hao Yu and Michael J. Neely, A new backpressure algorithm for joint rate control and routing with vanishing utility optimality gaps and finite queue lengths, IEEE/ACM Transactions on Networking **26** (2018), no. 4, 1605–1618.