# Content Distribution by Multiple Multicast Trees and Intersession Cooperation: Optimal Algorithms and Approximations

Xiaoying Zheng, Chunglae Cho and Ye Xia
Computer and Information Science and Engineering Department
University of Florida
Email: {xiazheng, ccho, yx1}@cise.ufl.edu

***Abstract*—The paper addresses the problem of massive content distribution in the network where multiple sessions coexist. In the traditional approaches, the sessions form separate overlay networks and operate independently from each other. In this case, some sessions may suffer from insufficient resources (e.g., aggregate upload bandwidth) even though other sessions have excessive resources. To cope with this problem, we consider the *universal swarming* approach, which allows multiple sessions cooperate with each other by forming a shared overlay network. We formulate the problem of finding the optimal resource allocation to maximize the sum of the session utilities under the network capacity constraints. The solution turns out to be optimal sharing of multiple minimum-cost multicast trees. We first present a subgradient algorithm and prove that, although the algorithm uses a single multicast tree per session at each iteration and hence does not converge in the conventional sense, it converges to the optimal solution in the time-average sense. The solution involves an NP-hard subproblem of finding a minimum-cost Steiner tree. We cope with this difficulty by using a column generation method, which reduces the number of Steiner-tree computation. Furthermore, we allow the use of approximate solutions to the Steiner-tree subproblem. We show that the approximation ratio to the overall problem turns out to be the same as that to the Steiner-tree subproblem. We give some experimental results showing that universal swarming improves the performance of resource-poor sessions with negligible impact to resource-rich sessions.***

## I. Introduction

The Internet is being applied to transfer content on a more and more massive scale. While many content distribution techniques have been introduced, most of the recently introductions are based on the *swarming* technique, such as BitTorrent [1], FastReplica [2], Bullet [3], [4], Chunkcast [5], and CoBlitz [6]. In a swarming session, the file to be distributed is broken into many chunks at the source node, which are then spread out in differential amount to the receivers; the receivers will then complete the download by exchanging the chunks with each other. By taking advantage of the resources of the receivers, swarming dramatically improves the distribution efficiency (e.g., average downloading rate, completion time) compared to the traditional client-server-based approach.

The problem addressed in this paper is how to conduct massive content distribution more efficiently in the network where multiple sessions coexist with various resource capacities, such as the source upload bandwidth, receiver download bandwidth, or aggregate upload bandwidth. For instance, there may exist some sessions with excessive aggregate upload bandwidth because their throughput bottleneck is at the source upload bandwidth, the receiver download bandwidth, or the internal network; at the same time, there may exist some other sessions whose throughput bottleneck is at their aggregate upload bandwidth. In traditional swarming approaches, the sessions operate independently by each forming a separate overlay network; this is called *separate swarming*, which does not provide the opportunity for the resource-poor sessions to use the surplus resource of the resource-rich sessions. However, if we conduct *universal swarming*, that is, we combine the sessions together into a single “super session” on a shared overlay network and allow them to share each other’s resources, the distribution efficiency of the resource-poor sessions can improve greatly with negligible impact on the resource-rich sessions.

A fundamental way to understand swarming is to follow the distribution paths of individual chunks. With some thought, it can be seen that each chunk is distributed over a tree rooted at the source and covering all the receivers of the file. Since the chunks travel down different trees, swarming can be thought as distribution over multiple multicast trees, if the delay is not modeled. In universal swarming, a distribution tree not only includes all the receivers interested in downloading the file but may also contain nodes that are not interested in the file; the latter are called *out-of-session nodes*. Thus, a distribution tree for a session is a *Steiner* tree rooted at the source where all the receivers are terminal nodes and the out-of-session nodes on the tree are Steiner nodes.

One of the interesting problems is how to find an optimal rate allocation on the multiple multicast trees so that it achieves the optimal performance objective. A similar problem was addressed in [7] in the context of separate swarming. The rate-allocation problem in universal swarming is substantially more difficult. The main reason is that, by the optimality condition, an optimal solution typically uses only the minimum-cost trees to distribute the file chunks. Hence, optimal universal swarming algorithms usually involves an NP-hard subproblem of finding a minimum-cost Steiner tree. How to cope with this issue is one of the main themes in this paper.

We present two solution approaches, which can be used in combination. First, we introduce into our rate-allocation algorithm a column-generation method, which can reduce the number of times the min-cost Steiner-tree is computed. Second, we allow the use of approximate solutions to the Steiner-tree subproblem, which makes it tractable. Some approximate solutions to the Steiner-tree problem in directed graphs can be found in [8]–[10]. Importantly, we show that the approximation ratio to the overall rate-allocation problem turns out to be the same as that to the Steiner-tree subproblem.

The overall rate-allocation algorithm that we will present is a subgradient algorithm. It has the characteristic of using a single multicast tree per session at each iteration; the rate assigned to the tree is computed based on the link prices at the iteration. We can show that even though the rate assigned to the tree in each iteration usually exceeds the capacity of some links on the tree, the time-average rate satisfies the link capacity constraints, and eventually the rate allocation to each session converges to the optimum (provided the Steiner-tree subproblem is solved optimally.) It is worth pointing out that other optimization algorithms may also be used here instead of the subgradient algorithm.

We now briefly discuss addition related work. A heuristic centralized algorithm for the multicast tree packing problem is proposed in [11]. Jansen [12] presents a centralized approximation algorithm for the multicast congestion control problem. [13]–[15] apply the network coding technique to achieve the multicast capacity; part of their solution techniques is similar to ours. A survey of optimization problems in multicast routing can be found in [16]. [17]–[19] model and analyze the peer-assisted file distribution system. The multipath routing problem has been studied in [20]–[23].

The paper is organized as follows. The formal problem description is given in Section II. The subgradient algorithm and its convergence proof are given in Section III. In Section IV, we present the column generation approach, combine it with the subgradient algorithm, and study the performance bound when approximation algorithms are applied to the minimum-cost tree subproblem. We show some simulation results about our approach in Section V. The conclusion is drawn in Section VI.

## II. Problem Description

Let the network be represented by a directed graph $G = (V, E)$, where $V$ is the set of nodes and $E$ is the set of links. For each link $e \in E$, $c_e > 0$ is its capacity. A *multicast session* consists of the source node and all the receivers corresponding to a file. Let $s$ denote a session or the source of a session interchangeably. In a session $s$, the data traffic is routed along multiple multicast trees, each rooted at the source $s$ and covering all the receivers. A multicast tree is a *Steiner* tree; it may contain nodes not in the session, which are called Steiner nodes. Let the set of all allowed multicast trees for session $s$ be denoted by $T_s$. Throughout the paper, we assume $T_s$ contains all possible multicast trees unless specified otherwise. Let $S$ be the set of all multicast sessions, then $T = \cup_{s\in S} T_s$ is the collection of all multicast trees for all sessions. The multicast trees can be indexed in an arbitrary order as $t_1, t_2, \cdots, t_{|T|}$, where $|\cdot|$ is the cardinality of a set. Though $|T|$ is finite, it might be exponential in the number of links. Let $x_s$ be the flow rate of session $s \in S$ and $y_t$ be the flow rate of a multicast tree $t$. We have $x_s = \sum_{t\in T_s} y_t$.

Each session $s \in S$ is associated with a utility function $U_s(x_s), 0 \le m_s \le x_s \le M_s$. The assumption on the utility functions is, for every $s \in S$,

- $A1$: $U_s$ is well-defined (real-valued), non-decreasing, strictly concave on $[m_s, M_s]$, and twice continuously differentiable on $(m_s, M_s)$.

