Wireless Multicasting via Iterative Optimization

Lihua Wan and Jie Luo
Electrical & Computer Engineering Department
Colorado State University, Fort Collins, CO 80523
Email: {carlawan, rockey}@engr.colostate.edu

***Abstract*— A class of wireless multicast utility optimization problems are considered. Assume network utility is only a function of link throughput and nodes' transmission power. Under a set of physical layer assumptions, the impact of physical and data-link layer configurations to the upper layers in a wireless network can be characterized using a configuration graph. Network layer utility optimization can consequently be carried out via iterations that optimize network layer algorithms over the configuration graph and revise physical, data-link layer configurations to improve the configuration graph. For a class of wireless multicast networks with optimal network coding, the number of point-to-multipoint links involved in network utility optimization is only polynomial, as opposed to exponential, in the number of nodes. This leads to a reduced complexity in exhaustive searching algorithms, such as optimal utility maximization. Consequently, simulations of these algorithms for moderate-sized networks become feasible.** [1]

I. Introduction

Unlike a wireline network, whose topology can be modeled by a graph, a wireless network can only be characterized by a bunch of nodes with channels between them being specified using statistical models. Extending the graphic model to wireless networks faces two key challenges. First, in principle, a wireless node can deliver information reliably to another node at a positive rate so long as the channel gain between them is not strictly zero. Achievable throughput over a wireless link strongly depends on the communication activities over other neighboring links. Without detailed physical and link layer specifications, using a fixed topology graph to model a wireless network is not informative. Second, due to the open nature of wireless medium, it is possible for a wireless node to communicate common information to multiple receivers simultaneously [1] using the same power and bandwidth of a point-to-point transmission. When such transmissions are modeled by point-to-multipoint links (termed hyperarc links [2]), the total number of feasible links in a wireless network can be exponential in the number of nodes. Consequently, network optimization requiring an exhaustive search over all links can be overly complex even for moderate-sized networks.

In this paper, we first consider the modeling of wireless networks in a class of constrained network utility maximization problems. Assume both network utility and constraints are only functions of link throughput and nodes' transmission power. Under a set of physical layer assumptions, we show that, the impact of physical and data-link layer configurations in a wireless network can be characterized by a configuration graph (defined in Section II, and firstly introduced in [4]), which is similar to a wireline network topology graph. If both the utility function and the constraint functions are convex in throughput and transmission power, optimal network utility can be obtained using iterative algorithms that optimize network layer operations over the configuration graph and incrementally revise physical, data-link layer configurations to improve the configuration graph.

Next, we consider a class of utility optimization problems for wireless multicast networks with optimal intra-session network coding. We show that the total number of links involved in the iterative optimization is only polynomial in the number of nodes[2]. With this result, complexity of exhaustive searching algorithms, such as optimal utility maximization, is reduced from double exponential to exponential in the number of nodes. The exponential complexity is due to the necessity of searching link activation combinations. Furthermore, we show that, by carefully exploiting fundamental properties of wireless communication and network coding, practical complexity of exhaustive searching algorithms can be made significantly lower than its theoretic value. Consequently, computer simulations of optimal multicast utility maximization becomes feasible for moderate-sized networks, and this is important in providing performance benchmarks for other low complexity suboptimal algorithms.

II. System Model

Consider a wireless network with node set V. We define a hyperarc link e_{iJ} from node i to a node set J, if i can deliver common information simultaneously and directly to all nodes in J. We say e_{iJ} achieves a throughput of μ_{iJ} if i communicates common information directly to all nodes in J at rate μ_{iJ} with a negligible error probability. We assume time is slotted. Each slot is long enough to support close link throughput approximations using the corresponding information capacity results[3].

Assume static channel conditions. We define a communication realization $C(t)$ as the simultaneous activation of a set of links, together with their physical layer configurations[4]. Each communication realization $C(t)$ determines a

[1]This work was supported by National Science Foundation under Grant CCF-0728826.

[2]But it is still exponential in the number of multicast destinations.

[3]This assumption can be relaxed by replacing the information theoretic capacity with rate estimation under none-zero error probability and finite codeword length constraints [5].

