Distributed Utility Maximization for Network Coding Based Multicasting: A Shortest Path Approach

Yunnan Wu, *Member, IEEE*, Sun-Yuan Kung, *Fellow, IEEE*.

***Abstract*— One central issue in practically deploying network coding is the adaptive and economic allocation of network resource. We cast this as an optimization, where the *net-utility* – the difference between a utility derived from the attainable multicast throughput and the total cost of resource provisioning – is maximized. By employing the MAX of flows characterization of the admissible rate region for multicasting, this paper gives a novel reformulation of the optimization problem that has a separable structure. The Lagrangian relaxation method is applied to decompose the problem into subproblems involving one destination each. Our specific formulation of the primal problem results in two key properties. First, the resulting subproblem after decomposition amounts to the problem of finding a shortest path from the source to each destination. Second, assuming the net-utility function is strictly concave, our proposed method enables a near-optimal primal variable to be uniquely recovered from a near-optimal dual variable.**

A numerical robustness analysis of the primal recovery method is also conducted. For ill-conditioned problems that arise, for instance, when the cost functions are linear, we propose to use the proximal method, which solves a sequence of well-conditioned problems obtained from the original problem by adding quadratic regularization terms. Furthermore, the simulation results confirm the numerical robustness of the proposed algorithms. Finally, the proximal method and the dual subgradient method can be naturally extended to provide an effective solution for applications with multiple multicast sessions.

***Index Terms*— Multicast, network coding, distributed optimization, dual, subgradient, shortest path.**

I. Introduction

Consider a network formed by a collection of lossless links, which can naturally be represented by a directed graph $G = (V, E)$, where the vertex set V and the edge set E denote the nodes and links, respectively. We examine information multicasting in such a network, where a source node s is transmitting common information to a set of destination nodes T. Suppose the bit-rate constraints on the links are specified by a vector $\boldsymbol{c}$ of length $|E|$; the capacity for link $vw \in E$ is denoted by c_{vw}. Given V, E, $\boldsymbol{c}$, s, and T, the *multicast capacity* refers to the maximum multicast throughput.

An upper bound of the multicast capacity can be established by examining the *cuts* that separate s from any destination $t \in T$. For $t \in T$, an s–t cut $(U, \overline{U})$ refers to a partition of the nodes $V = U + \overline{U}$, with $s \in U$, $t \in \overline{U}$. The *capacity* of the cut refers to the sum capacity of edges going from U to $\overline{U}$. An s–t cut with minimum capacity is called a *minimum s–t cut*. Let $\rho_{s,t}(\boldsymbol{c})$ denote the capacity of a minimum s–t cut for graph (V, E) with link capacities $\boldsymbol{c}$. Then

$$\min_{t \in T} \rho_{s,t}(\boldsymbol{c}) \tag{1}$$

is an upper bound of the multicast capacity since the capacity of any s–t cut is an upper bound on the rate at which information can be transferred from s to t.

In today's practical networks, end-to-end information delivery is done by *routing*, i.e., having intermediate nodes store and forward packets. For multicasting, Ahlswede et al. showed in [1] that the upper bound (1) cannot be achieved by routing in general, but it *can* be achieved, via more powerful techniques called *network coding*. Hence, (1) is the multicast capacity. Network coding generalizes routing by allowing a node to mix information, i.e., produce output data by computing certain functions of the data it received. Ahlswede et al.'s result was established via information theoretic arguments. Subsequently, important progress has been made regarding the low-complexity construction of network codes. Li et al. [2] showed that the maximum multicast capacity can be achieved by performing linear network coding. Ho et al. [3], Jaggi et al. [4] and Sanders et al. [5] showed that random linear network coding over a sufficiently large finite field can (asymptotically) achieve the multicast capacity. Following these constructive theoretical results about network coding, Chou et al. [6] proposed a practical scheme for performing network coding in real packet networks. In the scheme, each node maintains a buffer that stores the incoming packets as they arrive. Whenever the node is allowed to transmit a packet, a *mixture* packet is formed by combining the packets in the buffer using random coefficients. For each packet, the *global coding vector*, which indicates how this packet relates to the source packets, is recorded as header information, so as to describe the composition of the packet.

In the practical network coding scheme, encoding amounts to forming random mixture of packets; this is decentralized and simple to implement. However, another essential element is a distributed scheme for properly allocating bit-rate resource at each link for each multicast session in a shared network. Following a popular approach in economics theory, we cast this problem as the maximization of a *net-utility* function

$$U_{\text{net}}(r, \boldsymbol{g}) \triangleq U(r) - \sum_{vw \in E} p_{vw}(g_{vw}). \tag{2}$$

Manuscript received Sept. 1, 2005, revised Apr. 26, 2006.

Yunnan Wu is with Microsoft Research, One Microsoft Way, Redmond, WA, 98052. yunnanwu@microsoft.com Tel:425-706-5907.

Sun-Yuan Kung is with the Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544. kung@princeton.edu.

Here $U(r)$ represents the (raw) utility when an end-to-end throughput r is provided. The cost function p_{vw} associated with link vw maps the consumed bit-rate g_{vw} to the charge.

The critical constraint of such a maximization is that throughput r must be attainable using the resources g_{vw}. Let $\boldsymbol{g}$ be a length-$|E|$ vector collectively representing g_{vw}. For network coding based multicasting, this relation is characterized by $r \le \min_{t \in T} \rho_{s,t}(\boldsymbol{g})$.

More formally, we consider the following formulation

$$\begin{aligned} U^*_{\text{net}} \triangleq \max \quad & U_{\text{net}}(r, \boldsymbol{g}) \\ \text{subject to:} \quad & r \le \min_{t \in T} \rho_{s,t}(\boldsymbol{g}), \\ & r \in [r_0, r_1], \\ & \boldsymbol{0} \le \boldsymbol{g} \le \boldsymbol{c}, \end{aligned} \tag{3}$$

The constraint $r \in [r_0, r_1]$ models that the application may have a desired range of throughput.[1]

The objective of this paper is to design efficient distributed algorithms for finding an optimal solution $(r^*, \boldsymbol{g}^*)$ of (3). The distributed algorithms should, hopefully, incur low extra communication overhead and be adaptive to network dynamics.

The obtained r^* and $\boldsymbol{g}^*$ will be used as the operating parameters of the practical network coding system. Specifically, the source node will set the end-to-end multicast rate to a value slightly lower than r^*. Each link vw in the network will generate random mixture packets at a rate around g^*_{vw}. Then, since the practical network coding scheme based on random linear coding can achieve throughput close to the capacity, with high probability the destinations will be able to recover the source messages.

A. Overview of proposed approach

1) Formulation with a separable structure: The constraint $r \le \min_{t \in T} \rho_{s,t}(\boldsymbol{g})$ imposes that the rate r must be attained. Thanks to the well-known Max-Flow-Min-Cut theorem, this nonlinear constraint can be reduced to a system of linear inequalities. Specifically, $r \le \min_{t \in T} \rho_{s,t}(\boldsymbol{g})$ if and only if there exists a flow vector $\boldsymbol{f}_t \le \boldsymbol{g}$ from s to each destination $t \in T$ with rate r. When expressed in terms of flow variables, this leads to an optimization formulation with not only linear constraints but also a *separable structure*, in which the flow variables are coupled only through the constraints $\boldsymbol{f}_t \le \boldsymbol{g}$.

2) Lagrangian relaxation and dual subgradient method: The separable structure allows the problem to be decomposed into a collection of subproblems, each involving only a single destination. In optimization theory, a well-established decomposition approach to is by relaxing the coupling constraints with Lagrangian multipliers, or dual variables. In lieu of the original problem, the dual problem is then solved via, say, the subgradient method [7] (see also Appendix.A of this paper). Although the dual subgradient method represents a classical approach (see, e.g., Bertsekas and Tsitsiklis [8]), the construction of dual formulations is in general non-unique and an effective construction is often necessarily problem-specific.

Our specific dual subgradient algorithm (Algorithm 1) has two key properties. First, the subproblem involving the flow variable $\boldsymbol{f}_t$ for each destination t amounts to the problem of finding a shortest path from s to t with the dual variables $\boldsymbol{\varepsilon}_t$ being the path lengths. The shortest path problem has been well understood. In particular, the Bellman-Ford algorithm is a well-known distributed algorithm for the shortest-path problem.

Second, assuming the net-utility function is strictly concave, our proposed method outputs a sequence of primal variables $(r^{(k)}, \boldsymbol{g}^{(k)})$ converging to the optimal solution $(r^*, \boldsymbol{g}^*)$, provided that the dual iterates $\boldsymbol{\varepsilon}^{(k)}$ converge to an optimal solution $\boldsymbol{\varepsilon}^*$. The primal variables $(r^{(k)}, \boldsymbol{g}^{(k)})$ are determined from $\boldsymbol{\varepsilon}^{(k)}$ as the unique maximizers of the Lagrangian for $\boldsymbol{\varepsilon}^{(k)}$.

3) Proximal regularization for ill-conditioned problems: We show via some examples that when the net-utility function lacks sufficient degree of strict concavity, the generated primal solution $(r^{(k)}, \boldsymbol{g}^{(k)})$ may be far from optimal even if the associated dual variable $\boldsymbol{\varepsilon}^{(k)}$ is near optimal. The robustness of the proposed method is quantified.

For ill-conditioned problems, for instance, when the cost functions p_{vw} are linear, we propose to use the proximal method (see, e.g., the book [8] and the references therein) to regularize the problem. In lieu of the original problem, we solve a sequence of well-conditioned problems, obtained from the original problem by adding quadratic terms.

4) Generalization to multiple multicast sessions: We generalize the proposed approach to the case when there are multiple multicast sessions, and/or when there are flexibility in determining the supply of link bit-rates. In this context, the proximal method is again used in conjunction with the dual subgradient method to decompose the problem, and to generate a sequence of primal variables converging to an optimal solution.

B. Related works on dual subgradient methods

Dual subgradient methods have been applied to various formulations of network utility maximization problems. These include for example prior works on Internet flow control [9] and cross-layer optimizations via dual decomposition [10]–[14]. This work differentiates from these prior works in terms of the unique problem structure associated with network coding based multicasting.

Li and Li [15] applied the dual subgradient method to the dual linear program for computing the maximum multicast rate in an undirected graph. This results in a primal subgradient algorithm that iteratively adjusts the way the total capacity of each undirected edge is partitioned into two parts, one for each direction.

The work most closely related to this work is by Lun et al. [16]–[18]. Lun et al. proposed a dual subgradient method for the problem of minimizing a linear cost function for network coding based multicasting. This problem corresponds to a special case of (3). A fixed multicast rate is assumed, therefore $U(r)$ disappears. In addition, linear cost functions $p_{vw}(g_{vw})$ are considered in (3).