The problem is to find the optimal resource (i.e., session and multicast-tree rates) allocation to maximize the sum of session utilities under the capacity constraints and session rate constraints. The problem is formulated as

$$\begin{aligned} &\max \quad f(x,y) = \textstyle\sum_{s\in S} U_s(x_s) && && (1)\\ \text{s.t.} \quad & x_s = \textstyle\sum_{t\in T_s} y_t, && \forall s \in S\\ & \textstyle\sum_{t\in T: e\in t} y_t \le c_e, && \forall e \in E && (2)\\ & m_s \le x_s \le M_s, && \forall s \in S\\ & y_t \ge 0, && \forall t \in T. \end{aligned}$$

Without loss of generality, we make an assumption on the problem (1).

- $A2$: There exists a feasible solution $(\bar{x}, \bar{y})$ such that $m_s \le \bar{x}_s \le M_s$ for any session $s \in S$ and $f(\bar{x}, \bar{y}) > -\infty$ and (2) holds with strict inequality at $(\bar{x}, \bar{y})$.

Note that $f(x, y)$ is strictly concave on $x$, but linear on $y$. Let $\lambda_e$ be the Lagrangian multiplier associated with the constraint (2). The Lagrangian function of (1) is

$$\begin{aligned} L(x,y,\lambda) &= \sum_{s\in S} U_s(x_s) + \sum_{e\in E} \lambda_e (c_e - \sum_{t\in T: e\in t} y_t)\\ &= \sum_{s\in S} (U_s(x_s) - \sum_{t\in T_s} y_t \sum_{e\in t} \lambda_e) + \sum_{e\in E} \lambda_e c_e. \quad (3) \end{aligned}$$

The dual function is

$$\begin{aligned} \theta(\lambda) = \quad & \max \quad L(x,y,\lambda) && (4)\\ \text{s.t.} \quad & x_s = \textstyle\sum_{t\in T_s} y_t, \quad \forall s \in S\\ & m_s \le x_s \le M_s, \quad \forall s \in S\\ & y_t \ge 0, \quad \forall t \in T. \end{aligned}$$

Now the dual problem of (1) is

$$\begin{aligned} \text{Dual:} \quad & \min \quad \theta(\lambda) && (5)\\ \text{s.t.} \quad & \lambda \ge 0. \end{aligned}$$

## III. A Distributed Algorithm

In this section, we will illustrate how the problem (1) can be solved by a distributed subgradient algorithm.

### A. Subgradient Algorithm

The dual problem (5) can be solved by a standard subgradient method as in Algorithm 1, where $\delta_e(k)$ is a positive scalar step size, $[\cdot]_+$ and $[\cdot]_{m_s}^{M_s}$ denote the projection onto the non-negative domain and the interval of $[m_s, M_s]$, respectively [24] [25]. There are two step size rules:

- Rule I (Constant step size): $\delta_e(k) = \delta > 0$, for all time $k \geq K$ for some finite $K$.
- Rule II (Diminishing step size): $\delta_e(k) \leq \delta_e(k-1)$ for all time $k \geq K$ for some finite $K$. $\lim_{k\to\infty} \delta_e(k) = 0$ and $\lim_{k\to\infty} \sum_{u=0}^{k} \delta_e(u) = \infty$.

At the update (9) and (10) of Algorithm 1, we need to compute a minimum-cost Steiner tree problem. Under any fixed dual cost vector $\lambda \geq 0$, for any session $s \in S$, define a min-cost Steiner tree by

$$t(s,\lambda) = \text{argmin}_{t\in T_s}\{\sum_{e\in t} \lambda_e\}, \tag{6}$$

where the tie is broken arbitrarily. Because (6) is an optimization problem over all allowed trees, we call (6) *a global min-cost tree problem*, and the achieved minimum cost *the global minimum tree cost*. We denote this global minimum tree cost under a fixed $\lambda \geq 0$ by

$$\gamma(s,\lambda) = \sum_{e\in t(s,\lambda)} \lambda_e. \tag{7}$$

**Algorithm 1** Subgradient Algorithm

$$\lambda_e(k+1) = [\lambda_e(k) - \delta_e(k)(c_e - \sum_{t\in T: e\in t} y_t(k))]_+, \forall e \in E \tag{8}$$

$$x_s(k+1) = [(U_s')^{-1}(\gamma(s,\lambda(k+1)))]_{m_s}^{M_s}, \forall s \in S \tag{9}$$

$$y_t(k+1) = \begin{cases} x_s(k+1) & \text{if } t = t(s,\lambda(k+1)); \\ 0 & \text{otherwise,} \end{cases} \quad \forall t \in T. \tag{10}$$

**Remark**: Algorithm 1 is a distributed algorithm. In order to compute the tree cost, each link $e$ can independently compute its dual cost $\lambda_e$ based on the local aggregate rate passing through the link. Then, the tree cost can be accumulated by the source $s$ based on the link cost values along the tree. To find the minimum-cost tree $t(s,\lambda(k))$, each source needs to compute the minimum-cost directed Steiner tree, which is an NP-hard problem. We will address this issue and propose an approximation solution in Section IV. Other than that, the subgradient algorithm is completely decentralized.

### B. Convergence Results

Let $\Lambda^* = \{\lambda \geq 0 : \theta(\lambda) = \min_{\lambda\geq 0} \theta(\lambda)\}$ be the set of optimal dual variables. Let $f^*$ be the optimal function value of the problem (1) and $(x^*, y^*, \lambda^*)$ denote one of the optimal primal-dual solutions. Obviously, $f^*$ is bounded.

*Lemma 1:* Under assumptions $A1$ and $A2$,

- $(a)$ There is no duality gap between the primal problem (1) and the dual problem (5), i.e., $f^* = \theta(\lambda^*)$ for any $\lambda^* \in \Lambda^*$.
- $(b)$ For any $\lambda \geq 0$, $(x, y)$ obtained by (9) and (10) are one of the Lagrangian maximizers with the Lagrangian multiplier $\lambda$. Furthermore, $x$ obtained by (9) is the unique Lagrangian maximizer.
- $(c)$ For any $\lambda^* \in \Lambda^*$, the solution obtained by (9) is the unique optimal solution $x^*$ of (1).
- $(d)$ $\Lambda^*$ is a non-empty compact set.

*Proof:* See Appendix A. ■

*Theorem 2:* Let $d(\lambda, \Lambda^*) = \min_{\lambda^*\in\Lambda^*} ||\lambda - \lambda^*||$. For any $\epsilon > 0$, under both the step size rule I and II, there exist a sequence of step size $\{\delta(k)\}$ and a sufficiently large $K_0$ such that, with any initial $\lambda(0) \geq 0$, for all $k \geq K_0$, $d(\lambda(k), \Lambda^*) < \epsilon$ and $||x(k) - x^*|| < \epsilon$ [26].

*Proof:* The proof is standard and is omitted. ■

We now discuss the convergence of the tree rate vector $y(k)$. The difficulty of proving the convergence of $y(k)$ arises from the linearity of the Lagrangian function in (3) on the vector $y$, and there is no standard result about the convergence of $y(k)$. In fact, the tree rate vector $y(k)$ does not converge in the normal sense [27]. From the update (10), we see that the source only uses one tree each time and shifts flow from one tree to another from time to time. We further noticed that, by pushing the session flow onto only one tree at a time, the link capacity constraints are often violated. This means the resulted distribution solution on each time slot may not even be feasible. In the following proofs, we will show that the tree rate converges in the time average sense. In Theorem 3 to Theorem 5, $0 \leq k_0 < \infty$ is a finite starting time. Theorem 3 to Theorem 5 hold under both the step size rule I and II.