[4]Although not necessary, it is helpful to assume that the configuration should not involve any time-sharing operation within one time slot.

region of link throughput vectors achievable within time slot t, denoted by $\mathcal{C}(t)$. Because a communication realization $C(t)$ completely specifies the physical layer configuration, it determines the average transmission power of the nodes, denoted by a vector $\boldsymbol{p}(t) = \boldsymbol{p}(C(t))$, in time slot t. We define a transmission schedule $S = \{C(0), C(1), \dots, C(T-1)\}$ as the periodic extension of a communication realization sequence $C(0), C(1), \dots, C(T-1)$, with T being the period and $C(t)$ being the communication realization of time slots $kT + t$, $k = 0, 1, \dots, \infty$. Each transmission schedule S determines a region of achievable link throughput vectors, denoted by $\mathcal{S}$. The average transmission power of the nodes specified by transmission schedule S equals $\boldsymbol{p}(S) = \frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{p}(t) = \frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{p}(C(t))$. An illustration of links, communication realizations, and transmission schedule is given in Figure 1.

The physical and link layer model described above implies two key assumptions. First, we only distinguish successful and non-successful transmissions over each link. A successful transmission requires all receivers of the link should obtain the transmitted message reliably, which implies decoding and forward, as opposed to amplifying and forward, for information relay. Second, messages transmitted over different links are encoded and decoded independently at the physical layer. There is no joint encoding among multiple transmitters or joint decoding (e.g. successive interference cancellation) over multiple links at the physical layer.

Given a transmission schedule $S = \{C(0), \dots, C(T-1)\}$, we choose an achievable link throughput vector $\boldsymbol{\mu}(t) \in \mathcal{C}(t)$, for each $C(t)$. Construct a configuration graph $G(V, E, \boldsymbol{\mu}) = G(S)$ with E being the union of links of all communication realizations. We use rate vector $\boldsymbol{\mu} = \frac{1}{T}\sum_t \boldsymbol{\mu}(t)$ to specify a hyper-cubic region $\{\tilde{\boldsymbol{\mu}} | \tilde{\boldsymbol{\mu}} \le \boldsymbol{\mu}\}$ of link throughput vectors achievable on $G(V, E, \boldsymbol{\mu})$, i.e., $\tilde{\boldsymbol{\mu}} \in \mathcal{S}$ for all $\tilde{\boldsymbol{\mu}} \le \boldsymbol{\mu}$. We term $\boldsymbol{\mu}$ the configuration rate vector of $G(V, E, \boldsymbol{\mu})$. An illustration of the relation between a transmission schedule and a configuration graph is given in Figure 1. Note that the idea

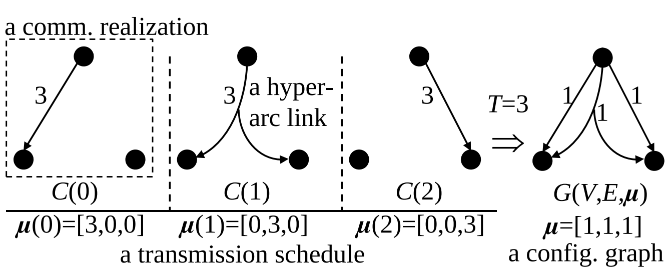

Fig. 1. An illustration of configuration graph construction.

of representing the impact of a transmission schedule using a configuration graph was originally proposed in [6], although the concept of "configuration graph" was formally introduced in [4]. Given a fixed transmission schedule S, the configuration rate vector $\boldsymbol{\mu}$ is similar to the link capacity vector in a wireline network topology graph in the sense that all link throughput vectors $\tilde{\boldsymbol{\mu}} \le \boldsymbol{\mu}$ are achievable under the same set of physical and link layer configurations.

Now consider a class of network utility maximization problems formulated as

$$\max_{S, \boldsymbol{\mu} \in \mathcal{S}} U(\boldsymbol{\mu}, \boldsymbol{p}), \qquad \text{s.t. } \boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p}) \le \boldsymbol{0}, \tag{1}$$

where, given a transmission schedule S, we assume both the utility $U(\boldsymbol{\mu}, \boldsymbol{p})$ and the constraints $\boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p})$ are functions of the link throughput $\boldsymbol{\mu} \in \mathcal{S}$ achievable under S, and the transmission power $\boldsymbol{p}$ determined by S. Because $\boldsymbol{\mu} \in \mathcal{S}$, (1) can be equivalently written as

$$\max_{S, \boldsymbol{\mu}, G(V,E,\boldsymbol{\mu}) = G(S)} U(\boldsymbol{\mu}, \boldsymbol{p}), \qquad \text{s.t. } \boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p}) \le \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{\mu}$ is now the configuration rate vector of the configuration graph $G(V, E, \boldsymbol{\mu})$ constructed based upon S.