This work differs from the dual subgradient approach of [18] in several major aspects mainly because of the difference in formulating the primal and dual problems. First, in [18], the

[1] In this paper, $\boldsymbol{a} \le \boldsymbol{b}$ is in the element-wise sense.

subproblem after decomposition amounts to the minimum cost flow problem, whereas the subproblems here amounts to the (simpler) shortest path problem.[2] Second, in [18], the primal variables $\{\boldsymbol{f}_t\}$ cannot be uniquely determined directly from the dual vectors; to generate a converging primal sequence $\{\boldsymbol{f}_t^{(k)}\}$, Lun et al. [18] propose to use the method of Sherali and Choi [19]. In comparison, for linear cost problems, we propose to use the proximal method in conjunction with Algorithm 1. Thus Algorithm 1 is applied multiple times, and each application is to a maximization of a strictly concave net-utility function. Third, it is far from being straightforward to generalize the approach in [18] to cope with the case of flexible rates. This is because each subproblem in [18] is a minimum cost flow from s to each $t \in T$ with rate r. When r is a variable, the multiple subproblems become coupled since they need to operate on the same rate r.

Lun et al. [18] also considered strictly convex link cost functions and proposed a primal–dual algorithm. Since this primal–dual algorithm is of a different nature than the dual subgradient approach discussed herein, it is not reviewed here.

II. The Basic Approach

This section makes use of the following assumptions:

A1: The problem (3) has an optimal solution.

A2: $U : \mathbb{R}_+ \to \mathbb{R}$ is strictly concave and $\forall vw \in E$, $p_{vw} : \mathbb{R}_+ \to \mathbb{R}$ is strictly convex, where $\mathbb{R}_+$ denotes the set of nonnegative real numbers. This implies that $U_{\text{net}}(r, \boldsymbol{g})$ is strictly concave in $(r, \boldsymbol{g})$.

Note that A1 and A2 together implies that there is a unique optimal solution of (3), which is denoted by $(r^*, \boldsymbol{g}^*)$.

A. MAX of flows characterization of $r \le \min_{t\in T} \rho_{s,t}(\boldsymbol{g})$

Let us begin by expressing the constraint $r \le \min_{t\in T} \rho_{s,t}(\boldsymbol{g})$ via a set of linear constraints. This hinges upon the Max-Flow-Min-Cut Theorem in graph theory.

An *s–t flow* is a length-$|E|$ nonnegative vector $\boldsymbol{f}$ satisfying the *flow conservation constraint*:

$$\text{excess}_v(\boldsymbol{f}) = 0, \quad \forall v \in V - \{s,t\}. \tag{4}$$

where

$$\text{excess}_v(\boldsymbol{f}) \triangleq \sum_{u:\ uv\in E} f_{uv} - \sum_{w:\ vw\in E} f_{vw}, \tag{5}$$

is the *flow excess of* v, viz., the amount of incoming traffic less the amount of outgoing traffic for node v. An s–t flow $\boldsymbol{f}$ essentially prescribes several parallel paths, along which information can be routed from s to t. The flow excess at t is called the *value of the flow*, which corresponds to the communication rate that can be achieved by routing along the paths associated with $\boldsymbol{f}$.

Let $\mathcal{F}_{s,t}(r)$ denote the set of s–t flows with value r. Then $\boldsymbol{f} \in \mathcal{F}_{s,t}(r)$ if and only if

$$\boldsymbol{f} \ge \boldsymbol{0}, \tag{6}$$
$$\text{excess}_t(\boldsymbol{f}) = r, \tag{7}$$
$$\text{excess}_v(\boldsymbol{f}) = 0, \quad \forall v \in V - \{s,t\} \tag{8}$$

Note that the above inequalities are linear in $\boldsymbol{f}$ and r; for this reason, $\mathcal{F}_{s,t}(r)$ is called the *s–t flow polyhedron*. A useful property of $\mathcal{F}_{s,t}(r)$ is its linearity in r, i.e.,

$$\mathcal{F}_{s,t}(r) = r\mathcal{F}_{s,t}(1) \triangleq \{r\boldsymbol{f} | \boldsymbol{f} \in \mathcal{F}_{s,t}(1)\}. \tag{9}$$

The Max-Flow-Min-Cut Theorem says that for graph (V, E) with edge capacities $\boldsymbol{g}$, the minimum s–t cut capacity $\rho_{s,t}(\boldsymbol{g})$ is equal to the maximum s–t flow value. It follows then

$$r \le \rho_{s,t}(\boldsymbol{g}) \Longleftrightarrow \exists \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r),\ \boldsymbol{f}_t \le \boldsymbol{g}. \tag{10}$$

Since

$$r \le \min_{t\in T} \rho_{s,t}(\boldsymbol{g}) \Longleftrightarrow \exists \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r),\ \max_{t\in T} \boldsymbol{f}_t \le \boldsymbol{g}, \tag{11}$$

this is termed the *MAX of flows* characterization for the *admissible rate region of multicasting* [20] [21]. The admissible rate region of multicasting refers to set of $\boldsymbol{g}$ such that rate r can be supported. Just as a flow is the critical structure for unicast communication, a MAX of flows is the critical structure for network coding based multicasting.

Formulation (3) is equivalent to the following maximization with linear constraints (the optimization variables are stated below the optimization operator, e.g., $\max$).

Lemma 1 (Linear constraint primal formulation):

$$\begin{aligned} U^*_{\text{net}} = \max_{r,\boldsymbol{g},\boldsymbol{f}} \quad & U_{\text{net}}(r, \boldsymbol{g}) \\ \text{subject to:} \quad & \boldsymbol{f}_t \le \boldsymbol{g}, \quad \forall t \in T \\ & \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r), \quad \forall t \in T, \\ & r \in [r_0, r_1], \\ & \boldsymbol{0} \le \boldsymbol{g} \le \boldsymbol{c}. \end{aligned} \tag{12}$$

B. Dual subgradient iterations based on shortest paths

Introduce a vector of *dual variables* $\boldsymbol{\varepsilon}_t$ for each constraint $\boldsymbol{f}_t \le \boldsymbol{g}$. Then form the Lagrangian as follows

$$L(r, \boldsymbol{g}, \boldsymbol{f}, \boldsymbol{\varepsilon}) \triangleq U(r) - \sum_{vw\in E} p_{vw}(g_{vw}) - \sum_{t\in T} \boldsymbol{\varepsilon}_t^T(\boldsymbol{f}_t - \boldsymbol{g}). \tag{13}$$

Here $\boldsymbol{f}$ (resp. $\boldsymbol{\varepsilon}$) denote the vector formed by stacking the vectors $\{\boldsymbol{f}_t,\ t \in T\}$ (resp. $\{\varepsilon_t,\ t \in T\}$) together. In the following, $r, \boldsymbol{g}, \boldsymbol{f}$ will be referred to as the *primal variables*.

The *dual problem* refers to

$$\min_{\boldsymbol{\varepsilon}} \quad D(\boldsymbol{\varepsilon}), \quad \text{subject to:} \quad \boldsymbol{\varepsilon} \ge \boldsymbol{0}, \tag{14}$$

where the *dual function value* $D(\boldsymbol{\varepsilon})$ is given by the maximization of the Lagrangian

$$\begin{aligned} D(\boldsymbol{\varepsilon}) \triangleq \max_{r,\boldsymbol{g},\boldsymbol{f}} \quad & L(r, \boldsymbol{g}, \boldsymbol{f}, \boldsymbol{\varepsilon}) \\ \text{subject to:} \quad & \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r), \quad \forall t \in T, \\ & r \in [r_0, r_1], \\ & \boldsymbol{0} \le \boldsymbol{g} \le \boldsymbol{c}. \end{aligned} \tag{15}$$

[2] Bellman–Ford's shortest path algorithm has a serial complexity of $O(|V| \cdot |E|)$ where $|V|$ and $|E|$ are respectively the number of nodes and the number of links. The ϵ-relaxation algorithm for the minimum cost flow problem has a serial complexity of $O(|V|^3 + |V|^2\beta/\epsilon)$ (according to Bertsekas and Tsitsiklis [8], Section 5.4), where β is a measure of suboptimality of the initial dual variables. Hence for sparse networks, finding a shortest path is simpler than finding a minimum cost flow.

Since (12) is a convex optimization subject to linear constraints, the strong duality theorem in [22] (page 99) can be applied. Assuming that problem (3) is feasible and its optimal value is finite, then there is no duality gap and there exists at least one optimal dual vector. In short, assuming the existence of an optimal primal solution, there exist $\boldsymbol{\varepsilon}^* \geq \mathbf{0}$ satisfying

$$D(\boldsymbol{\varepsilon}^*) = \min_{\boldsymbol{\varepsilon} \geq \mathbf{0}} D(\boldsymbol{\varepsilon}) = U^*_{\text{net}}. \tag{16}$$

It is well known that the minimization of dual function (14) can be done via a subgradient method; see, e.g., [22]. Let $\boldsymbol{\xi}$ denote the subgradient of the dual function. The subgradient update of $\boldsymbol{\varepsilon}$ is

$$\boldsymbol{\varepsilon}_t^{(k+1)} = \left[\boldsymbol{\varepsilon}_t^{(k)} - \alpha_k \boldsymbol{\xi}_t^{(k)}\right]^+, \tag{17}$$

where $[x]^+ = \max\{x, 0\}$, α_k is the step size, $\boldsymbol{\varepsilon}_t^{(k)}$ is the current iterate. According to the Danskin's Theorem (see [22]), a subgradient of the dual function can be obtained essentially at no cost. More exactly, a subgradient of the dual function at the current dual vector $\boldsymbol{\varepsilon}^{(k)}$ can be obtained as the vector $\boldsymbol{\xi}^{(k)}$ composed by $|T|$ subvectors

$$\boldsymbol{\xi}_t^{(k)} = \boldsymbol{g}^{(k)} - \boldsymbol{f}_t^{(k)}, \tag{18}$$

where $(r^{(k)}, \boldsymbol{f}^{(k)}, \boldsymbol{g}^{(k)})$ maximizes the Lagrangian associated with $\boldsymbol{\varepsilon}^{(k)}$.

To implement the subgradient iterations, we need to maximize the Lagrangian over the primal variables. This will yield $(\boldsymbol{f}^{(k)}, \boldsymbol{g}^{(k)})$ required in (17). Due to the separable structure of (12), these computations can be separately carried out, as we shall show below.

Theorem 1 (Uniqueness of Lagrangian maximizer): Let $(r^{(k)}, \boldsymbol{f}^{(k)}, \boldsymbol{g}^{(k)})$ be a maximizer of (15) associated with $\boldsymbol{\varepsilon}^{(k)}$. Then

(i) $\boldsymbol{g}^{(k)}$ is uniquely determined from $\boldsymbol{\varepsilon}^{(k)}$ as

$$\boldsymbol{g}^{(k)} = \boldsymbol{g}(\boldsymbol{\varepsilon}^{(k)}), \tag{19}$$

where

$$\boldsymbol{g}(\boldsymbol{\varepsilon}) \triangleq \arg\max_{\mathbf{0} \leq \boldsymbol{g} \leq \boldsymbol{c}} \quad \left(\sum_{t \in T} \boldsymbol{\varepsilon}_t\right)^T \boldsymbol{g} - \sum_{vw \in E} p_{vw}(g_{vw}). \tag{20}$$

(ii) $r^{(k)}$ is uniquely determined from $\boldsymbol{\varepsilon}^{(k)}$ as

$$r^{(k)} = r(\boldsymbol{\varepsilon}^{(k)}), \tag{21}$$

where

$$r(\boldsymbol{\varepsilon}) \triangleq \arg\max_{r_0 \leq r \leq r_1} U(r) - r \cdot \sum_{t \in T} \mathcal{E}_t(\boldsymbol{\varepsilon}_t), \tag{22}$$

$$\mathcal{E}_t(\boldsymbol{\varepsilon}_t) \triangleq \min_{\boldsymbol{f}'_t} \ \boldsymbol{\varepsilon}_t^T \boldsymbol{f}'_t, \text{ subject to: } \boldsymbol{f}'_t \in \mathcal{F}_{s,t}(1). \tag{23}$$

(iii) $\boldsymbol{f}^{(k)}$ may not be unique. It is given by

$$\boldsymbol{f}^{(k)} = r^{(k)} \boldsymbol{f}'^{(k)}_t, \tag{24}$$

where $\boldsymbol{f}'^{(k)}_t$ is any optimizer of (23).