*Theorem 3:* For any link $e$ and a finite time $k_0$, there exists a constant $M_e < \infty$, [1] such that for any time $k$,

$$\sum_{u=k_0}^{k} \sum_{t\in T: e\in t} y_t(u) \leq c_e(k - k_0 + 1) + M_e.$$

*Proof:* See Appendix A. ■

Let $H$ denote the $|E| \times |T|$ link-tree incidence matrix which associates the trees with the links (i.e., $[H]_{et} = 1$ if $e \in t$; and $[H]_{et} = 0$ otherwise). Let $A$ denote the $|S| \times |T|$ session-tree incidence matrix which associates the sessions with the trees in $T$ (i.e., $[A]_{st} = 1$ if $t \in T_s$; and $[A]_{st} = 0$ otherwise). For a fixed finite $k_0$, let define a sequence $\{\bar{y}(k)\}$ for shorthand, where

$$\bar{y}(k) = \frac{\sum_{u=k_0}^{k} y(u)}{k - k_0 + 1}. \tag{11}$$

*Theorem 4:*

$$\lim_{k\to\infty} \sup H\bar{y}(k) \leq c.$$

Furthermore, for any limit point $\bar{y}^*$ of the sequence $\{\bar{y}(k)\}$, $H\bar{y}^* \leq c$.

*Proof:* See Appendix A. ■

[1] $M_e$ only depends on $k_0$ and is independent of $k$.

For any $\epsilon > 0$, let define $\mathcal{Y}^*(\epsilon) = \{y \geq 0 : Hy \leq c, ||Ay - x^*|| < \epsilon\}$. When $\epsilon = 0$, $\mathcal{Y}^*(0) = \mathcal{Y}^* = \{y \geq 0 : Hy \leq c, Ay = x^*\}$ is the set of optimal tree rate allocation.

*Theorem 5:* For any $\epsilon > 0$, with any initial $\lambda(0) \geq 0$, every limit point of the sequence $\{\bar{y}(k)\}$ is in the set $\mathcal{Y}^*(\epsilon)$.

*Proof:* See Appendix A. ■

**Remark**: By Theorem 5, the time average tree rate vector $\bar{y}(k)$ converges to one optimal tree rate vector $y^*$.

## IV. Column Generation Method With Imperfect Global Min-cost Tree Scheduling

In Section III, we develop a distributed algorithm to solve the problem (1), if the min-cost Steiner tree problem (6) can be solved precisely in a distributed fashion. However the subproblem (6) is an NP-hard problem [28]. The column generation method with imperfect tree scheduling can be introduced to solve the overall problem approximately. The column generation part reduces the number of times when the min-cost Steiner tree problem is invoked. Imperfect tree scheduling applies fast approximation or heuristic algorithms to the Steiner tree problem. This column generation method with approximation was first proposed in [29] to solve the problem of wireless link scheduling.

### A. Column Generation Method

The main idea of column generation is to start with a subset of the tree set $T$ and bring in new trees only when needed. Consider a subset of $T$ containing only a small number of trees, i.e., $T^{(q)} = \{t_i \in T : \forall i = 1, \cdots, q\}$. We can formulate the following restricted master problem (RMP) for $T^{(q)}$.

$$\begin{array}{llll} q^{th}\text{-RMP:} & \max \sum_{s\in S} U_s(x_s) & & (12) \\ \text{s.t.} & x_s = \sum_{t\in T_s^{(q)}} y_t, & \forall s \in S & \\ & \sum_{t\in T^{(q)}:e\in t} y_t \leq c_e, & \forall e \in E & (13) \\ & m_s \leq x_s \leq M_s, & \forall s \in S & \\ & y_t \geq 0, & \forall t \in T^{(q)}. & \end{array}$$

The value of $q$ is usually small and the trees in the set $T^{(q)}$ can be examined one-by-one. The Lagrangian function, the dual function, and the dual problem of the $q^{th}$-RMP can be formulated similarly as in (3), (4), and (5), where the set $T$ is replaced by the set $T^{(q)}$.

The $q^{th}$-RMP is more restricted than the MP. Thus, any optimal solution to the $q^{th}$-RMP is feasible to the MP and serves as a lower bound of the optimal value of the MP. By gradually introducing more trees (columns) into $T^{(q)}$ and expanding the subset $T^{(q)}$, we will improve the lower bound of the MP [30]–[32].

### B. Apply the Subgradient Algorithm to the RMP

The distributed subgradient algorithm can be used to solve the $q^{th}$-RMP. Here, we define the following problem of finding the min-cost tree $t^{(q)}(s, \lambda)$ under the link cost vector $\lambda \geq 0$.

$$t^{(q)}(s, \lambda) = \text{argmin}_{t\in T_s^{(q)}} \{\sum_{e\in t} \lambda_e\}, \tag{14}$$

The optimization is taken over the $|T_s^{(q)}|$ currently known trees. The problem in (14) is called the *local min-cost tree problem*, and the achieved minimum cost is called the *local minimum tree cost*. We denote this local minimum cost under $\lambda \geq 0$ by

$$\gamma^{(q)}(s, \lambda) = \sum_{e\in t^{(q)}(s,\lambda)} \lambda_e. \tag{15}$$

If there are more than one tree achieving the local minimum cost, the tie is broken arbitrarily.

### C. Introduce One More Tree (Column)

Now the question is how to check whether the optimum of the $q^{th}$-RMP is optimal for the MP, and if not, how to introduce a new column (tree). It turns out there is an easy way to do both. Let $(\bar{x}^{(q)}, \bar{y}^{(q)}, \bar{\lambda}^{(q)})$ denote one of the optimal primal-dual solutions of the $q^{th}$-RMP.

*Lemma 6:* $(\bar{x}^{(q)}, \bar{y}^{(q)}, \bar{\lambda}^{(q)})$ is optimal to the MP if and only if $h_s(\gamma(s, \bar{y}^{(q)})) = h_s(\gamma^{(q)}(s, \bar{y}^{(q)}))$, for all $s \in S$, where

$$h_s(w) = U_s([(U_s')^{-1}(w)]_{m_s}^{M_s}) - [(U_s')^{-1}(w)]_{m_s}^{M_s} \cdot w, \ w \geq 0.$$

*Proof:* See Appendix B. ■

From Lemma 6, if the local minimum tree cost is equal to the global minimum tree cost (i.e., $\gamma(s, \bar{\lambda}^{(q)}) = \gamma^{(q)}(s, \bar{\lambda}^{(q)})$), then $h_s(\gamma(s, \bar{\lambda}^{(q)})) = h_s(\gamma^{(q)}(s, \bar{\lambda}^{(q)}))$; otherwise, $T$ is not sufficiently well characterized by $T^{(q)}$ and a new tree should be added to the RMP. We state the rule of introducing a new column in the following.

*Fact 7:* Any tree achieving a cost less than the local minimum tree cost could enter the subset $T^{(q)}$ in the RMP. The tree achieving the global minimum tree cost is one possible candidate and is often preferred [29].