Note that in a practical system, expressions of the $U(\boldsymbol{\mu}, \boldsymbol{p})$ and the $\boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p})$ functions may depend on network layer operations, as illustrated in the following example.

Example 1: Consider a multi-hop wireless network with a source node s multicasting common information reliably to a set of destination nodes D. Denote the multicast throughput by R_{sD}. Consider the following utility optimization problem.

$$\max_{S, R_{sD}(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{S}} U(R_{sD}, \boldsymbol{p}), \qquad \text{s.t. } \boldsymbol{p} - \boldsymbol{P} \le \boldsymbol{0}. \tag{3}$$

Given S, $R_{sD} = R_{sD}(\boldsymbol{\mu})$ can be written as a function of a link throughput vector $\boldsymbol{\mu} \in \mathcal{S}$. However, the exact expression of $R_{sD}(\boldsymbol{\mu})$ depends on the network layer protocol. If we assume optimal network coding, and let $\boldsymbol{\mu}$ be the configuration rate vector of a configuration graph $G(V, E, \boldsymbol{\mu}) = G(S)$, then R_{sD} equals the max-flow rate of the minimum $s - D$ cut of $G(V, E, \boldsymbol{\mu})$, which is the sum configuration rates of links crossing the minimum cut that separates s from at least one destination node in D [3][2].

III. Iterative Network Utility Optimization

Consider the network utility optimization problem given in (2). When the utility function $U(\boldsymbol{\mu}, \boldsymbol{p})$ is concave and the constraint functions $\boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p})$ are convex in $\boldsymbol{\mu}$ and $\boldsymbol{p}$ (e.g. Example 1 with $U(R_{sD}, \boldsymbol{p}) = R_{sD}$), (2) is a convex optimization problem. The unique optimal solution $(\boldsymbol{\mu}, \boldsymbol{p})^*$ can be obtained by solving

$$\min_{S, \boldsymbol{\mu}, G(V,E,\boldsymbol{\mu}) = G(S)} \max_{\boldsymbol{\lambda} \ge \boldsymbol{0}} L(\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{\lambda}),$$
$$L(\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{\lambda}) = -U(\boldsymbol{\mu}, \boldsymbol{p}) + \boldsymbol{\lambda}^T \boldsymbol{H}(\boldsymbol{\mu}, \boldsymbol{p}), \tag{4}$$

where $L(\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{\lambda})$ is the Lagrangian function and $\boldsymbol{\lambda} \ge \boldsymbol{0}$ is the vector of Lagrangian multipliers.

In a practical wireless network, it is often difficult to determine whether a $(\boldsymbol{\mu}, \boldsymbol{p})$ pair is supported by any transmission schedule. Consequently, (4) can only be solved via iterative and constructive updates of the transmission schedule. Next, we show that, given a fixed Lagrangian parameter $\boldsymbol{\lambda} \ge \boldsymbol{0}$, $L(\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{\lambda})$ in (4) can indeed be minimized *monotonically* via iterative and *incremental* updates of S.

Definition 1: Let $S_1 = \{C_1(0), C_1(1), \dots, C_1(T_1 - 1)\}$ and $S_2 = \{C_2(0), C_2(1), \dots, C_2(T_2 - 1)\}$ be two transmission schedules. We say S_2 is incrementally different from S_1 if there exists a nonnegative integer k such that $T_2 = kT_1 + 1$, and for all nonnegative integers $\tilde{k} < k$ and $t \leq T_1 - 1$, we have $C_2(\tilde{k}T_1 + t) = C_1(t)$.

Theorem 1: Let $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}, \boldsymbol{p}) = L(\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{\lambda})$ in (4) for a given $\boldsymbol{\lambda}$. Define $L^*_{\boldsymbol{\lambda}}$ as $L^*_{\boldsymbol{\lambda}} = \min_{S,\boldsymbol{\mu},G(V,E,\boldsymbol{\mu})=G(S)} L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}, \boldsymbol{p})$. Let S_1 be a transmission schedule whose transmission power vector is $\boldsymbol{p}_1$. Let $\boldsymbol{\mu}_1$ be the configuration rate vector of a configuration graph $G_1(V, E, \boldsymbol{\mu}_1)$ constructed based upon S_1. If $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1) > L^*_{\boldsymbol{\lambda}}$, then there exists another transmission schedule S_2 with transmission power vector $\boldsymbol{p}_2$ and a configuration rate vector $\boldsymbol{\mu}_2$ of $G_2(V, E, \boldsymbol{\mu}_2)$ constructed based upon S_2, such that,

1. S_2 is incrementally different from S_1.
2. $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_2, \boldsymbol{p}_2) < L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)$.