Remark: Note that $\mathcal{E}_t(\boldsymbol{\varepsilon}_t)$ refers to the minimum cost of an s–t flow providing unit rate, when the link prices are $\boldsymbol{\varepsilon}_t$. Since there are no other (bounding) constraints on the flow vector $\boldsymbol{f}'_t$, one solution of this problem is a flow corresponding to a shortest s–t path where the path lengths are given by $\boldsymbol{\varepsilon}_t$.

Proof: In (15), the maximization over the primal variables $r, \boldsymbol{g}, \boldsymbol{f}$ decouples into two sub-problems. The first sub-problem involves only $\boldsymbol{g}$:

$$\begin{aligned} D_g(\boldsymbol{\varepsilon}) \triangleq \max_{\boldsymbol{g}} \quad & \left(\sum_{t \in T} \boldsymbol{\varepsilon}_t\right)^T \boldsymbol{g} - \sum_{vw \in E} p_{vw}(g_{vw}) \\ \text{subject to:} \quad & \mathbf{0} \leq \boldsymbol{g} \leq \boldsymbol{c}. \end{aligned} \tag{25}$$

Since each p_{vw} is assumed to be strictly convex, there is a unique maximizer of (25). This establishes (i).

The second sub-problem involves only r and $\{\boldsymbol{f}_t\}$:

$$\begin{aligned} D_{r,f}(\boldsymbol{\varepsilon}) \triangleq \max_{r, \boldsymbol{f}} \quad & U(r) - \sum_{t \in T} \boldsymbol{\varepsilon}_t^T \boldsymbol{f}_t \\ \text{subject to:} \quad & \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r), \quad \forall t \in T, \\ & r \in [r_0, r_1]. \end{aligned} \tag{26}$$

In (26), for each fixed r, the maximization over $\{\boldsymbol{f}_t\}$ further decouples into $|T|$ sub-problems involving one destination each:

$$\begin{aligned} \max_{\boldsymbol{f}_t} \quad & -\boldsymbol{\varepsilon}_t^T \boldsymbol{f}_t \\ \text{subject to:} \quad & \boldsymbol{f}_t \in \mathcal{F}_{s,t}(r), \quad \forall t \in T. \end{aligned} \tag{27}$$

Since $\mathcal{F}_{s,t}(r) = r\mathcal{F}_{s,t}(1)$, the maximum value of (27) is linear in r. Then $D_{r,f}(\boldsymbol{\varepsilon})$ is given by the following scalar optimization.

$$\begin{aligned} D_{r,f}(\boldsymbol{\varepsilon}) = \max_{r} \quad & U(r) - r \cdot \sum_{t \in T} \mathcal{E}_t(\boldsymbol{\varepsilon}_t), \\ \text{subject to:} \quad & r \in [r_0, r_1]. \end{aligned} \tag{28}$$

Since $U(r)$ is assumed to be strictly concave, there is a unique maximizer of (28). This establishes (ii) and (iii). ■

C. Algorithm Summary and Economic Interpretation

It is well known that dual variables can be interpreted as prices for violating the relaxed constraints. In this specific context, the dual variables $\boldsymbol{\varepsilon}_t$ play the role of prices on the links. The algorithm can be interpreted as performing market adaptation via pricing. Given the current market price, the supplier decides how much bandwidth $\boldsymbol{g}$ to produce, trying to maximize its profit. This is captured in the *producer's optimization* (25), where the profit is the total payment from the $|T|$ consumers minus the operational cost. Given the current market price, the consumer decides how much bandwidth to consume, trying to maximize its net utility. This is captured in the *consumer's optimization* (26).

If the demand $\boldsymbol{f}_t$ exceeds the supply $\boldsymbol{g}$ on an edge, then the edge price is increased. Loosely speaking, the price increase on a link discourages using the link (i.e., the demand $\boldsymbol{f}_t$ will tend to decrease); at the same time, the price increase encourages the resource production (i.e., the supply $\boldsymbol{g}$ will tend to increase). Conversely, if the supply $\boldsymbol{g}$ exceeds the demand $\boldsymbol{f}_t$, then the edge price drops or remains 0 if it was 0. A price decrease will generally encourage consumption and discourage production.

We now summarize the proposed distributed algorithm.

Algorithm 1 (Dual subgradient method via shortest paths): Given the current dual vector $\boldsymbol{\varepsilon}^{(k)}$, the following steps are performed.

1. Solve the problem (25) in parallel to obtain $\boldsymbol{g}^{(k)}$. Note that this decouples into $|E|$ scalar optimizations, one for each edge. The optimal solution $\boldsymbol{g}^{(k)}$ is stored distributively in the network.
2. For each destination t, run a distributed shortest path algorithm with $\boldsymbol{\varepsilon}_t^{(k)}$ being the path lengthes. As a result, s knows the shortest path length $\mathcal{E}_t(\boldsymbol{\varepsilon}_t^{(k)})$; a binary flow vector $\boldsymbol{f}_t'^{(k)}$ corresponding to a shortest path from s to t is stored distributively in the network.
3. Solve the scalar maximization (28) at s to obtain $r^{(k)}$. The value $r^{(k)}$ is conveyed to each node involved in a shortest path, e.g., by passing it along the shortest paths.
4. (Optional) The source s broadcasts the current step size α_k. This may be combined with the broadcasting of $r^{(k)}$.
5. Generate a new dual vector $\boldsymbol{\varepsilon}^{(k+1)}$ according to (17), with

$$\boldsymbol{\xi}_t^{(k)} = \boldsymbol{g}^{(k)} - r^{(k)}\boldsymbol{f}_t'^{(k)}. \tag{29}$$

Synchronization of the algorithm may be achieved by having the source s broadcast the step size α_k and $r^{(k)}$. Then for each node, the receipt of such information serves as a clocking signal.

From Bootstrapping to Steady Phase: Recall that our main objective is to find a pair $(r, \boldsymbol{g})$ that approximately solves (3). The proposed distributed algorithm may be used in two possible phases. In the *bootstrapping phase*, the algorithm computes a near-optimal pair $(r, \boldsymbol{g})$, which are then used to set up the practical network coding system. In the *steady phase*, the practical network coding system is already running with some parameters $(r, \boldsymbol{g})$ and the proposed algorithm is used to fine tune the solution, or to adapt the solution in response to some changes in problem parameters. Note that this requires no changes to the actual network coding, since every coded packet is simply obtained by computing a random mixture of the buffered packets.

When the algorithm is used in the steady phase, it is desirable to update the current solution (in use) into another feasible solution, which can be put in use immediately. Note that $(r^{(k)}, \boldsymbol{g}^{(k)})$ obtained in the proposed algorithm may not be a feasible solution to (3); equivalently, $\boldsymbol{g}^{(k)}$ may not be able to exactly support rate $r^{(k)}$.

We now address the issue of determining the optimal operating rate $\hat{r}(\boldsymbol{g})$ for a given feasible $\boldsymbol{g}$. This can be done by optimizing over r in (3) with $\boldsymbol{g}$ fixed, i.e.,

$$\begin{aligned} \max_{r} \quad & U(r), \\ \text{subject to:} \quad & r \le \min_{t\in T} \rho_{s,t}(\boldsymbol{g}), \\ & r \in [r_0, r_1]. \end{aligned} \tag{30}$$

Note that if U is nondecreasing in r, then the optimal multicast rate associated with $\boldsymbol{g}$ reduces to

$$\hat{r}(\boldsymbol{g}) = \min\left\{r_1,\ \min_{t\in T} \rho_{s,t}(\boldsymbol{g})\right\}. \tag{31}$$

Typically, an application does not pose an upper bound on the required rate, i.e., $r_1 = \infty$. In this case, $\hat{r}(\boldsymbol{g})$ equals $\min_{t\in T} \rho_{s,t}(\boldsymbol{g})$. Evaluating $\min_{t\in T} \rho_{s,t}(\boldsymbol{g})$ can be done via a max-flow algorithm, e.g., Goldberg and Tarjan's Preflow–Push Algorithm [23]. Alternatively, if the practical network coding system [6] is running with $\boldsymbol{g}$, then the value $\min_{t\in T} \rho_{s,t}(\boldsymbol{g})$ can be obtained *algebraically* with a minor overhead. We now briefly explain this; for more details, see [24]. The practical network coding scheme [6] based on random linear network coding can achieve throughput close to the capacity. The source sends out packets in *generations*; packets within each generation are mixed randomly over the network. Suppose the source sends a generation of h_0 packets every τ seconds. If the rate h_0/τ is less than the capacity $\min_{t\in T} \rho_{s,t}(\boldsymbol{g})$ for some margin, with high probability, each destination will receive h_0 linearly independent (referring to the global coding vectors) mixture packets and can solve for the original h_0 source packets. As the operating parameters $\boldsymbol{g}$ are being gradually adapted, the capacity changes gradually as well. The source sending rate h_0/τ should be controlled to be always slightly less than the current capacity. In order to discover the actual capacity, we use a global coding vector of length h that is set to be slightly larger than an estimate of the true capacity, and treat as if there are h source packets with the last $h-h_0$ packets being all zero. By doing so, a destination t can use the rank of the global coding matrix for received packets to compute an estimate of the current min-cut value $\rho_{s,t}(\boldsymbol{g})$. Although the rank of the global coding matrix may be larger than h_0, the destination can still decode with high probability because there are only h_0 unknown packets. Each destination can report to the source node the rank of its global coding matrix; the source node can then estimate the current multicast capacity.

D. Recovery of primal optimal solution

Solving the dual problem (14) gives the optimal value of the objective function in the primal problem. However, our main objective is to find an optimal primal solution $(r^*, \boldsymbol{g}^*)$. Indeed, $(r^*, \boldsymbol{g}^*)$ is what critically needed in practical applications.

It is well known in Lagrangian duality theory that when the primal objective function is strictly concave, then a primal optimal solution can be recovered from an optimal dual solution. Particularizing to our context, we have the following result.

Lemma 2 (Recovery of optimal primal solution): Let $\boldsymbol{\varepsilon}^*$ be an optimal dual vector. Then $(r(\boldsymbol{\varepsilon}^*), \boldsymbol{g}(\boldsymbol{\varepsilon}^*))$ is the unique maximizer of (3).