### D. Column Generation by Imperfect Global Tree scheduling

The min-cost Steiner tree problem (6) is NP-hard, which makes the step of column generation very difficult. However, according to Fact 7, we do not have to solve it precisely. Instead, we may solve it approximately, and this is referred as *imperfect global tree scheduling*. [2]

Suppose we are able to solve (6) with an approximation ratio $\rho \geq 1$, i.e.,

$$\frac{1}{\rho}\gamma_\rho(s, \lambda) \leq \gamma(s, \lambda), \tag{16}$$

where $\gamma_\rho(s, \lambda)$ is the cost of the tree given by the approximate solution. Note that both $\gamma(s, \lambda)$ and $\gamma_\rho(s, \lambda)$ are non-negative for all vectors $\lambda \geq 0$.

*1) A $\rho$-approximation Approach:* We develop a column generation method with imperfect global min-cost tree scheduling as follows. Later, we will show a guaranteed performance bound of this approach. Algorithm 2 was originally proposed in [29], and in fact describes a whole class of algorithms representing different performance, convergence speed and complex tradeoffs. More detailed comments about the property of this class of algorithms can be found in [29].

[2] Note that the local min-cost tree problem (14) can be easily solved precisely since the number of extreme points (i.e., candidate trees) of $T^{(q)}$ is usually small, and hence, enumerable.

**Algorithm 2** Column Generation with Imperfect Global Tree scheduling

```
• Initialize: Start with a collection of $T^{(q)}$ trees. Assume
  Assumption $A2$ holds on this initial set $T^{(q)}$.
• Step 1: Run the subgradient algorithm (8)-(10) for several
  (a finite number) times on the $q^{th}$-RMP.
• Step 2: For each source, solve the global min-cost tree
  problem (6) with approximation ratio $\rho$ under the current
  dual cost $\lambda$.
    – If the tree corresponding to the approximate solution
      of (6) is already in the current collection of trees, go
      to Step 1;
    – otherwise, introduce this tree into the current collection
      of trees, increase $q$ by 1, and go to Step 1.
```

**Remark** 1: If the approximate tree derived in step 2 has a higher tree cost than that of the local min-cost tree among the existing trees already selected, we define the existing tree with the lowest cost as the solution to the approximation algorithm. Hence, the cost of the imperfect (approximate) tree cannot be higher than any of the existing trees.

**Remark** 2: Note that since the set of $T^{(q)}$ is usually small and enumerable, it is possible for a source to manage its current collection of trees. In order to compute the cost of each known tree, each link $e$ can independently compute its link dual cost based on its aggregate link flow rate. Then, those components of the tree cost can be propagated back to the source. The source can compute the cost of each known tree and the local min-cost tree. Furthermore, if the global min-cost tree problem (6) can be solved approximately in a decentralized fashion, then Algorithm 2 is completely decentralized. In Section V, we will introduce some approximation algorithms for min-cost Steiner tree problem.

### *2) Convergence with Imperfect Global Tree scheduling:*

*Theorem 8:* There exists a $q$, $1 \leq q \leq |T|$, such that Algorithm 2 converges to one optimal primal-dual solution of this particular $q^{th}$-RMP, i.e., $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$. Furthermore, after Algorithm 2 converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$, $\gamma_\rho(s, \bar{\lambda}^{(q)}) = \gamma^{(q)}(s, \bar{\lambda}^{(q)})$ for any source $s \in S$.

*Proof:* The proof follows the counterpart in [29]. ■

### *3) Performance Bound under Imperfect Tree scheduling:*

Theorem 8 says that the column generation method with imperfect global tree scheduling converges to a sub-optimum of the MP. Next, we will prove that the performance of this sub-optimum is bounded. We make the assumptions $A3$ and $A4$.

- $A3$: For any source $s \in S$, $m_s \geq 0$ is sufficiently small such that, if the column generation method with imperfect global tree scheduling converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$ on the $q^{th}$-RMP, then $\bar{x}_s^{(q)} > m_s$.
- $A4$: $U_s(m_s) - m_s \cdot U_s'(m_s) \geq 0, \forall s \in S$.

*Theorem 9 (Bound of Imperfect Global Tree scheduling):* Under the additional assumptions $A3$ and $A4$, if the column generation method with imperfect global tree scheduling converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$ on the $q^{th}$-RMP, we have

$$\theta^{(q)}(\bar{\lambda}^{(q)}) \leq \sum_{s \in S} U_s(x_s^*) \leq \theta(\rho\bar{\lambda}^{(q)}) \leq \rho\theta^{(q)}(\bar{\lambda}^{(q)}). \quad (17)$$

*Proof:* See Appendix B. ■

Since the strong duality holds on the $q^{th}$-RMP, $\sum_{s \in S} U_s(\bar{x}_s^{(q)}) = \theta^{(q)}(\bar{\lambda}^{(q)})$, we have the following.

*Corollary 10 ($\rho$-Approximation Solution to the MP):* Under the additional assumptions $A3$ and $A4$, we have

$$\sum_{s \in S} U_s(\bar{x}_s^{(q)}) \leq \sum_{s \in S} U_s(x_s^*) \leq \rho \sum_{s \in S} U_s(\bar{x}_s^{(q)}). \quad (18)$$

If $\rho = 1.0$, (18) holds with equality, then Algorithm 2 is the column generation method with perfect global min-cost tree, and this algorithm converges to one optimum of MP.

Corollary 10 says that the column generation method with imperfect global tree scheduling converges to a sub-optimum of the MP and achieves the same approximation ratio as the approximate solution to the global min-cost tree problem.

**Remark:** The possible utility functions could be $U_s(x_s) = w_s \ln(x_s + e)$ and $U_s(x_s) = \frac{w_s}{1-\beta} x_s^{1-\beta}$, where $0 < \beta < 1$ and $w_s > 0$.

## V. Illustrative examples

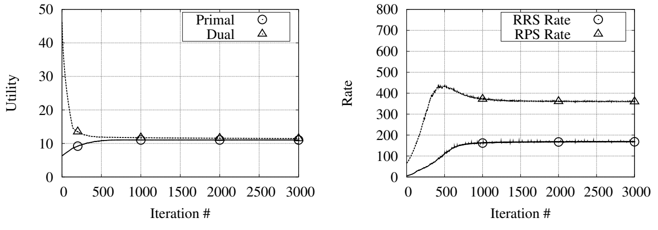
In this section, we give illustrative examples showing the effect of universal swarming and the performance of our algorithms. We show that the subgradient algorithm achieves the optimum in the time-average sense. We also show that the column generation method can be used for reducing the number of global min-cost tree computation.

We test our algorithms in various scenarios by varying the sizes of the resource-rich and resource-poor sessions and the bottleneck point in the network. We have nine test cases where we assume the internal network has large capacity so that it cannot be the bottleneck; therefore, the bottleneck lies on the access links. In each profile $A1$, $A2$ and $A3$, we have a large resource-rich session(RRS) and a small resource-poor session(RPS); in profile $B1$, $B2$ and $B3$, we have the equal-sized RRS and RPS; and in profile $C1$, $C2$ and $C3$, we have a small RRS and a large RPS. Each large session contains 90 receivers; each small session contains 10 receivers; and each medium session contains 50 receivers. Each session has a single source. We also vary the bottleneck of the sessions so that we can examine how the intersession cooperation affects the rate allocation in each case. In profiles $A1$, $B1$ and $C1$, the bottleneck of the RRS is at the download links; in profile $A2$, $B2$ and $C2$, the bottleneck of the RRS is at the upload link of its source; and in profile $A3$, $B3$ and $C3$, the RRS is bottlenecked by its aggregate upload bandwidth. In all cases, the RPS is bottlenecked at its aggregate upload bandwidth. Note that if the bottleneck of the RPS is at its source upload link or the receiver download links, then there is no way to improve its session rate.