Proof: By assumption, $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}, \boldsymbol{p})$ is convex in $(\boldsymbol{\mu}, \boldsymbol{p})$. Since $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1) > L^*_{\boldsymbol{\lambda}}$, we can find a transmission schedule $\tilde{S} = \{\tilde{C}(0), \ldots, \tilde{C}(\tilde{T} - 1)\}$ with $\boldsymbol{p}(\tilde{S}) = \tilde{\boldsymbol{p}}$, and a rate vector $\tilde{\boldsymbol{\mu}} \in \tilde{\mathcal{S}}$, such that

$$\left(\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{\mu}}\right)^T \tilde{\boldsymbol{\mu}} + \left(\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{p}}\right)^T \tilde{\boldsymbol{p}} < 0, \quad (5)$$

where $\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{\mu}}$ and $\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{p}}$ are partial sub-derivatives of $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}, \boldsymbol{p})$ at $(\boldsymbol{\mu}_1, \boldsymbol{p}_1)$.

Since $\boldsymbol{p}(\tilde{S}) = \tilde{\boldsymbol{p}}$, $\tilde{\boldsymbol{\mu}} \in \tilde{\mathcal{S}}$, we can find a sequence of $(\tilde{\boldsymbol{\mu}}(t), \tilde{\boldsymbol{p}}(t))$ pairs, $t \in \{0, \ldots, \tilde{T} - 1\}$, to satisfy $\tilde{\boldsymbol{p}}(t) = \boldsymbol{p}(\tilde{C}(t))$, $\tilde{\boldsymbol{\mu}}(t) \in \tilde{\mathcal{C}}(t)$, $\forall t$, and $\tilde{\boldsymbol{\mu}} = \frac{1}{\tilde{T}} \sum_{t=0}^{\tilde{T}-1} \tilde{\boldsymbol{\mu}}(t)$, $\tilde{\boldsymbol{p}} = \frac{1}{\tilde{T}} \sum_{t=0}^{\tilde{T}-1} \tilde{\boldsymbol{p}}(t)$. Hence there exists a $j \in \{0, \ldots, \tilde{T} - 1\}$, such that the $(\tilde{\boldsymbol{\mu}}(j), \tilde{\boldsymbol{p}}(j))$ pair corresponding to communication realization $\tilde{C}(j)$ satisfies

$$\left(\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{\mu}}\right)^T \tilde{\boldsymbol{\mu}}(j) + \left(\frac{\partial L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)}{\partial \boldsymbol{p}}\right)^T \tilde{\boldsymbol{p}}(j) < 0. \quad (6)$$

Consequently, we can find a positive integer k to satisfy

$$L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1) > L_{\boldsymbol{\lambda}}\left(\frac{kT_1\boldsymbol{\mu}_1 + \tilde{\boldsymbol{\mu}}(j)}{kT_1 + 1}, \frac{kT_1\boldsymbol{p}_1 + \tilde{\boldsymbol{p}}(j)}{kT_1 + 1}\right). \quad (7)$$

Now construct a transmission schedule $S_2 = \{C_2(0), C_2(1), \ldots, C_2(T_2 - 1)\}$ with $T_2 = kT_1 + 1$. For all nonnegative integers $\tilde{k} < k$ and $t \leq T_1 - 1$, we choose $C_2(\tilde{k}T_1 + t) = C_1(t)$, where $C_1(t)$ is the tth communication realization of S_1. We also let $C_2(T_2 - 1) = \tilde{C}(j)$. With this construction, S_2 is only incrementally different from S_1, and $L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_2, \boldsymbol{p}_2) < L_{\boldsymbol{\lambda}}(\boldsymbol{\mu}_1, \boldsymbol{p}_1)$ according to (7). ∎

Theorem 1 implies that (4) can be solved via iterative updates of $\boldsymbol{\lambda}$ and $(\boldsymbol{\mu}, \boldsymbol{p})$ pair, where each update of $(\boldsymbol{\mu}, \boldsymbol{p})$ pair only involves an incremental update, as opposed to a direct construction, of the transmission schedule. This extended the similar iterative optimization framework proposed in [4].