Proof: Let $(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*)$ be an optimal solution to (12). From Lagrangian duality theory, if there is no duality gap, then $(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*)$ optimizes the Lagrangian for $\boldsymbol{\varepsilon}^*$, i.e.,

$$L(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*, \boldsymbol{\varepsilon}^*) = D(\boldsymbol{\varepsilon}^*). \tag{32}$$

Since U is strictly concave and each p_{vw} is strictly convex, $r^* = r(\boldsymbol{\varepsilon}^*)$ and $\boldsymbol{g}^* = \boldsymbol{g}(\boldsymbol{\varepsilon}^*)$. ■

In practice, very often one can only hope for an approximate optimal dual solution. For example, if a constant step size ($\alpha_k = h$) is used in the subgradient iterations (17), then the subgradient algorithm converges to within some neighborhood of the optimal value (proportional to the step size used); see Appendix and the references therein for more information. Thus it is of importance to investigate the robustness of the primal recovery method.

Suppose we have an approximate minimizer of the dual problem $\boldsymbol{\varepsilon} \geq \mathbf{0}$ in the sense that $D(\boldsymbol{\varepsilon}) - D(\boldsymbol{\varepsilon}^*)$ is small. In the following we quantify the distance between $(\boldsymbol{g}(\boldsymbol{\varepsilon}), \boldsymbol{r}(\boldsymbol{\varepsilon}))$ and $(\boldsymbol{g}^*, r^*) = (\boldsymbol{g}(\boldsymbol{\varepsilon}^*), \boldsymbol{r}(\boldsymbol{\varepsilon}^*))$ via $D(\boldsymbol{\varepsilon}) - D(\boldsymbol{\varepsilon}^*)$.

The subsequent robustness analysis assumes

A3: $\{p_{vw}\}$ and $U(r)$ are twice continuously differentiable.

Let $\dot{f}(x)$ and $\ddot{f}(x)$ denote respectively the first and second order derivatives of a function $f(x)$.

Theorem 2 (Robustness of primal recovery): Let $\boldsymbol{\varepsilon}^*$ be an optimal dual vector. Consider (an approximate minimizer of the dual problem) $\boldsymbol{\varepsilon} \geq \mathbf{0}$. Define

$$d \triangleq D(\boldsymbol{\varepsilon}) - D(\boldsymbol{\varepsilon}^*). \tag{33}$$

Assuming A1, A2, and A3, then $\forall \epsilon > 0, \exists \delta > 0$, such that $d < \delta$ implies

$$\|\boldsymbol{g}(\boldsymbol{\varepsilon}) - \boldsymbol{g}^*\|^2 \leq \frac{2d(1+\epsilon)}{\min_{vw \in E} \ddot{p}_{vw}(\boldsymbol{g}^*)}, \tag{34}$$

$$(r(\boldsymbol{\varepsilon}) - r^*)^2 \leq \frac{2d(1+\epsilon)}{-\ddot{U}(r^*)}. \tag{35}$$

In the special case where the link cost functions p_{vw} are quadratic functions, we have

$$\|\boldsymbol{g}(\boldsymbol{\varepsilon}) - \boldsymbol{g}^*\|^2 \leq \frac{2d}{\min_{vw \in E} \ddot{p}_{vw}(\boldsymbol{g}^*)}. \tag{36}$$

Proof: See Appendix. ■

This theorem implies the convergence of the primal and dual sequences if a proper step size rule is used.

Corollary 1 (Convergence of primal sequence): Assuming A1, A2, A3, if $D(\boldsymbol{\varepsilon}^{(k)}) \longrightarrow D(\boldsymbol{\varepsilon}^*)$, then

$$r^{(k)} \longrightarrow r^*, \tag{37}$$

$$\boldsymbol{g}^{(k)} \longrightarrow \boldsymbol{g}^*, \tag{38}$$

$$\hat{r}^{(k)} \triangleq \hat{r}(\boldsymbol{g}^{(k)}) \longrightarrow \hat{r}(\boldsymbol{g}^*) = r^*. \tag{39}$$

Therefore, when a proper step size rule is used (e.g., $\alpha_k = a/(k+b)$), the dual sequence $\boldsymbol{\varepsilon}^{(k)}$ will converge to an optimal dual vector $\boldsymbol{\varepsilon}^*$ and thus the feasible primal sequences $\{(\hat{r}^{(k)}, \boldsymbol{g}^{(k)})\}$ will converge to the unique optimal primal variables.

Proof: The convergence of $\{r^{(k)}\}$ and $\boldsymbol{g}^{(k)}$ follows directly from Theorem 2. Because the function $\hat{r}(\boldsymbol{g})$ is continuous in $\boldsymbol{g}$ and $\boldsymbol{g}^{(k)}$ converges to $\boldsymbol{g}^*$, $\{\hat{r}^{(k)}\}$ converges to $\hat{r}(\boldsymbol{g}^*)$, which equals the unique optimal rate r^*. ■

From this Theorem 2, it is seen that $\ddot{p}_{vw}(g^*_{vw})$ and $-\ddot{U}(r^*)$ are critical in determining the robustness of the primal recovery method.

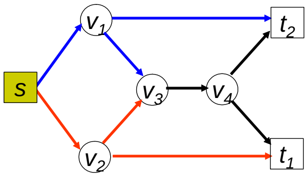


Fig. 1. An example network. This is the classical example of network coding, introduced in [1].

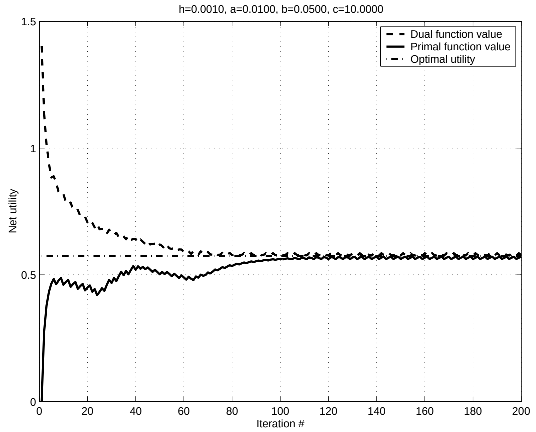


Fig. 2. Primal and dual function values vs. iterations. $a = 0.01$, $b = 0.05, c = 10$, $h = 0.001$.

E. Illustrative examples

Let us now examine some examples. Consider the graph given in Fig. 1, with

$$\begin{aligned} U(r) &= \ln(1+r), \\ p_{vw}(g_{vw}) &= ag^2 + bg, \quad \forall vw \in E, \\ c_{vw} &= c, \\ [r_0, r_1] &= [0, \infty]. \end{aligned}$$

Thus the parameters are a, b, c. We use a constant step size $\alpha_k = h$ in the subgradient update (17).

1) A well-conditioned scenario: For $a = 0.01$, $b = 0.05, c = 10$, and step size $h = 0.001$, Fig. 2 plots the primal

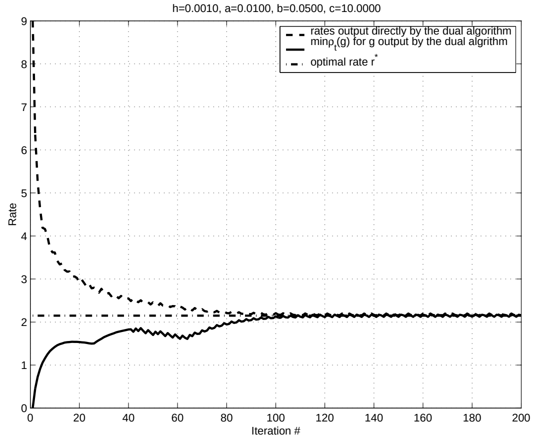


Fig. 3. The rates $r^{(k)}$ output by the algorithm and the multicast rate supported by $\boldsymbol{g}^{(k)}$. $a = 0.01$, $b = 0.05, c = 10$, $h = 0.001$.

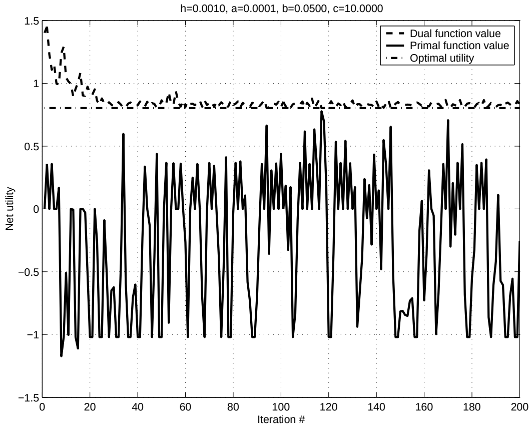


Fig. 4. Primal and dual function values vs. iterations. $a = 0.0001,\ b = 0.05, c = 10,\ h = 0.001$.

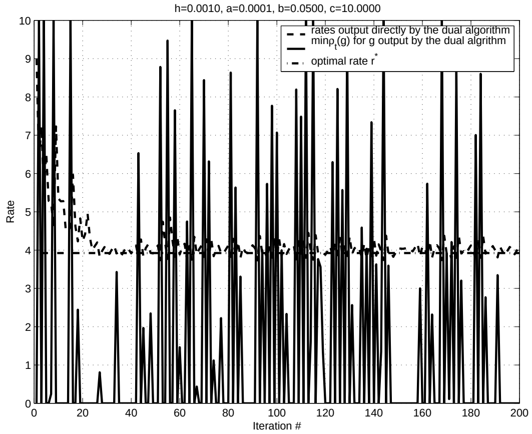


Fig. 5. The rates $r^{(k)}$ output by the algorithm and the multicast rate supported by $\boldsymbol{g}^{(k)}$. $a = 0.0001,\ b = 0.05, c = 10,\ h = 0.001$.

and dual function values vs. iterations. More precisely, for the k-th iteration, the dual function value shown is $D(\boldsymbol{\varepsilon}^{(k)})$; the primal function value shown is

$$U_{\text{net}}^{(k)} \triangleq U\left(\min_{t \in T} \rho_{s,t}(\boldsymbol{g}^{(k)})\right) - P(\boldsymbol{g}^{(k)}), \tag{40}$$

where $P(\boldsymbol{g}^{(k)})$ is the short-hand notation for $\sum_{vw \in E} p_{vw} g_{vw}^{(k)}$. The horizontal straight line gives the optimal value U_{net}^*. It is seen that both $D(\boldsymbol{\varepsilon}^{(k)})$ and $U_{\text{net}}^{(k)}$ approaches U_{net}^*. The sequence $D(\boldsymbol{\varepsilon}^{(k)})$ appears to have a consistent descent trajectory, whereas the sequence $U_{\text{net}}^{(k)}$ has some oscillations at the beginning. The fact that the former sequence is more consistent is not unexpected, since the proposed method works by trying to minimize the dual function $D(\boldsymbol{\varepsilon})$.