In each test case, we compare the rate allocation results of the separate swarming with that of the universal swarming. For the separate swarming, we use the subgradient algorithm with

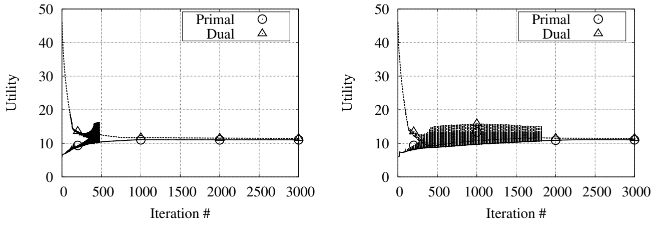
TABLE I
COMPARISON OF RATE ALLOCATION BETWEEN SEPARATE SWARMING AND UNIVERSAL SWARMING

| Test cases | | Link bandwidth | | | Rate allocation | |
|---|---|---|---|---|---|---|
| Profile | Session | $u_s$ | $u_i$ | $d_i$ | Separate | Universal |
| A1 | Large RRS | 640 | 360 | 360 | 360 | 329.5 |
| | Small RPS | 640 | 36 | 360 | 100 | 359.7 |
| A2 | Large RRS | 280 | 360 | 360 | 280 | 280 |
| | Small RPS | 280 | 36 | 360 | 64 | 280 |
| A3 | Large RRS | 640 | 200 | 360 | 207 | 170.2 |
| | Small RPS | 640 | 20 | 360 | 84 | 360 |
| B1 | Medium RRS | 640 | 360 | 360 | 360 | 205.6 |
| | Medium RPS | 640 | 36 | 360 | 48.8 | 201.4 |
| B2 | Medium RRS | 280 | 360 | 360 | 280 | 203.8 |
| | Medium RPS | 280 | 36 | 360 | 41.6 | 199.9 |
| B3 | Medium RRS | 640 | 200 | 360 | 212.8 | 125.6 |
| | Medium RPS | 640 | 20 | 360 | 32.8 | 123.2 |
| C1 | Small RRS | 640 | 360 | 360 | 360 | 353 |
| | Large RPS | 640 | 36 | 360 | 43.1 | 50.6 |
| C2 | Small RRS | 280 | 360 | 360 | 280 | 283 |
| | Large RPS | 280 | 36 | 360 | 39.1 | 51.2 |
| C3 | Small RRS | 640 | 200 | 360 | 264 | 263.8 |
| | Large RPS | 640 | 20 | 360 | 27.1 | 27.1 |



(a) (b)

Fig. 1. Convergence of the subgradient algorithm. (a) primal and dual function values versus iterations. (b) session rates versus iterations.



(a) (b)

Fig. 2. Primal and dual function values versus iterations with column generation method. (a) interval of global min-cost tree computation = 5. (b) interval of global min-cost tree computation = 20.

the minimum spanning tree solution. Note that if sessions are separated from each other, we can use a minimum spanning tree solution because the overlay network for each session contains no Steiner nodes. On the other hand, for the universal swarming, we use the algorithm by Charikar *et. al* with tree level 2, as proposed in [8], for getting an approximate minimum-cost tree solution.

Table I summarizes the simulation results for our test cases.[3] Let $u_s$, $u_i$, and $d_i$ be the source upload bandwidth, and each receiver's upload and download bandwidth, respectively. Note that the subgradient algorithm always achieves the optimal rate allocation in separate swarming.[4] The simulation results show that the RPS can obtain the excessive resource of the RRS at small expense of the RRS. When the small RPS is combined with the large RRS, its session rate improves significantly while the large RRS loses a little of its session rate. When the session sizes of the RRS and RPS are the same, the resulting session rates tend to be equalized, which is a desirable result. When the large RPS is combined with the small RRS, its session rate still improves slightly with negligible impact on the small RRS; this is also desirable since it implies that the RRS would not give up its resource if it is not sufficiently abundant.

Fig. 1 plots the primal and dual function values and session rates versus the subgradient iterations for profile $A3$. It shows that rate allocation converges as the primal and dual function values converge.[5]

We have also tested the column generation method with the same profiles. Fig. 2 shows the convergence of the primal and dual function values when we use the column generation method in profile $A3$ with different interval sizes for the global min-cost tree computation. Note that there is a trade-off in selecting the interval size. As the interval size increases, the number of iterations needed for convergence tends to increase while the number of global min-cost tree computation decreases. Therefore, if the global min-cost tree computation dominates the overall time complexity, then we should use a large interval size; on the other hand, if message communication dominates the overall cost and time complexity, we should use a small interval size.

## VI. CONCLUSION

This paper studies the universal swarming technique, which allows multiple sessions to help each other to speed up the overall distribution speed. In universal swarming, the data of each session is distributed by a set of multicast trees rooted at the source and spanning all the receivers. Each multicast tree is in general a Steiner tree containing out-of-session nodes. The question is how to optimally allocate rates to the multicast trees to maximize the sum of all sessions' utilities. A distributed subgradient algorithm is developed. Due to the partial linearity of the problem, there is no standard convergence result for the algorithm and the algorithm does

[3] In the simulation, we use $U_s(x_s) = \ln(x_s + e)$ as the utility function, and run the subgradient algorithm for 10000 iterations so that we reach convergence for all the cases. The step size rule and the initial step size used in each profile is slightly different from each other. It is hard to apply the same step size rule for all the profiles and reach convergence within 10000 iterations.

[4] In the separate swarming cases, the optimal rate of each session can be easily computed as $\min\{u_s, \min_{1\leq i\leq L} d_i, (u_s + \sum_{1\leq i\leq L} u_i)/L\}$ where $L$ is the number of receivers.

[5] In some test cases, we have experienced small oscillation of the allocated rates even though its primal and dual function values seem to converge. However, the time-average rates converge in all cases.

not converge in the normal sense. We prove that the subgradient converges in the time average sense. Furthermore, the subgradient algorithm involves an NP-hard subproblem of finding a minimum-cost Steiner tree. We adopt a column generation method with imperfect min-cost tree scheduling. If the imperfect min-cost tree has bounded performance, then our overall utility optimization algorithm converges to a suboptimum with bounded performance.

## VII. Appendix A: Proofs in Section III

*Proof of Lemma 1:* The proof of (a), (b) and (c) is standard and is omitted.

$(d)$: From part $(a)$, $\Lambda^*$ is non-empty. Suppose $\Lambda^*$ is not bounded, we can make $\theta(\lambda^*)$ arbitrarily large by choosing $\lambda^* \in \Lambda^*$ with large norm $||\lambda^*||$, since (2) holds with the strict inequality at $(\bar{x}, \bar{y})$ and $f(\bar{x}, \bar{y})$ is bounded from below under assumption $A2$. This contradicts with the fact that $\theta(\lambda^*) = f^*$ is bounded, hence $\Lambda^*$ is a non-empty compact set [33]. ■

*Proof of Theorem 3:* According to (8),

$$\frac{1}{\delta_e(k)}(\lambda_e(k+1) - \lambda_e(k)) \geq \sum_{t \in T: e \in t} y_t(k) - c_e, \forall e \in E.$$