IV. Constrain The Number of Hyperarc Links

According to the proof of Theorem 1, the key task of the incremental transmission schedule update is to find a single communication realization $\tilde{C}(j)$ with $\tilde{\boldsymbol{p}}(j) = \boldsymbol{p}(\tilde{C}(j))$ and $\tilde{\boldsymbol{\mu}}(j) \in \tilde{\mathcal{C}}(j)$, such that updating $(\boldsymbol{\mu}, \boldsymbol{p})$ in the direction of $(\tilde{\boldsymbol{\mu}}(j), \tilde{\boldsymbol{p}}(j))$ improves the Lagrangian utility. The complexity of an exhaustive search of the communication realizations is exponential in the number of feasible links, and is therefore double exponential in the number of nodes since, in principle, the number of feasible links is exponential in the number of nodes. In this section, we show that the link throughput region determined by an arbitrary communication realization possesses a *hyper-coordinate-convexity* property. Due to this property, for the class of multicast utility optimization problems considered in Example 1, optimal utility can be achieved by searching a number of links that is only polynomial in the number of nodes. This consequently reduces the complexity of exhaustive searching algorithms from double exponential to exponential in the number of nodes.

Definition 2: Let $\boldsymbol{\nu}$ be a vector whose each element ν_{iJ} corresponds to a feasible link e_{iJ}. We say $\boldsymbol{\nu}$ is a *throughput degrading vector* if there exists a node i, two node subsets J, $\tilde{J} \subset J$, and a constant $\delta \geq 0$, such that $\nu_{iJ} = -\delta$, $\nu_{i\tilde{J}} = \delta$, and all other elements of $\boldsymbol{\nu}$ equal 0.

Let $\boldsymbol{\mu}_d$, $\boldsymbol{\mu}$ be two link throughput vectors. We say $\boldsymbol{\mu}_d$ is *a degraded version* of $\boldsymbol{\mu}$ if $\boldsymbol{\mu}_d - \boldsymbol{\mu}$ equals the summation of some throughput degrading vectors.

Definition 3: A link throughput vector region Γ is *hyper-coordinate convex* if for all throughput vectors $\tilde{\boldsymbol{\mu}}$, we have $\tilde{\boldsymbol{\mu}} \in \Gamma$ so long as there exist $\boldsymbol{\mu}$, $\boldsymbol{\mu}_d$, with $\boldsymbol{\mu} \in \Gamma$, $\boldsymbol{\mu}_d$ being a degraded version of $\boldsymbol{\mu}$, and $\tilde{\boldsymbol{\mu}} \leq \boldsymbol{\mu}_d$.

Lemma 1: Link throughput vector region $\mathcal{C}$ determined by any communication realization C is hyper-coordinate convex.

Proof: Multicasting common information from node i to a node set J can be achieved using any hyperarc link $e_{i\tilde{J}}$ so long as $\tilde{J} \supseteq J$. ∎

Theorem 2: Consider the class of multicast networks given in Example 1. Assume optimal network coding. Assume R_{sD} is achievable under a transmission schedule S with power $\boldsymbol{p}$. Then we can construct a configuration graph $G(V, E, \boldsymbol{\mu}) = G(S)$ based upon S with $\boldsymbol{\mu} \in \mathcal{S}$, such that R_{sD} is no larger than the max-flow of the minimum $s - D$ cut of $G(V, E, \boldsymbol{\mu})$, and $\boldsymbol{\mu}$ satisfies $\mu_{iJ} = 0$, $\forall i, J, |J| > |D|$.

Proof: By assumption, we can construct a configuration graph $G(V, E, \tilde{\boldsymbol{\mu}})$ based upon S with $\tilde{\boldsymbol{\mu}} \in \mathcal{S}$, such that R_{sD} is no larger than the max-flow of the minimum $s - D$ cut of $G(V, E, \tilde{\boldsymbol{\mu}})$. According to [2, Thoerem 2], we can find a set of virtual link throughput $\tilde{x}^{(d)}_{iJj} \geq 0$, each being defined for an arm e_{iJj} $(j \in J)$ of a link e_{iJ} corresponding to one destination d, such that

$$\tilde{\mu}_{iJ} \geq \sum_{j \in J} \tilde{x}^{(d)}_{iJj}, \quad \forall (i, J), e_{iJ} \in E, d \in D,$$
$$\sum_{\{J | (i,J), e_{iJ} \in E\}} \sum_{j \in J} \tilde{x}^{(d)}_{iJj} - \sum_{\{j | (j,I), e_{jI} \in E, i \in I\}} \tilde{x}^{(d)}_{jIi} = \sigma^{(d)}_i,$$
$$\forall i \in V, t \in T, \quad (8)$$

where $\sigma^{(d)}_s = R_{sD}$, $\sigma^{(d)}_d = -R_{sD}$, and $\sigma^{(d)}_{i \neq s,d} = 0$.