We next examine the optimality of the rate sequence $\{r^{(k)}\}$ output by the algorithm; recall that $r^{(k)}$ is obtained as the optimizer of (28). Fig. 3 plots $r^{(k)}$ and compare it with $\hat{r}^{(k)} = \min_{t \in T} \rho_{s,t}(\boldsymbol{g}^{(k)})$ – the multicast rate supported by $\boldsymbol{g}^{(k)}$. It is seen that $r^{(k)}$ is close to $\hat{r}^{(k)}$ when the algorithm approaches optimality. This confirms Corollary 1. If $\boldsymbol{\varepsilon}^{(k)}$ is near optimal, then $\boldsymbol{g}^{(k)}$ is close to $\boldsymbol{g}^*$ and $r^{(k)}$ is close to r^* (see Theorem 2). Since $\boldsymbol{g}^{(k)}$ is close to $\boldsymbol{g}^*$, $\hat{r}^{(k)}$ is close to r^*. Therefore, in this operating region, $r^{(k)}$ can be used as an approximation of the supportable rate $\hat{r}^{(k)}$.

2) An ill-conditioned scenario: For $a = 0.0001$, $b = 0.05, c = 10$, and step size $h = 0.001$, Fig. 4 plots the primal and dual function values vs. iterations. In this case, the dual sequence reaches a fairly close neighborhood of U_{net}^*. However, the dual sequence $D(\boldsymbol{\varepsilon}^{(k)})$ oscillates significantly. It can be far from U_{net}^* even when the dual sequence is close to U_{net}^*. This *ill-conditioning* behavior can be attributed to the small a used; this is consistent with Theorem 2. Thus when the link cost functions are almost linear, to ensure the recovered primal sequence to be close to optimum, the dual problem must be solved very accurately.

Fig. 5 plots $r^{(k)}$ and $\hat{r}^{(k)}$ for this ill-conditioned scenario. It is seen that $r^{(k)}$ is still close to r^* when the dual function values $D(\boldsymbol{\varepsilon})$ are near optimal. This is because the utility function $U(r)$ is still well-conditioned. The sequence $\{\hat{r}^{(k)}\}$, however, exhibits a highly oscillatory behavior just as $\{\boldsymbol{g}^{(k)}\}$ does; indeed, they are determined from $\{\boldsymbol{g}^{(k)}\}$.

In summary, the degree of strict concavity of the objective function plays a critical role in determining the robustness of the algorithm in producing a high quality primal sequence $\boldsymbol{g}^{(k)}$.

III. Dealing with Lack of Strict Concavity

Comparing Fig. 2 with Fig. 4, we see that the numerical degree of strict concavity of $U_{\text{net}}(r, \boldsymbol{g})$ is critical to the robust recovery of the primal solutions from the (suboptimal) dual vectors. More generally, Theorem 2 shows that $\ddot{p}_{vw}(g_{vw}^*)$ and $\ddot{U}(r^*)$ are critical in determining the robustness of the primal recovery method. In this section we focus on the case that the given problem does not have sufficient degree of strict concavity. Such scenario is neither rare nor ignorable. Indeed, linear cost functions – which arise in many useful applications – are not strictly convex at all.

We propose to apply the *proximal method* (see e.g., [22]) when the problem lacks a sufficient degree of strict concavity/convexity. This method is well known in optimization theory. (For a historical note, see [8].) The basic idea is to add a quadratic term to "regularize" the optimization of the objective function; then a *sequence* of modified problems is solved in lieu of the (single) original problem.[3]

Specifically, for the maximization of a concave function, $\max_{\boldsymbol{x} \in X} F(\boldsymbol{x})$, the *proximal method* uses the following iterations

$$\boldsymbol{x}_{(n)} = \arg\max_{\boldsymbol{x} \in X} \left\{ F(\boldsymbol{x}) - a_n \|\boldsymbol{x} - \boldsymbol{x}_{(n-1)}\|^2 \right\}. \tag{41}$$

Note that with the presence of the quadratic term, the revised function is strictly concave in $\boldsymbol{x}$. Hence it has a unique maximizer. It can be shown that if $\{a_n\}$ is a sequence of positive numbers bounded away from 0, then the sequence $\{\boldsymbol{x}_{(n)}\}$ converges to an optimal solution; for a proof, please refer to [8]. An extension of the proximal method is the

[3] A related approach was used by Xiao et al. in [10], where they proposed to add a small quadratic regularization term to recover the primal variables when applying the dual decomposition method to problems with a separable structure. The proximal method can be viewed as a systematic way of adding regularization terms.

partial proximal method [25] [26], where the quadratic term in (41) involves only some of the minimization variables. The convergence results can be found in [25] [26].

Regarding the selection of the regularization parameter a_n, generally speaking, there is a tradeoff. A larger a_n makes the primal recovery more robust, but it also makes the regularized problem more different from the original problem, especially at the beginning, when $\boldsymbol{x}_{(0)}$ is far from optimal.

Now we describe the particular application of the proximal method to our current problem. Given the *previous solution* $(r_{(n-1)}, \boldsymbol{g}_{(n-1)})$, maximize the regularized utility function[4]

$$U_{\text{net}}^{(n)}(r, \boldsymbol{g}) \triangleq U_{\text{net}}(r, \boldsymbol{g}) - a_n \|\boldsymbol{g} - \boldsymbol{g}_{(n-1)}\|^2 - a_{2,n} \|r - r_{(n-1)}\|^2. \tag{42}$$

We denote this problem by *subproblem* n. This maximization can be approximately solved using the algorithm described in Section II. The resulting dual vector is denoted by $\boldsymbol{\varepsilon}_{(n)}$ and the recovered primal vectors are denoted by

$$r_{(n)} = r(\boldsymbol{\varepsilon}_{(n)}), \quad \boldsymbol{g}_{(n)} = \boldsymbol{g}(\boldsymbol{\varepsilon}_{(n)}).$$

Note that due to the presence of the quadratic term, such a maximization is expected to be better conditioned than the original problem. Provided that each maximization of $U_{\text{net}}^{(n)}$ is sufficiently accurate, the sequence $(r_{(n)}, \boldsymbol{g}_{(n)})$ converges to an optimal solution of the original problem (3).

A. Using previous solution to initialize current problem

For a sufficiently large n, $\boldsymbol{g}_{(n)}$ is close to $\boldsymbol{g}_{(n-1)}$. In other words, the previous solution $(r_{(n-1)}, \boldsymbol{g}_{(n-1)})$ approximately maximizes the current objective $U_{\text{net}}^{(n)}(r, \boldsymbol{g})$. Therefore, it is desirable to use $(r_{(n-1)}, \boldsymbol{g}_{(n-1)})$ to guide the initialization of the current subproblem. However, the algorithm described in Section II is a dual subgradient approach, which is driven by the dual vector $\boldsymbol{\varepsilon}$. Thus we need an initial dual vector $\boldsymbol{\varepsilon}_{(n)}^{(0)}$.[5] We assume that $a_{n-1} \approx a_n$ and $a_{2,n-1} \approx a_{2,n}$. Then subproblem $n-1$ is similar to subproblem n, and consequently the dual of subproblem $n-1$ is similar to the dual of subproblem n. This justifies setting $\boldsymbol{\varepsilon}_{(n)}^{(0)} := \boldsymbol{\varepsilon}_{(n-1)}$, the (near-)optimal dual vector for subproblem $n-1$.

B. Illustrative examples

We now apply the proximal method to the ill-conditioned problem in Section II-E. Since $U(r)$ is well-conditioned, we do not regularize the rates. A constant sequence $a_n = 0.0099$ is used. Thus, the first proximal iteration amounts to the problem discussed in Fig. 2.

Fig. 6 plots the progression of the proximal method with respect to the proximal iteration n. Specifically, three curves are shown. The lowest one is the regularized primal value for the solution $\boldsymbol{g}_{(n)}$ found by the dual subgradient method; in other words, it plots

$$U\left(\min_{t \in T} \rho_{s,t}(\boldsymbol{g}_{(n)})\right) - P(\boldsymbol{g}_{(n)}) - a_n \|\boldsymbol{g}_{(n)} - \boldsymbol{g}_{(n-1)}\|^2.$$

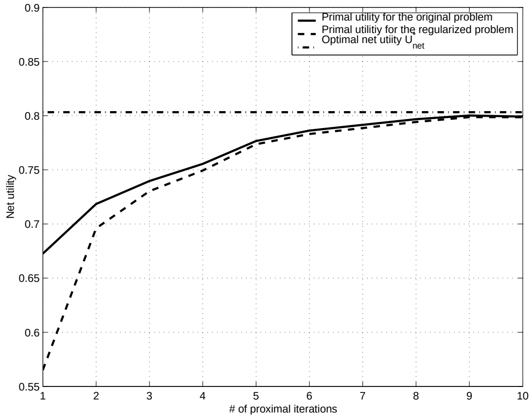


Fig. 6. The progression of the proximal method with respect to the proximal iteration n. $a = 0.0001,\ b = 0.05, c = 10,\ h = 0.001$.

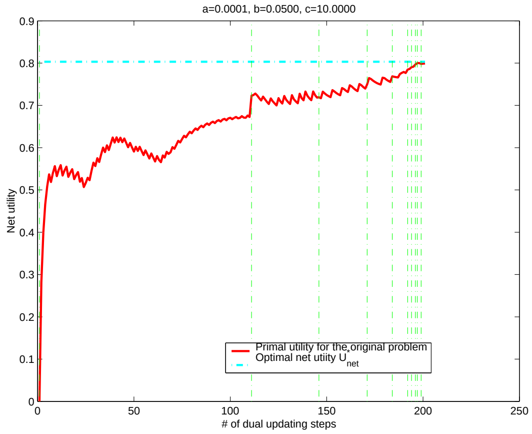


Fig. 7. Original primal function values vs. iterations using the proximal method. Each iteration is one execution of a dual subgradient update. $a = 0.0001,\ b = 0.05, c = 10,\ h = 0.001$.

The middle curve plots the original primal utility

$$U\left(\min_{t \in T} \rho_{s,t}(\boldsymbol{g}_{(n)})\right) - P(\boldsymbol{g}_{(n)}).$$

The difference between these two curves corresponds to

$$a_n \|\boldsymbol{g}_{(n)} - \boldsymbol{g}_{(n-1)}\|^2 \geq 0.$$

Therefore Fig. 6 shows that the sequence $\boldsymbol{g}_{(n)}$ converges. The uppermost line plots the maximum utility U_{net}^*. Thus the generated primal function values converge to the optimum.

Fig. 7 plots the original primal function values versus the subgradient iterations. The graph is partitioned into several parts by the vertical dashed lines. The first part corresponds to the subgradient iterations for primal iteration $n = 1$, and so on. Note that each part has an unequal number of iterations. This is because a stopping criterion is used here to terminate the subgradient iterations when it already reaches near-optimum. For this graph, the subgradient iterations are stopped when the gap between the optimal primal value found and the current dual value is sufficiently small. Since the later proximal iterations start with a good initial solution, it requires a relatively small number of iterations.

[4] If $U(r)$ is already sufficiently concave (as quantized by its second order derivative around r^*), the quadratic term involving r may be dropped.

[5] Note that the subscript denotes the proximal iteration number and the superscript denotes the dual subgradient iteration number.

Fig. 7 is the counterpart of Fig. 4. It is seen that the proximal method has successfully made each subproblem better-conditioned.