Summing the above inequality from time slots $k_0$ to $k$, we have

$$\begin{aligned}
&\sum_{u=k_0}^{k} \sum_{t \in T: e \in t} y_t(u) \\
\leq& c_e(k - k_0 + 1) + \frac{1}{\delta_e(k)}\lambda_e(k+1) + \sum_{u=k_0+1}^{k} (\frac{1}{\delta_e(u-1)} \\
&- \frac{1}{\delta_e(u)})\lambda_e(u) - \frac{1}{\delta_e(k_0)}\lambda_e(k_0) \\
\leq& c_e(k - k_0 + 1) + \frac{1}{\delta_e(k)}\Delta_e + \sum_{u=k_0+1}^{k} (\frac{1}{\delta_e(u-1)} - \frac{1}{\delta_e(u)})\Delta_e \\
=& c_e(k - k_0 + 1) + \frac{1}{\delta_e(k_0)}\Delta_e \\
=& c_e(k - k_0 + 1) + M_e.
\end{aligned}$$

The first inequality is due to the rearrangement of the terms. Regarding to the second inequality, by Theorem 2, the sequence $\{\lambda(k)\}$ converges to the compact set $\Lambda^*$ under both step size rule I and II, which means there exists a large enough constant $0 < \Delta_e < \infty$ such that $\lambda_e(k) \leq \Delta_e$ for all time $k$. Note that $\frac{1}{\delta_e(u-1)} - \frac{1}{\delta_e(u)} \geq 0$ under step size rule I and II. The second inequality throws away the additional negative term $-\frac{1}{\delta_e(k_0)}\lambda_e(k_0)$ as well. The first equality holds due to the cancelation of the cross terms. The last equality holds since $k_0$ is finite and $\frac{1}{\delta_e(k_0)}$ is finite as well. ■

*Proof of Theorem 4:* By Theorem 3, for any link $e \in E$, there exists a constant $M_e < \infty$ such that

$$\sum_{t \in T: e \in t} \bar{y}_t(k) \leq c_e + \frac{M_e}{k - k_0 + 1}.$$

Taking the limits of the above inequality on both sides, it yields

$$\lim_{k \to \infty} \sup \sum_{t \in T: e \in t} \bar{y}_t(k) \leq \lim_{k \to \infty} (c_e + \frac{M_e}{k - k_0 + 1}) = c_e,$$

for any link $e \in E$. Equivalently,

$$\lim_{k \to \infty} \sup H\bar{y}(k) \leq c.$$

The sequence $\{\bar{y}(k)\}$ is a bounded sequence since $x(k)$ and $y(k)$ are bounded. Hence $\{\bar{y}(k)\}$ has at least one limit point. For any limit point $\bar{y}^*$ of the sequence $\{\bar{y}(k)\}$, there exists a subsequence $\{\bar{y}(k)\}_{\mathcal{K}}$ converges to $\bar{y}^*$, i.e., $\lim_{k \to \infty}\{\bar{y}(k)\}_{\mathcal{K}} = \bar{y}^*$. By Theorem 3,

$$H\bar{y}(k) \leq c + \frac{M}{k - k_0 + 1}, \forall k \in \mathcal{K}.$$

Since the subsequence $\{H\bar{y}(k)\}_{\mathcal{K}}$ converges as well by the continuity of the mapping from $y$ to $Hy$, we take the limits on the both sides of the above inequality, which yields

$$\lim_{k \to \infty, k \in \mathcal{K}} H\bar{y}(k) \leq \lim_{k \to \infty, k \in \mathcal{K}} (c + \frac{M}{k - k_0 + 1}) = c.$$

By the continuity of the mapping from $y$ to $Hy$, we have

$$H\bar{y}^* = H \lim_{k \to \infty, k \in \mathcal{K}} \bar{y}(k) = \lim_{k \to \infty, k \in \mathcal{K}} H\bar{y}(k) \leq c.$$

■

*Proof of Theorem 5:* Let $\bar{y}^*$ be a limit point of the sequence $\{\bar{y}(k)\}$. By Theorem 4, we have

$$H\bar{y}^* \leq c. \tag{19}$$

At any time slot $k$, $Ay(k) = x(k)$ by (10). By Theorem 2, for any $\epsilon > 0$, there exist a sequence of step size $\{\delta(k)\}$ and a sufficiently large $K_0$ such that, for any initial $\lambda(0) \geq 0$, for all $k \geq K_0$, $d(\lambda(k), \Lambda^*) < \epsilon$ and $||x(k) - x^*|| < \epsilon$. So for all $k \geq K_0$, $||Ay(k) - x^*|| < \epsilon$. It is easy to see that there exists a time $K_1 > K_0$ such that, for all $k \geq K_1$, $||A\bar{y}(k) - x^*|| < \epsilon$. Hence

$$||A\bar{y}^* - x^*|| < \epsilon. \tag{20}$$

From (19) and (20), we have $\bar{y}^* \in \mathcal{Y}^*(\epsilon)$. ■

## VIII. Appendix B: Proofs in Section IV

*Proof of Lemma 6:* Since the strong duality holds for both the master and the restricted problems, we have

$$\sum_{s \in S} U_s(x_s^*) = \theta(\lambda^*), \sum_{s \in S} U_s(\bar{x}_s^{(q)}) = \theta^{(q)}(\bar{\lambda}^{(q)}). \tag{21}$$

Since the $q^{th}$-RMP is more restricted than the MP, we have

$$\sum_{s \in S} U_s(x_s^*) \geq \sum_{s \in S} U_s(\bar{x}_s^{(q)}). \tag{22}$$

Combining (21) and (22), we get the following lower bound for the optimal objective value of the MP.

$$\sum_{s \in S} U_s(x_s^*) \geq \sum_{s \in S} U_s(\bar{x}_s^{(q)}) = \theta^{(q)}(\bar{\lambda}^{(q)}). \tag{23}$$

By the weak duality [24], for any $\lambda$ feasible to the dual problem of the MP, $\theta(\lambda)$ is an upper bound for the optimal objective value of the MP. In particular, consider $\bar{\lambda}^{(q)}$, which is optimal to the dual of the $q^{th}$-RMP and feasible to the dual of the MP. $\theta(\bar{\lambda}^{(q)})$ is an upper bound of $\sum_{s\in S} U_s(x_s^*)$, i.e.,

$$\theta(\bar{\lambda}^{(q)}) \geq \sum_{s\in S} U_s(x_s^*). \tag{24}$$

By the definitions of the dual functions,

$$\begin{aligned}
&\theta(\bar{\lambda}^{(q)}) - \theta^{(q)}(\bar{\lambda}^{(q)}) \\
=&\sum_{s\in S}\Big(\max_{x_s=\sum_{t\in T_s} y_t,\ m_s\leq x_s\leq M_s,\ y\geq 0}\{U_s(x_s) - \sum_{t\in T_s} y_t \sum_{e\in t} \bar{\lambda}_e^{(q)}\} \\
-&\max_{x_s=\sum_{t\in T_s^{(q)}} y_t,\ m_s\leq x_s\leq M_s,\ y\geq 0}\{U_s(x_s) - \sum_{t\in T_s^{(q)}} y_t \sum_{e\in t} \bar{\lambda}_e^{(q)}\}\Big) \\
=&\sum_{s\in S}(h_s(\gamma(s,\bar{\lambda}^{(q)})) - h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)}))).
\end{aligned}$$

In the last equality, we plug in the Lagrangian maximizers according to Lemma 1 part $(b)$. Hence, the gap between the upper and lower bounds for the optimal objective value of the MP is $\sum_{s\in S}(h_s(\gamma(s,\bar{\lambda}^{(q)})) - h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)})))$. We will show later in the proof of Theorem 9 that $h_s(w)$ is a non-increasing function. Thus $\gamma(s,\bar{\lambda}^{(q)}) \leq \gamma^{(q)}(s,\bar{\lambda}^{(q)}))$ implies $h_s(\gamma(s,\bar{\lambda}^{(q)})) - h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)})) \geq 0$ for any $s \in S$. Then the optimality gap is 0 if and only if $h_s(\gamma(s,\bar{\lambda}^{(q)})) = h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)}))$ for all source $s \in S$. ■

*Proof of Theorem 9:* Since the $q^{th}$-RMP is more restricted than the MP, we have $\theta^{(q)}(\bar{\lambda}^{(q)}) \leq \sum_{s\in S} U_s(x_s^*)$. Note that $\rho\bar{\lambda}^{(q)} \geq 0$ is a feasible dual variable vector. By the weak duality, we have $\sum_{s\in S} U_s(x_s^*) \leq \theta(\rho\bar{\lambda}^{(q)})$.