Next, we will show that, by applying a recursive throughput degrading algorithm, $\tilde{\boldsymbol{\mu}}$ can be degraded to $\boldsymbol{\mu}$ that satisfies

1. $\mu_{iJ} = 0$, $\forall i, J, |J| > |D|$.

2. There exists a set of virtual link throughput $x_{iJj}^{(d)} \geq 0$ such that the following inequalities hold.

$$\mu_{iJ} \geq \sum_{j\in J} x_{iJj}^{(d)}, \quad \forall (i,J), e_{iJ} \in E, d \in D,$$
$$\sum_{\{J|(i,J),e_{iJ}\in E\}} \sum_{j\in J} x_{iJj}^{(d)} - \sum_{\{j|(j,I),e_{jI}\in E, i\in I\}} x_{jIi}^{(d)} = \sigma_i^{(d)},$$
$$\forall i \in V, d \in D. \tag{9}$$

The throughput degrading algorithm is described below.

Initialization: Let $\boldsymbol{\mu} = \tilde{\boldsymbol{\mu}}$ and $x_{iJj}^{(d)} = \tilde{x}_{iJj}^{(d)}$, $\forall i, J, j, d$.

Step 1: Find a link e_{iJ} with $|J| > |D|$ and $\mu_{iJ} > 0$. The algorithm stops if such a link does not exist.

Step 2: If for all $d \in D$ and $j \in J$, $x_{iJj}^{(d)} = 0$, we choose an arbitrary $j \in J$. Define a throughput degrading vector $\boldsymbol{\nu}$ by $\nu_{iJ} = -\mu_{iJ}$, $\nu_{ij} = \mu_{iJ}$, and set all other elements of $\boldsymbol{\nu}$ at zero. We degrade $\boldsymbol{\mu}$ by $\boldsymbol{\mu} = \boldsymbol{\mu} + \boldsymbol{\nu}$. Go to Step 1.

Step 3: Assume we can find $d \in D$ and $j \in J$, such that $x_{iJj}^{(d)} > 0$. Define a positive constant δ by $\delta = \min_{j\in J, d\in D, x_{iJj}^{(d)}>0} x_{iJj}^{(d)}$. For every $d \in D$, we define a node index j_d as follows. If $x_{iJj}^{(d)} = 0$ for all $j \in J$, we let $j_d =$ NULL. Otherwise, $j_d = \text{argmin}_{j\in J, x_{iJj}^{(d)}>0} x_{iJj}^{(d)}$.

Define node subset $\tilde{J}$ as the collection of the j_d indices, $\tilde{J} = \{j_d | d \in D, j_d \neq \text{NULL}\}$. Define a throughput degrading vector $\boldsymbol{\nu}$ by $\nu_{iJ} = -\delta$, $\nu_{i\tilde{J}} = \delta$, and set all other elements of $\boldsymbol{\nu}$ at zero. We degrade $\boldsymbol{\mu}$ by $\boldsymbol{\mu} = \boldsymbol{\mu} + \boldsymbol{\nu}$. For each $d \in D$, we also revise $x_{iJj}^{(d)}$ and $x_{i\tilde{J}j}^{(d)}$ as follows. If $j_d \neq$ NULL, we let $x_{i\tilde{J}j_d}^{(d)} = x_{i\tilde{J}j_d}^{(d)} + \delta$ and $x_{iJj_d}^{(d)} = x_{iJj_d}^{(d)} - \delta$.

Go to Step 1.

The above algorithm reduces μ_{iJ} monotonically for all links e_{iJ} with $|J| > |D|$. The degraded throughput is allocated to link $e_{i\tilde{J}}$ in Steps 2 and 3 with $|\tilde{J}| \leq |D|$. Therefore, when the algorithm stops, we must have $\mu_{iJ} = 0$ for all $|J| > |D|$, and $\boldsymbol{\mu}$ is a degraded version of $\tilde{\boldsymbol{\mu}}$. Because (9) is satisfied in every step, it remains satisfied when the algorithm stops. The validity of (9) implies that R_{sD} is no larger than the max-flow of the minimum $s - D$ cut of configuration graph $G(V, E, \boldsymbol{\mu})$. In other words, R_{sD} is achievable on $G(V, E, \boldsymbol{\mu})$ with optimal network coding. ■