IV. Extension to Multiple Multicast Sessions and Flexible Supplies

Consider the scenario where there are multiple multicast sessions in the network. Label them with indices $m = 1, \ldots, M$. Denote the source and the destination set of the m-th session by s_m and T_m, respectively. For this multi-source communication problem, a simple communication scheme is to partition the available resource into M shares and let each session communicate using its exclusive share of resource; this will be referred to as the *resource-partitioning* approach. This approach is in general suboptimal, even if the messages in different sessions are independent, as first pointed out by Yeung [27]. The multi-source communication problem remains an open challenge; for some discussions, see, e.g., [16], [28], [29]. In the following we consider the distributed resource allocation issues when using the resource-partitioning approach.

If the resource share for the m-th multicast session is $\boldsymbol{g}_m$, the maximum multicast rate for this session is

$$\min_{t \in T_m} \rho_{s_m,t}(\boldsymbol{g}_m). \tag{43}$$

In some networks, the supply of bit-rate resources $\boldsymbol{c}$ offers an additional degree of freedom in the system. A useful application of this nature is the cross-layer optimization in wireless ad hoc networks. Cross-layer optimizations in a wireless network using network coding have been formulated in [20], [21], [30].

The following optimization models the resource sharing among multiple multicast sessions and the potential flexibility in choosing $\boldsymbol{c}$.

$$\begin{aligned} \max_{\{r_m, \boldsymbol{g}_m\}, \boldsymbol{c}} \quad & \sum_{m=1}^{M} U_m(r_m) - \sum_{vw \in E} p_{vw}(c_{vw}) \\ \text{subject to:} \quad & r_m \le \min_{t \in T_m} \rho_{s_m,t}(\boldsymbol{g}_m), \\ & r_m \in [r_{m0}, r_{m1}], \\ & \boldsymbol{0} \le \boldsymbol{g}_m, \quad \forall m \\ & \boldsymbol{g}_1 + \ldots + \boldsymbol{g}_M \le \boldsymbol{c}, \\ & \boldsymbol{c} \in \mathcal{C}. \end{aligned} \tag{44}$$

Note that here the cost function is with respect to $\boldsymbol{c}$ whereas in (3) the cost function is with respect to $\boldsymbol{g}$.

The variables in (44) are the supply-side variables $\boldsymbol{c}$ and the demand-side variables $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M$. It can be seen from (44) that the supply side interacts with the demand side only through the constraint

$$\boldsymbol{g}_1 + \ldots + \boldsymbol{g}_M \le \boldsymbol{c}. \tag{45}$$

It is by now well known that problems with such type of cross-coupling can be decomposed into subproblems via Lagrangian relaxation and subgradient iterations. For example, several previous works, e.g., [10]–[12], [14], have applied similar techniques to various formulations of cross-layer optimizations in wireless networks.

However, we note that a direct application of the Lagrangian relaxation would result in subproblems that correspond to (3) with linear cost functions; this can be seen from the derivations below. Due to the lack of strict concavity, this would lead to difficulties in recovering the primal solutions. As a remedy for this issue, we propose to apply the proximal method. The proximal method adds strictly concave terms to regularize the objective functions without affecting the separable structure of the problem.

We next describe how to use the proximal method in conjunction with Lagrangian relaxation. A sequence of regularized subproblems is solved in lieu of the original problem. In subproblem n, the regularized net-utility function is

$$\begin{aligned} U_{\text{net}}^{(n)}(\{r_m\}, \boldsymbol{c}) \triangleq & \sum_{m=1}^{M} U_m(r_m) - \sum_{vw \in E} p_{vw}(c_{vw}) \\ & - a_r \sum_m (r_m - r_m^{(n-1)})^2 \\ & - a_g \sum_m \|\boldsymbol{g}_m - \boldsymbol{g}_m^{(n-1)}\|^2 \\ & - a_c \|\boldsymbol{c} - \boldsymbol{c}^{(n-1)}\|^2, \end{aligned} \tag{46}$$

where $a_g > 0$, $a_r \ge 0$ and $a_c \ge 0$ are regularization coefficients. Note that the regularization coefficients a_r and a_c may not be necessary if U_m and p_{vw} are well conditioned.

We can then perform the dual decomposition on each regularized subproblem. Introduce dual variables $\boldsymbol{\lambda}$ and form the Lagrangian

$$\begin{aligned} & L(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M, \boldsymbol{c}, \boldsymbol{\lambda}) \\ = & U_{\text{net}}^{(n)}(\{r_m\}, \boldsymbol{c}) - \boldsymbol{\lambda}^T(\boldsymbol{g}_1 + \ldots + \boldsymbol{g}_M - \boldsymbol{c}) \end{aligned} \tag{47}$$

Then the maximization of $L(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M, \boldsymbol{c}, \boldsymbol{\lambda})$ over the primal variables $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M, \boldsymbol{c}$ decouples into a supply-side sub-problem and M demand-side sub-problems, one for each session. The supply-side problem is

$$\begin{aligned} \max_{\boldsymbol{c}} \quad & \boldsymbol{\lambda}^T \boldsymbol{c} - \sum_{vw \in E} p_{vw}(c_{vw}) - a_c \|\boldsymbol{c} - \boldsymbol{c}^{(n-1)}\|^2 \\ \text{subject to:} \quad & \boldsymbol{c} \in \mathcal{C}. \end{aligned} \tag{48}$$

The demand-side sub-problem for the m-th session is

$$\begin{aligned} \max_{r_m, \boldsymbol{g}_m} \quad & U_m(r_m) - a_r(r_m - r_m^{(n-1)})^2 \\ & - \boldsymbol{\lambda}^T \boldsymbol{g}_m - a_g \|\boldsymbol{g}_m - \boldsymbol{g}_m^{(n-1)}\|^2 \\ \text{subject to:} \quad & r_m \le \min_{t \in T_m} \rho_{s_m,t}(\boldsymbol{g}_m), \\ & r_m \in [r_{m0}, r_{m1}], \\ & \boldsymbol{0} \le \boldsymbol{g}_m. \end{aligned} \tag{49}$$

Note that without the quadratic regularization term on $\boldsymbol{g}_m$, the objective function would not be strictly concave. Hence the regularization on $\boldsymbol{g}_m$ is critical to the unique recovery of primal solution from a dual vector.

For some $\boldsymbol{\lambda}$ and U_m, the problem (49) may diverge since the feasible region is unbounded. This issue can be avoided by adding an upper-bound on $\boldsymbol{g}_m$ so that the constraint becomes

$$\mathbf{0} \leq \boldsymbol{g}_m \leq \boldsymbol{u}. \tag{50}$$

The upper bound can be sufficiently large such that it is loose in the primal formulation. If the set $\mathcal{C}$ is bounded, we can set $\boldsymbol{u}$ such that $\boldsymbol{u} \geq \boldsymbol{c}$, $\forall \boldsymbol{c} \in \mathcal{C}$.

The dual variables $\boldsymbol{\lambda}$ can be updated via a subgradient method as

$$\boldsymbol{\lambda}^{(k+1)} = \left[\boldsymbol{\lambda}^{(k)} + \alpha_k(\boldsymbol{g}_1^* + \ldots + \boldsymbol{g}_M^* - \boldsymbol{c}^*)\right]^+, \tag{51}$$

where $\boldsymbol{g}_1^*, \ldots, \boldsymbol{g}_M^*, \boldsymbol{c}^*$ solve the respective subproblems.

V. Simulations

We tested the algorithms in a large scale scenario. The graph (V, E) is the topology of an ISP (Exodus) backbone obtained from the Rocketfuel project at the University of Washington [31]. There are 79 nodes and 294 links. We arbitrarily placed a source node at New York, and 8 destination nodes at Oak Brook, Jersey City, Weehawken, Atlanta, Austin, San Jose, Santa Clara, and Palo Alto. The utility function, the link cost functions and link capacities are set as

$$U(r) = \ln(1+r), \tag{52}$$

$$p_{vw}(g) = ag^2 + 0.005g, \quad \forall vw \in E, \tag{53}$$

$$c_{vw} = 10, \quad \forall vw \in E, \tag{54}$$

$$[r_0, r_1] = [0, \infty]. \tag{55}$$

A. Test Problem 1: $a = 0.001$

Fig. 8(a)(b)(c) plot the primal and dual function values vs. iterations for different choices of step sizes.[6] Fig. 8(a)(b) are obtained by using constant step sizes, $\alpha_k = 0.0001$ and $\alpha_k = 0.00003$, respectively. With a constant step size, the subgradient method is guaranteed to eventually reach some neighborhood of the optimum; see Appendix.A. It is observed that the primal and dual curves with the larger step size converge faster, but the primal curve with the smaller step size is smoother and the dual curve with the smaller step size reaches closer to the optimum.

In Fig. 8(c), a diminishing step size $\alpha_k = 0.0005/\sqrt{k}$ is used. The curve is smoother than Fig. 8(a) and converges faster than Fig. 8(b).

B. Test Problem 2: $a = 0$

For the case where the cost function is linear, the proximal method is applied with a quadratic regularization term $0.001\|\boldsymbol{g} - \boldsymbol{g}_{(n-1)}\|^2$. Initially, we set $\boldsymbol{g}_{(0)} = \mathbf{0}$. Thus the first subproblem amounts to test problem 1. We simulated 5 proximal iterations, with 200 subgradient updates in each proximal iteration. The step sizes used in the 5 subproblems are $0.0005/\sqrt{k}$, $0.0004/\sqrt{k}$, $0.0003/\sqrt{k}$, $0.0002/\sqrt{k}$, and $0.0001/\sqrt{k}$, respectively. Fig. 9 plots the sequence of original primal function values obtained. The results confirm the convergence of the algorithm.

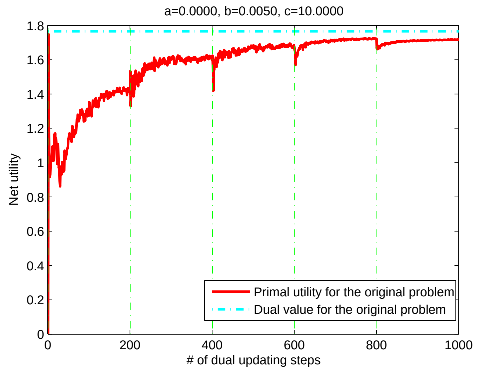


Fig. 9. Original primal function values vs. iterations using the proximal method.

VI. Conclusion

By employing the unique characterization of the multicast throughput attainable via network coding and a problem-specific dual formulation, we have decomposed the utility maximization problem into subproblems each involving one destination only. Theoretically, the paper has two key contributions. First, the resulting subproblem amounts to the problem of finding a shortest path from the source to each destination, for which there exist well established distributed algorithms. Second, assuming the net-utility function is strictly concave, our approach enables a near-optimal primal variable to be uniquely recovered from a near-optimal dual variable. From practical and numerical perspective, we have adopted the proximal method to reformulate original ill-conditioned problem (e.g., with linear cost functions) into a sequence of well-conditioned problems. The simulation results further confirm the numerical robustness of the proposed algorithms. Finally, the proximal method and the dual subgradient method are naturally extended to solve the generalized problem where multiple multicast sessions are simultaneously transmitted.