By the definition of the dual functions, we have

$$\begin{aligned}
\theta(\rho\bar{\lambda}^{(q)}) =&\sum_{s\in S}\Big(\max_{x_s=\sum_{t\in T_s} y_t,\ m_s\leq x_s\leq M_s,\ y\geq 0}\{U_s(x_s) \\
&- \sum_{t\in T_s} y_t \sum_{e\in t} \rho\bar{\lambda}_e^{(q)}\}\Big) + \sum_{e\in E} \rho\bar{\lambda}_e^{(q)} c_e,
\end{aligned}$$

and

$$\begin{aligned}
\theta^{(q)}(\bar{\lambda}^{(q)}) =&\sum_{s\in S}\Big(\max_{x_s=\sum_{t\in T_s^{(q)}} y_t,\ m_s\leq x_s\leq M_s,\ y\geq 0}\{U_s(x_s) \\
&- \sum_{t\in T_s^{(q)}} y_t \sum_{e\in t} \bar{\lambda}_e^{(q)}\}\Big) + \sum_{e\in E} \bar{\lambda}_e^{(q)} c_e.
\end{aligned}$$

One of the Lagrangian maximizers of $\theta(\cdot)$ at $\rho\bar{\lambda}^{(q)}$ is

$$x_s(\rho\bar{\lambda}^{(q)}) = [(U_s')^{-1}(\rho\gamma(s,\bar{\lambda}^{(q)}))]_{m_s}^{M_s}, \forall s \in S,$$

and

$$y_t(\rho\bar{\lambda}^{(q)}) = \begin{cases} x_s(\rho\bar{\lambda}^{(q)}) & \text{if } t = t(s,\rho\bar{\lambda}^{(q)}); \\ 0 & \text{otherwise,} \end{cases} \quad \forall t \in T.$$

To see that $(x_s(\rho\bar{\lambda}^{(q)}), y_t(\rho\bar{\lambda}^{(q)}))$ is a Lagrangian maximizer, we note that after the link dual vector is linearly scaled up by $\rho$, the global min-cost tree is not changed and the global minimum tree cost is $\rho\gamma(s,\bar{\lambda}^{(q)})$. By the similar arguments as in the proof of Theorem 1, the above selected vector is a Lagrangian maximizer.

For each source $s$, we define a function

$$g_s(w) = U_s((U_s')^{-1}(w)) - (U_s')^{-1}(w) \cdot w$$

for all $w > 0$. $g_s(w)$ is a non-increasing function for all $w > 0$. This monotonicity can be verified by checking $g_s'(w)$.

$$\begin{aligned}
g'(w) =&U_s'((U_s')^{-1}(w)) \cdot ((U_s')^{-1})'(w) \\
&-((U_s')^{-1})'(w) \cdot w - (U_s')^{-1}(w) \\
=&w \cdot ((U_s')^{-1})'(w) - ((U_s')^{-1})'(w) \cdot w - (U_s')^{-1}(w) \\
=&-(U_s')^{-1}(w) \leq 0.
\end{aligned}$$

Here $U_s(\cdot)$ is non-decreasing function and $U_s'(x) \geq 0$ for all $0 \leq m_s \leq x \leq M_s$. Hence $(U_s')^{-1}(w) \geq 0$ for all $w > 0$.

Recall that we define $h_s(w)$ for each source $s$ as

$$h_s(w) = U_s([(U_s')^{-1}(w)]_{m_s}^{M_s}) - [(U_s')^{-1}(w)]_{m_s}^{M_s} \cdot w$$

for all $w > 0$. $h_s(w)$ is also a non-increasing function for all $w > 0$, that is

$$h_s(w_1) \geq h_s(w_2) \text{ if } 0 < w_1 \leq w_2. \tag{25}$$

We first note that $U_s'(\cdot)$ is a decreasing function since $U_s(\cdot)$ is strictly concave. Hence $(U_s')^{-1}(\cdot)$ is also a decreasing function. To prove the monotonicity of the function $h_s(w)$, we need to discuss serval cases.

**Case** 1: $U_s'(M_s) \leq w_1 \leq w_2 \leq U_s'(m_s)$. From the monotonicity of $(U_s')^{-1}(\cdot)$, $[(U_s')^{-1}(w_1)]_{m_s}^{M_s} = (U_s')^{-1}(w_1)$, and $[(U_s')^{-1}(w_2)]_{m_s}^{M_s} = (U_s')^{-1}(w_2)$ in this case. From $0 < w_1 \leq w_2$ and the fact that $g_s(w)$ is a non-increasing function, we have $g_s(w_1) \geq g_s(w_2)$, which yields (25).

**Case** 2: $w_1 \leq U_s'(M_s) \leq w_2 \leq U_s'(m_s)$. As in case 1, we can show that

$$\begin{aligned}
&U_s([(U_s')^{-1}(w_2)]_{m_s}^{M_s}) - [(U_s')^{-1}(w_2)]_{m_s}^{M_s} \cdot w_2 \leq \\
&U_s([(U_s')^{-1}(U_s'(M_s))]_{m_s}^{M_s}) - [(U_s')^{-1}(U_s'(M_s))]_{m_s}^{M_s} \cdot U_s'(M_s).
\end{aligned}$$

Further

$$\begin{aligned}
&U_s([(U_s')^{-1}(w_1)]_{m_s}^{M_s}) - [(U_s')^{-1}(w_1)]_{m_s}^{M_s} \cdot w_1 \\
=&U_s(M_s) - M_s \cdot w_1 \\
\geq&U_s(M_s) - M_s \cdot U_s'(M_s) \\
=&U_s([(U_s')^{-1}(U_s'(M_s))]_{m_s}^{M_s}) - [(U_s')^{-1}(U_s'(M_s))]_{m_s}^{M_s} \cdot U_s'(M_s).
\end{aligned}$$

Hence (25) holds by combining the above two inequalities.

**Case** 3: $w_1 \leq w_2 \leq U_s'(M_s)$. In this case

$$U_s([(U_s')^{-1}(w_2)]_{m_s}^{M_s}) - [(U_s')^{-1}(w_2)]_{m_s}^{M_s} \cdot w_2 = U_s(M_s) - M_s \cdot w_2,$$

and

$$U_s([(U_s')^{-1}(w_1)]_{m_s}^{M_s}) - [(U_s')^{-1}(w_1)]_{m_s}^{M_s} \cdot w_1 = U_s(M_s) - M_s \cdot w_1.$$

So (25) holds trivially.