Theorem 2 implies that, for the multicast utility optimization considered in Example 1, in terms of achieving optimal utility, one only needs to consider links e_{iJ} with $|J| \leq |D|$. In other words, (3) can be *equivalently* written as

$$\max_{S, R_{sD}(\boldsymbol{\mu}\in S), \mu_{iJ}=0\ \forall e_{iJ} \text{with } |J|>|D|} U(R_{sD}, \boldsymbol{p}), \quad \text{s.t. } \boldsymbol{p} - \boldsymbol{P} \leq \boldsymbol{0}. \tag{10}$$

The maximum number of links involved in solving (10) optimally is only polynomial in the number of nodes $|V|$ (but is still exponential in $|D|$). It is straightforward to extend this conclusion to utility optimization problems with a more general set of constraints. It is also straightforward to extend the conclusion to networks with multiple multicast sessions and optimal *intra-session* network coding.

V. Further Complexity Reduction for Algorithms Involving Search of Link Combinations

Consider the class of multicast utility optimization problems in Example 1. Assume the network contains N nodes with one multicast session from 1 source to D destinations. The number of feasible links equals $N \sum_{i=1}^{D} \binom{N-1}{i} \propto O(N^{D+1})$. This number equals 50 for a network with $N = 5$ and $D = 2$, which means the complexity of an exhaustive search of link activation combinations equals 2^{50}. In this section, we show that by exploiting fundamental properties of wireless communication and network coding, the practical complexity of link combination search can be significantly reduced. With the reduced complexity, simulations of optimal utility maximization in moderate-sized networks with around 25 nodes and a multicast session from 1 source to 4 destinations become feasible.

A well known property of wireless communication is that a wireless node cannot simultaneously transmit and receive information in the same channel. Consequently, link activations involving the transmission-reception conflict at any node should not be considered in the communication realization construction. With the exploitation of this property, the number of link activation combinations in a network with $N = 5$, $D = 2$, is reduced to 5117 (obtained via enumeration). However, this number equals 14680149 for a network with $N = 7$, $D = 2$. Clearly, exploiting the wireless communication property alone is not enough to enable the simulation of optimal utility maximization for a reasonable-sized network.

A key property of optimal network coding is that, given a configuration graph, throughput of a multicast session is bottlenecked by the maximum flow of the minimum $s - D$ cut. Therefore, when searching for a communication realization to incrementally update the transmission schedule, the communication realization should not involve links that do not contribute to any of the minimum $s - D$ cuts. To introduce the principle of exploiting this property for complexity reduction, we will first present the detailed iterative utility optimization algorithm in the following.

Given a configuration graph $G(V, E, \boldsymbol{\mu})$ constructed base upon transmission schedule S. For any destination $d \in D$, we define $\gamma_d(G)$ as a cut that separates source s from d. Without causing any confusion, the cut value is also denoted by $\gamma_d(G)$. According to the network coding property, utility optimization problem (3) can be equivalently written as

$$\max_{S, \boldsymbol{\mu}, G(V,E,\boldsymbol{\mu})=G(S)} U\left(\min_{d\in D} \min_{\gamma_d} \gamma_d(G), \boldsymbol{p}\right), \quad \text{s.t. } \boldsymbol{p} - \boldsymbol{P} \leq \boldsymbol{0}. \tag{11}$$

This problem can be further transformed to

$$\min_{\lambda_d \geq 0, \sum_{d\in D} \lambda_d = 1} \ \min_{\lambda_{\gamma_d} \geq 0, \sum_{\gamma_d} \lambda_{\gamma_d} = 1} \ \max_{S, \boldsymbol{\mu}, G(V,E,\boldsymbol{\mu})=G(S)}$$

$$U\left(\sum_{d\in D}\lambda_d\sum_{\gamma_d}\lambda_{\gamma_d}\gamma_d(G),\boldsymbol{p}\right),\ \text{s.t. } \boldsymbol{p}-\boldsymbol{P}\le\boldsymbol{0}, \tag{12}$$

where we have introduced several sets of auxiliary variables, $\{\lambda_d, \forall d \in D\}$, and $\{\lambda_{\gamma_d}, \forall\gamma_d\}$ for each $d \in D$.

Consequently, we can obtain the optimal solution of the problem using the following iterative algorithm.

Initialization: Construct an arbitrary transmission schedule and the corresponding configuration graph. Initialize λ_d and λ_{γ_d} for all γ_d and $d \in D$.