Appendix

A. Review: preliminaries on subgradient methods

Definition 1 (Subgradient): Given a convex function f, a vector $\boldsymbol{\xi}$ is said to be a *subgradient* of f at $\boldsymbol{x} \in \text{dom } f$ if

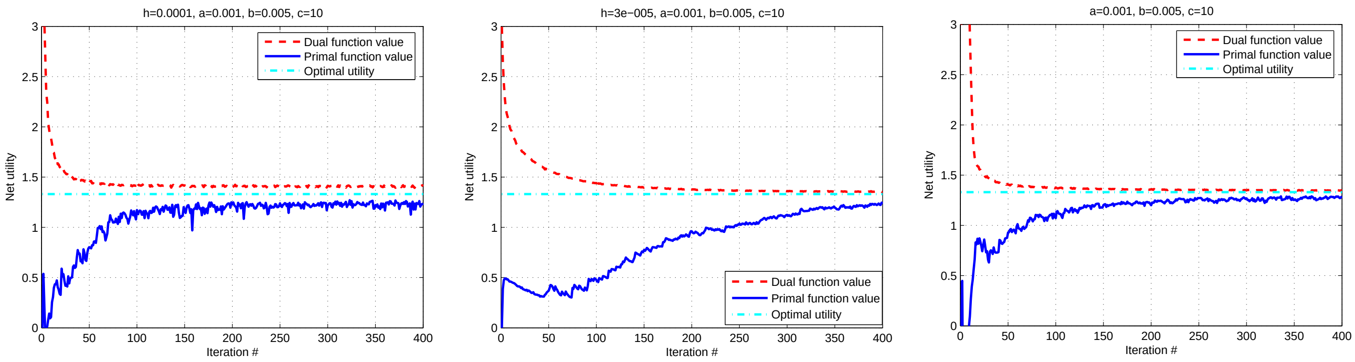
$$f(\boldsymbol{x}') \geq f(\boldsymbol{x}) + \boldsymbol{\xi}^T(\boldsymbol{x}' - \boldsymbol{x}), \quad \forall \boldsymbol{x}' \in \text{dom } f. \tag{56}$$

The subgradient method [7] minimizes a non-differentiable convex function in a way similar to gradient methods for differentiable functions – in each step, the variables are updated in the negative direction of a subgradient. However, such a direction may not be a descent direction; instead, the subgradient method relies on a different property. If the variable moves a sufficiently small step along the direction of a subgradient, the new point is closer to any optimal solution.

Consider a generic, constrained convex minimization:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to:} \quad & \boldsymbol{x} \in \mathcal{C}, \end{aligned} \tag{57}$$

[6] For ease of visualization and comparison, we have plotted the y-axis with a range of $[0, 3]$. At the beginning of the iterations, the curves for (a) and (c) go out of this range.



(a) $\alpha_k = 0.0001$

(b) $\alpha_k = 0.00003$

(c) $\alpha_k = 0.0005/\sqrt{k}$

Fig. 8. Primal and dual function values vs. iterations for test problem 1 with $a = 0.001$. (a) $\alpha_k = 0.0001$ (b) $\alpha_k = 0.00003$ (c) $\alpha_k = 0.0005/\sqrt{k}$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, and $\mathcal{C}$ is a closed and nonempty convex set. The subgradient method uses the iteration

$$\boldsymbol{x}^{(k+1)} = P\left[\boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{\xi}^{(k)}\right], \tag{58}$$

where $\boldsymbol{x}^{(k)}$ is the k-th iterate, $\boldsymbol{\xi}^{(k)}$ is any subgradient of f at $\boldsymbol{x}^{(k)}$, $\alpha_k > 0$ is the k-th step size, and P is the projection on $\mathcal{C}$:

$$P[\boldsymbol{x}] \triangleq \arg\min_{\boldsymbol{x}' \in \mathcal{C}} \|\boldsymbol{x}' - \boldsymbol{x}\|^2. \tag{59}$$

Lemma 3 (Convergence of subgradient methods [32]): Assume $\boldsymbol{x}^*$ is a minimizer of (57) and there exists a G such that $\|\boldsymbol{\xi}^k\| \le G$, $\forall k$. Then

$$\min_{i=1,\ldots,k} f(\boldsymbol{x}^{(i)}) - f^* \le \frac{\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^*\| + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i}. \tag{60}$$

In particular,

- if constant step size is used, i.e., $\alpha_k = h$, then the right hand side of (60) converges to $G^2 h/2$ as $k \to \infty$.
- if the step sizes satisfy

$$\lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty, \tag{61}$$

then right hand side of (60) converges to 0 as $k \to \infty$. Step sizes that satisfy this condition are called *diminishing step size rules*. Examples include $\alpha_k = a/\sqrt{k}$ and $\alpha_k = a/(k+b)$, where $a, b > 0$.

For more discussions on the step size selection, see, e.g., [33].

B. Proof of Theorem 2

Lemma 4: Let ε^* be an optimal dual vector. For fixed $\boldsymbol{\varepsilon} \ge \boldsymbol{0}$, let

$$U_1(\boldsymbol{g}) \triangleq \left(\sum_{t\in T} \boldsymbol{\varepsilon}_t\right)^T \boldsymbol{g} - \sum_{vw\in E} p_{vw}(g_{vw}), \tag{62}$$

$$U_2(r) \triangleq U(r) - r \cdot \sum_{t\in T} \mathcal{E}_t(\boldsymbol{\varepsilon}_t). \tag{63}$$

Then

$$d \triangleq D(\boldsymbol{\varepsilon}) - D(\boldsymbol{\varepsilon}^*) \ge \max_{\boldsymbol{0}\le \boldsymbol{g}\le \boldsymbol{c}} U_1(\boldsymbol{g}) - U_1(\boldsymbol{g}^*) + \max_{r\in[r_0,r_1]} U_2(r) - U_2(r^*). \tag{64}$$

Remark: This lemma establishes that when d is small, then $\boldsymbol{g}^*$ approximately solves $\max_{\boldsymbol{0}\le\boldsymbol{g}\le\boldsymbol{c}} U_1(\boldsymbol{g})$ and r^* approximately solves $\max_{r\in[r_0,r_1]} U_2(r)$.

Proof: Let $(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*)$ be an optimal primal solution. Since $\boldsymbol{\varepsilon} \ge \boldsymbol{0}$ and $\boldsymbol{f}_t^* \le \boldsymbol{g}^*$, we have

$$-\sum_{t\in T} \boldsymbol{\varepsilon}_t^T (\boldsymbol{f}_t^* - \boldsymbol{g}^*) \ge 0.$$

Thus from the form of Lagrangian (13), we have

$$L(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*, \boldsymbol{\varepsilon}) \ge U_{\text{net}}^* = D(\boldsymbol{\varepsilon}^*).$$

Next, since $D(\boldsymbol{\varepsilon})$ is obtained by maximizing the Lagrangian over the primal variables, we have

$$D(\boldsymbol{\varepsilon}) \ge L(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*, \boldsymbol{\varepsilon}).$$

Putting these together, we have

$$U_{\text{net}}^* + d = D(\boldsymbol{\varepsilon}) \ge L(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*, \boldsymbol{\varepsilon}) \ge U_{\text{net}}^* = D(\boldsymbol{\varepsilon}^*).$$

Thus

$$\begin{aligned} d &\ge D(\boldsymbol{\varepsilon}) - L(r^*, \boldsymbol{g}^*, \boldsymbol{f}^*, \boldsymbol{\varepsilon}), \\ &\ge D(\boldsymbol{\varepsilon}) - \max_{\boldsymbol{f}_t \in \mathcal{F}_{s,t}(r^*)} L(r^*, \boldsymbol{g}^*, \boldsymbol{f}, \boldsymbol{\varepsilon}), \\ &= \max_{\boldsymbol{0}\le\boldsymbol{g}\le\boldsymbol{c}} U_1(\boldsymbol{g}) + \max_{r\in[r_0,r_1]} U_2(r) - U_1(\boldsymbol{g}^*) - U_2(r^*). \end{aligned}$$

■

Lemma 5 (Robustness of approximate maximizer): Let $f : \mathbb{R} \mapsto \mathbb{R}$ be a twice continuously differentiable and strictly concave function. Let

$$x^* \triangleq \arg\max_{x\in[\alpha,\beta]} f(x), \tag{65}$$

where $[\alpha, \beta]$ is a nonempty bounded interval. Then $\forall \epsilon > 0, \exists \delta > 0$, such that if

$$x \in [\alpha, \beta] \text{ and } f(x^*) - f(x) < \delta, \tag{66}$$

then

$$(x-x^*)^2 \le (f(x^*)-f(x))\,\frac{2(1+\epsilon)}{-\ddot{f}(x)}. \tag{67}$$

In the special case where f is quadratic, we have

$$(x-x^*)^2 \le (f(x^*)-f(x))\,\frac{2}{-\ddot{f}}. \tag{68}$$

Proof: Perform Taylor expansion of $f(x)$ around x^* up to the second order as

$$f(x) = f(x^*) + \dot{f}(x^*)(x-x^*) + \frac{1}{2}\ddot{f}(\bar{x})(x-x^*)^2,$$

where $\bar{x}$ is between x and x^*.

According to the first order optimality condition for constrained convex optimization [22],

$$\dot{f}(x^*)(x-x^*) \le 0, \quad \forall x \in [\alpha, \beta].$$

The strict concavity of $f(x)$ implies that $\ddot{f}(x) < 0$, $\forall x$. Hence

$$f(x^*) - f(x) \ge -\frac{1}{2}\ddot{f}(\bar{x})(x-x^*)^2, \tag{69}$$

$$\ge \frac{1}{2}m(x-x^*)^2, \tag{70}$$

where

$$m \equiv \inf_{x\in[\alpha,\beta]} \left\{-\ddot{f}(x)\right\} > 0.$$

For the special case where f is quadratic, the claim (68) follows from (70).