**Case** 4: $U_s'(M_s) \leq w_1 \leq U_s'(m_s) \leq w_2$. The proof is the same as that of the case 2.

**Case** 5: $U_s'(m_s) \leq w_1 \leq w_2$. (25) holds trivially as in case 3.

Without loss of generality, we can assume that $\gamma(s,\bar{\lambda}^{(q)}) > 0$. From (16), we have $0 < \gamma(s,\bar{\lambda}^{(q)}) \leq \gamma_\rho(s,\bar{\lambda}^{(q)}) \leq \rho\gamma(s,\bar{\lambda}^{(q)})$, which implies

$$h_s(\rho\gamma(s,\bar{\lambda}^{(q)})) \leq h_s(\gamma_\rho(s,\bar{\lambda}^{(q)})). \tag{26}$$

Now the dual function $\theta(\rho\bar{\lambda}^{(q)})$ can be written explicitly as

$$\begin{aligned}\theta(\rho\bar{\lambda}^{(q)}) =& \rho\sum_{e\in E}\bar{\lambda}_e^{(q)}c_e + \sum_{s\in S}h_s(\rho\gamma(s,\bar{\lambda}^{(q)}))\\ \leq& \rho\sum_{e\in E}\bar{\lambda}_e^{(q)}c_e + \sum_{s\in S}h_s(\gamma_\rho(s,\bar{\lambda}^{(q)}))\\ =& \rho\sum_{e\in E}\bar{\lambda}_e^{(q)}c_e + \sum_{s\in S}h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)}))\\ \leq& \rho\sum_{e\in E}\bar{\lambda}_e^{(q)}c_e + \rho\sum_{s\in S}h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)}))\\ =& \rho\theta^{(q)}(\bar{\lambda}^{(q)}).\end{aligned}$$

The first equality is to plug in the Lagrangian maximizer at $\rho\bar{\lambda}^{(q)}$, and to recognize $h_s(\cdot)$. The first inequality is from (26). The second equality holds because $\gamma_\rho(s,\bar{\lambda}^{(q)}) = \gamma^{(q)}(s,\bar{\lambda}^{(q)})$ by Theorem 8. The second inequality holds because, under the assumptions $A3$ and $A4$, $h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)})) \geq 0$ and $\rho \geq 1$. To see that $h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)})) \geq 0$, we note that $\gamma^{(q)}(s,\bar{\lambda}^{(q)}) < U_s'(m_s)$ by the assumption $A3$ (i.e., $\bar{x}_s^{(q)} > m_s$). From the non-increasing property of $h_s(w)$, we have $h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)})) \geq h_s(U_s'(m_s)) = U_s(m_s) - m_s \cdot U_s'(m_s) \geq 0$. The non-negativity of $h_s(U_s'(m_s))$ is by the assumption $A4$. The last equality holds by recognizing that $\theta^{(q)}(\bar{\lambda}^{(q)}) = \sum_{e\in E}\bar{\lambda}_e^{(q)}c_e + \sum_{s\in S}h_s(\gamma^{(q)}(s,\bar{\lambda}^{(q)}))$. ■

## References

[1] BitTorrent Website, http://www.bittorrent.com/.

[2] J. Lee and G. de Veciana, "On application-level load balancing in FastReplica," *Computer Communications*, vol. 30, no. 17, pp. 3218–3231, November 2007.

[3] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: high bandwidth data dissemination using an overlay mesh," in *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, October 2003.

[4] D. Kostić, R. Braud, C. Killian, E. Vandekieft, J. W. Anderson, A. C. Snoeren, and A. Vahdat, "Maintaining high bandwidth under dynamic network conditions," in *Proceedings of USENIX Annual Technical Conference*, 2005.

[5] B.-G. Chun, P. Wu, H. Weatherspoon, and J. Kubiatowicz, "ChunkCast: An anycast service for large content distribution," in *Proceedings of the Internaltional Workshop on Peer-to-Peer Systems (IPTPS)*, February 2006.

[6] K. Park and V. S. Pai, "Scale and performance in the CoBlitz large-file distribution service," in *Proceedings of the 3rd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, May 2006.

[7] X. Zheng, C. Cho, and Y. Xia, "Optimal peer-to-peer technique for massive content distribution," in *Proceedings of IEEE INFOCOM*, 2008.

[8] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," in *ACM-SIAM symposium on Discrete algorithms*, San Francisco, California, 1998, pp. 192–200.

[9] L. Zosin and S. Khuller, "On directed Steiner trees," in *ACM-SIAM symposium on Discrete algorithms*, San Francisco, California, 2002, pp. 59–63.

[10] M.-I. Hsieh, E. H.-K. Wu, and M.-F. Tsai, "FasterDSP: A faster approximation algorithm for directed Steiner tree problem," *Journal Of Information Science and Engineering*, vol. 22, pp. 1409–1425, 2006.

[11] S. Chen, O. Gunluk, and B. Yener, "The multicast packing problem," *IEEE/ACM Transaction on Networking*, vol. 8, no. 3, pp. 311– 318, June 2000.

[12] K. Jansen and H. Zhang, "An approximation algorithm for the multicast congestion problem via minimum steiner trees," in *In 3rd International Workshop on Approximation and Randomized Algorithms in Communication Networks ARANCE*, 2002, pp. 152–164.

[13] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *The Proceedings of IEEE International Symposium on Information Theory (ISIT)*, June 2004.

[14] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proceedings of IEEE INFOCOM*, 2007.

[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Mdard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proceedings of IEEE INFOCOM*, 2005, pp. 1607–1617.

[16] C. A. Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers and Operations Research*, 2005.

[17] R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in BitTorrent via biased neighbor selection," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'06)*, 2006.

[18] H. Zhang, G. Neglia, D. Towsley, and G. L. Presti, "On unstructured file sharing networks," in *Proceedings of INFOCOM*, May 2007.

[19] R. Kumar and K. Ross, "Peer-assisted file distribution: The minimum distribution time," in *IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB)*, 2006.

[20] P. Key, L. Massouliè, and D. Towsley, "Path selection and multipath congestion control," in *Proceedings of INFOCOM 2007*, May 2007.

[21] F. Paganini, "Congestion control with adaptive multipath routing based on optimization," in *The 40th Annual Conference on Information Sciences and Systems*, 2006.

[22] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *Automatic Control, IEEE Transactions on*, vol. 51, no. 5, pp. 766–781, May 2006.

[23] I. Lestas and G. Vinnicombe, "Combined control of routing and flow: a multipath routing approach," in *43rd IEEE Conference on Decision and Control*, 2004.

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

[25] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. Wiley-Interscience, 2006.

[26] X. Lin, N. B. Shroff, and R. Srikant, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE/ACM Transaction on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.

[27] J. Wang, L. Li, S. H. Low, and J. C. Doyle, "Can shortest-path routing and TCP maximize utility," in *Proceedings of INFOCOM*, 2003.

[28] F. K. Hwang, D. S. Richards, and P. Winter, "The Steiner tree problems," *Annals of Discrete Mathematics*, vol. 53, 1992.

[29] X. Zheng, F. Chen, Y. Xia, and Y. Fang, "A class of cross-layer optimization algorithms for performance and complexity trade-offs in wireless networks," http://www.cise.ufl.edu/~yx1/paper_by_area.html.

[30] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach," in *Proceedings of INFOCOM*, 2003.

[31] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transaction on Wireless Communications*, vol. 5, no. 2, pp. 435–445, Feb. 2006.

[32] S. Kompella, J. E. Wieselthier, and A. Ephremides, "A cross-layer approach to optimal wireless link scheduling with SINR constraints," in *MilCom 2007*, 2007.

[33] A. L. Stolyar, "Maximizing queueing network utility subject to stability: greedy primal-dual algorithm," *Queueing Systemes*, vol. 50, no. 4, pp. 401–457, 2005.