Next we establish the claim for the general case. By assumption, $\ddot{f}(x)$ is continuous at x^*. Hence $\forall \epsilon_1 > 0$, $\exists \delta_1(\epsilon_1) > 0$ such that

$$|x-x^*| < \delta_1(\epsilon_1) \Longrightarrow |\ddot{f}(x) - \ddot{f}(x^*)| < \epsilon_1. \tag{71}$$

Now for any $\epsilon > 0$, let

$$\epsilon_1 \equiv \frac{-\ddot{f}(x^*)\epsilon}{2+\epsilon} \tag{72}$$

$$\delta \equiv \frac{m\delta_1^2(\epsilon_1)}{2}. \tag{73}$$

Then if $x \in [\alpha, \beta]$ and $f(x^*) - f(x) < \delta$, from (70), we have

$$|\bar{x} - x^*| \le |x - x^*| < \delta_1(\epsilon_1). \tag{74}$$

From (71), we have

$$|\ddot{f}(x) - \ddot{f}(x^*)| < \frac{-\ddot{f}(x^*)\epsilon}{2+\epsilon}, \tag{75}$$

$$|\ddot{f}(\bar{x}) - \ddot{f}(x^*)| < \frac{-\ddot{f}(x^*)\epsilon}{2+\epsilon} \tag{76}$$

From (75),

$$\frac{\ddot{f}(x^*)\epsilon}{2+\epsilon} < \ddot{f}(x) - \ddot{f}(x^*), \tag{77}$$

and hence

$$\frac{2\ddot{f}(x^*)}{2+\epsilon} < \frac{\ddot{f}(x)}{1+\epsilon}. \tag{78}$$

From (76),

$$\ddot{f}(\bar{x}) < \ddot{f}(x^*) - \frac{\ddot{f}(x^*)\epsilon}{2+\epsilon} = \frac{2\ddot{f}(x^*)}{2+\epsilon}. \tag{79}$$

From (78),

$$\ddot{f}(\bar{x}) < \frac{\ddot{f}(x)}{1+\epsilon}. \tag{80}$$

Using (69), we have

$$f(x^*) - f(x) \ge -\frac{1}{2}\ddot{f}(\bar{x})(x-x^*)^2 \ge -\frac{\ddot{f}(x)(x-x^*)^2}{2(1+\epsilon)}.$$

This establishes (67). ■

Proof of Theorem 2: With $\boldsymbol{\varepsilon}$ fixed, $U_1(\boldsymbol{g})$ is a sum of $|E|$ uni-variable functions, where the function corresponding to $vw \in E$, denoted by $U_1^{vw}(g_{vw})$, involves g_{vw} only. Define

$$d_{vw} \triangleq \max_{0\le g_{vw}\le c_{vw}} U_1^{vw}(g_{vw}) - U_1^{vw}(g^*_{vw}), \tag{81}$$

$$d_r \triangleq \max_{r\in[r_0,r_1]} U_2(r) - U_2(r^*) \tag{82}$$

Thus from Lemma 4, we have

$$\sum_{vw\in E} d_{vw} + d_r \le d. \tag{83}$$

The claim (35) can be established by applying Lemma 5 with $x^* = r(\boldsymbol{\varepsilon})$ and $x = r^*$.

To establish (34), we apply Lemma 5 with $x^* = g_{vw}(\boldsymbol{\varepsilon})$ and $x = g^*_{vw}$, where $g_{vw}(\boldsymbol{\varepsilon})$ denotes the vw-entry of $\boldsymbol{g}(\boldsymbol{\varepsilon})$. Then, for any $\epsilon > 0$, $\exists \delta_{vw}(\epsilon) > 0$, such that

$$d_{vw} < \delta_{vw}(\epsilon) \Longrightarrow (g_{vw}(\boldsymbol{\varepsilon}) - g^*_{vw})^2 \le \frac{2(1+\epsilon)d_{vw}}{\ddot{p}_{vw}(g^*_{vw})}. \tag{84}$$

For any $\epsilon > 0$, let $\delta \triangleq \min_{vw\in E} \delta_{vw}(\epsilon)$. Then

$$d < \delta \Longrightarrow d_{vw} < \delta_{vw}(\epsilon), \forall vw \in E. \tag{85}$$

Consequently, $d < \delta$ implies that

$$\begin{aligned}
\|\boldsymbol{g}(\boldsymbol{\varepsilon}) - \boldsymbol{g}^*\|^2 &= \sum_{vw\in E} (g_{vw} - g^*_{vw})^2 \\
&\le \sum_{vw\in E} \frac{2(1+\epsilon)d_{vw}}{\ddot{p}_{vw}(g^*_{vw})} \\
&\le \frac{\sum_{vw\in E} 2(1+\epsilon)d_{vw}}{\min_{vw\in E} \ddot{p}_{vw}(g^*_{vw})} \\
&\le \frac{2(1+\epsilon)d}{\min_{vw\in E} \ddot{p}_{vw}(g^*_{vw})}
\end{aligned}$$

This establishes (34).

The claim (36) for the special case where the link cost functions p_{vw} are quadratic functions can be established similarly. ■

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. IT-49, no. 2, pp. 371–381, Feb. 2003.

[3] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. Int'l Symp. Information Theory*. Yokohama, Japan: IEEE, June 2003.

[4] S. Jaggi, P. A. Chou, and K. Jain, “Low complexity optimal algebraic multicast codes,” in *Proc. Int’l Symp. Information Theory*. Yokohama, Japan: IEEE, June 2003.

[5] P. Sander, S. Egner, and L. Tolhuizen, “Polynomial time algorithms for network information flow,” in *Symposium on Parallel Algorithms and Architectures (SPAA)*. San Diego, CA: ACM, June 2003, pp. 286–294.

[6] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. 41st Allerton Conf. Comm., Ctrl. and Comp.*, Monticello, IL, Oct. 2003.

[7] N. Z. Shor, *Minimization methods for non-differentiable functions*, ser. Springer Series in Computational Mathematics. Springer-Verlag, 1985, translated from Russian by K. C. Kiwiel and A. Ruszczyński.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Belmont, MA: Athena Scientific, 1997.

[9] S. H. Low and D. E. Lapsley, “Optimization flow control, i: basic algorithm and convergence,” *IEEE/ACM Trans. on Networking*, vol. 7, pp. 861–874, Dec. 1999.

[10] L. Xiao, M. Johansson, and S. Boyd, “Simultaneous routing and resource allocation via dual decomposition,” *IEEE Trans. Communications*, vol. 52, no. 7, pp. 1136–1144, July 2004.

[11] L. Chen, S. H. Low, and J. C. Doyle, “Joint congestion control and media access control design for wireless ad hoc networks,” in *Proc. INFOCOM*. Miami, FL: IEEE, Mar. 2005.

[12] M. Chiang, “Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control,” *IEEE J. Sel. Areas in Comm.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.

[13] J. Wang, L. Li, S. H. Low, and J. C. Doyle, “Cross-layer optimization in TCP/IP networks,” *IEEE/ACM Trans. Networking*, 2005.

[14] J. Yuan, Z. Li, W. Yu, and B. Li, “A cross-layer optimization framework for multicast in multi-hop wireless networks,” in *Proc. IEEE 1st Int’l Conf. on Wireless Internet (Wicon 2005)*, Visegrad-Budapest, Hungary, July 2005.

[15] Z. Li and B. Li, “Efficient and distributed computation of maximum multicast rates,” in *Proc. INFOCOM*, Miami, FL, Mar. 2005, pp. 1618–1628.

[16] D. S. Lun, M. Medard, T. Ho, and R. Koetter, “Network coding with a cost criterion,” Massachussete Institute of Technology, Cambridge, MA, Tech. Rep. LIDS-P-2584, Apr. 2004.

[17] ——, “Network coding with a cost criterion,” in *Proc. Int’l. Symp. Information Theory and its Applications*. IEEE, Oct. 2004.

[18] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, “Achieving minimum-cost multicast: A decentralized approach based on network coding,” in *Proc. INFOCOM*. Miami, FL: IEEE, Mar. 2005.

[19] H. D. Sherali and G. Choi, “Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear problems,” *Operations Research Letters*, pp. 105–113, 1996.

[20] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” *IEEE J. Sel. Areas in Comm.*, vol. 23, no. 1, pp. 136–150, Jan. 2005.

[21] Y. Wu, P. A. Chou, and S.-Y. Kung, “Minimum-energy multicast in mobile ad hoc networks using network coding,” *IEEE Trans. Communications*, vol. 53, no. 11, pp. 1906–1918, Nov. 2005.

[22] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[23] A. V. Goldberg and R. E. Tarjan, “A new approach to the maximum flow problem,” *J. of the ACM*, vol. 35, no. 4, pp. 921–940, Oct. 1988.

[24] Y. Wu, M. Chiang, and S.-Y. Kung, “Distributed utility maximization for network coding based multicasting: a critical cut approach,” in *Proc. 2nd Workshop on Network Coding, Theory, and Applications*, Apr. 2006.

[25] C. D. Ha, “A generalization of the proximal point algorithm,” *SIAM J. on Control and Optimization*, vol. 28, pp. 503–512, 1990.

[26] D. P. Bertsekas and P. Tseng, “Partial proximal minimization algorithms for convex programming,” *SIAM J. Opt.*, vol. 4, pp. 551–572, 1994.

[27] R. W. Yeung, “Multilevel diversity coding with distortion,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 412–422, Mar. 1995.

[28] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[29] Y. Wu, “Network coding for multicasting,” Ph.D. dissertation, Princeton University, Nov. 2005, http://research.microsoft.com/∼yunnanwu/.

[30] Y. Wu, Q. Zhang, W. Zhu, and S.-Y. Kung, “Bounding the power rate function of wireless ad hoc networks,” in *Proc. INFOCOM*, Miami, FL, Mar. 2005.

[31] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with Rocketfuel,” in *Proc. SIGCOMM*. Pittsburg, PA: ACM, Aug. 2002.

[32] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[33] Y. Nesterov, *Introductory lectures on convex optimization*. Kluwer Academic, 2004.

PLACE
PHOTO
HERE

Yunnan Wu (S’02-M’06) received the B.SE. degree in computer science from the University of Science and the Technology of China, Hefei, China, in 2000, and the M.A. and the Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 2002 and 2006, respectively.

Since 2005, he has been a researcher with Microsoft Research, Redmond, WA. He was with Microsoft Research, Asia, Beijing, China, from 1999 to 2001 as a Research Assistant with Bell Laboratory, Lucent Technologies, as a Summer Intern in 2002, and with Microsoft Research, Redmond, WA, as a Summer Intern in 2003. His research interests include networking, information theory, graph theory, wireless communications, signal processing, and multimedia communications.

Dr. Wu received the Best Student Paper Award in the 2000 SPIE and IS&T Visual Communication and Image Processing Conference, and the Student Paper Award in the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing. He is a recipient of a Microsoft Research Graduate Fellowship for 20032005.

PLACE
PHOTO
HERE

Sun-Yuan Kung (S’74-M’78-SM’84-F’88) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1977.

In 1974, he was an Associate Engineer with Amdahl Corporation, Sunnyvale, CA. From 1977 to 1987, he was a Professor of electrical engineering systems with the University of Southern California, Los Angeles. Since 1987, he has been a Professor of electrical engineering at Princeton University, Princeton, NJ. He held a Visiting Professorship at Stanford University and the Delft University of Technology, Delft, The Netherlands, in 1984, a Toshiba Chair Professorship at Waseda University, Japan, in 1984, an Honorary Professorship at Central China University of Science and Technology, in 1994, and a Distinguished Chair Professorship at the Hong Kong Polytechnic University since 2001. He has coauthored more than 400 technical publications and numerous textbooks, including *VLSI Array Processors* (Englewood Cliffs, NJ: Prentice-Hall, 1988), *Digital Neural Networks* (Englewood Cliffs, NJ: Prentice-Hall, 1993), *Principal Component Neural Networks* (New York: Wiley, 1996), and *Biometric Authentication: A Machine Learning and Neural Network Approach* (Englewood Cliffs, NJ: Prentice-Hall, 2004). His research interests include VLSI array processors, system identification, neural networks, wireless communication, multimedia signal processing, bioinformatic data mining, and biometric authentication. Since 1990, he has been the Editor-in-Chief of the Journal of VLSI Signal Processing Systems.

Prof. Kung was a recipient of the IEEE Signal Processing Societys Technical Achievement Award for his contributions on parallel processing and neural network algorithms for signal processing in 1992, the IEEE Signal Processing Societys Best Paper Award for his publication on principal component neural networks in 1996, and the IEEE Third Millennium Medal in 2000. He was a Distinguished Lecturer of the IEEE Signal Processing Society in 1994.