Step 1: Update λ_d as

$$\lambda_d=\lambda_d-\delta_1\frac{\partial U(R_{sD},\boldsymbol{p})}{\partial R_{sD}}\sum_{\gamma_d}\lambda_{\gamma_d}\gamma_d(G),\quad \forall d\in D, \tag{13}$$

where $\delta_1 > 0$ is a small step size parameter. Then normalize λ_d to satisfy $\lambda_d \ge 0$ for all $d \in D$ and $\sum_{d\in D}\lambda_d = 1$.

Step 2: For each $d \in D$, update λ_{γ_d} as

$$\lambda_{\gamma_d}=\lambda_{\gamma_d}-\delta_2\frac{\partial U(R_{sD},\boldsymbol{p})}{\partial R_{sD}}\lambda_d\gamma_d(G),\quad \forall d\in D, \tag{14}$$

where $\delta_2 > 0$ is a small step size parameter. Then normalize λ_{γ_d} to satisfy $\lambda_{\gamma_d} \ge 0$ and $\sum_{\gamma_d}\lambda_{\gamma_d} = 1$ for each $d \in D$.

Step 3: Construct a communication realization C with throughput region $\mathcal{C}$ and power $\boldsymbol{p}(C) < \boldsymbol{P}$. Let $\gamma_d(\mathcal{C})$ be the sum throughput of links in C that cross cut γ_d. Communication realization C should be constructed to maximize

$$\frac{\partial U(R_{sD},\boldsymbol{p})}{\partial R_{sD}}\sum_{d\in D}\lambda_d\sum_{\gamma_d}\lambda_{\gamma_d}\gamma_d(\mathcal{C})+\left[\frac{\partial U(R_{sD},\boldsymbol{p})}{\partial \boldsymbol{p}}\right]^T\boldsymbol{p}(C). \tag{15}$$

Carry out an incremental update of the transmission schedule S using communication realization C.

Go to Step 1 till convergence.

We say a cut γ_d is "effective" if $\lambda_{\gamma_d} \ne 0$. The key idea of complexity reduction is to maintain a small list of effective cuts in the iterative algorithm, such that, when constructing the communication realization C in Step 3, hyperarc link e_{iJ} with at least one arm e_{iJj} not crossing any effective cut should not be considered.

Assume for each $d \in D$, the system maintains a list of effective $s-d$ cuts, denoted by Γ_d. The list is initially empty. Let $\epsilon > 0$ be a small threshold parameter (e.g. $\epsilon = 0.3$). We add an extra step, Step 1.5, into the iterative algorithm.

Step 1.5: For each $d \in D$, find a minimum $s-d$ cut γ_d^* in the configuration graph $G(V, E, \boldsymbol{\mu})$ and add γ_d^* into the cut list Γ_d (with a small λ_{γ_d} value). For all $\gamma_d \in \Gamma_d$, if $\gamma_d(G) \ge (1+\epsilon)\gamma_d^*(G)$, remove the cut γ_d from the list. After that, normalize λ_{γ_d} to satisfy $\lambda_{\gamma_d} \ge 0$ and $\sum_{\gamma_d}\lambda_{\gamma_d} = 1$ for each $d \in D$.

Note that Step 1.5 corresponds to an aggressive update of the λ_{γ_d} variables. Due to page limitations, discussions on the optimality of the revised iterative algorithm is skipped.

VI. Simulation Results

Consider a grid network with 16 nodes. Locations of the nodes are set at (xr, yr), where $x, y \in \{0, 1, 2, 3\}$ and $r = 2$ is a distance parameter. Assume source node $s = (0, 0)$ wants to multicast common information to three destinations $d_1 = (0, 3r)$, $d_2 = (3r, 3r)$, $d_3 = (3r, 0)$. We assume in each time slot, a node can at most transmit over one link with a *fixed* transmission power of $P = 100$. This assumption is made to avoid the challenge of transmission power optimization in (15). The channel gain between two nodes A and B at distance r_{AB} is set at $h_{AB} = 1/r_{AB}^3$. Assume additive Gaussian noise with zero mean and unit variance. We set $\epsilon = 0.3$ and use an adaptive step size adjustment algorithm whose detailed specification is skipped. The solid curve in Figure 2 shows the achieved multicast throughput obtained by the revised iterative algorithm. With complexity reduction, the maximum number of link activation combinations searched by the algorithm in one iteration is 3449. For performance comparison, the dashed curve shows the achieved throughput if only point-to-point transmission is allowed.

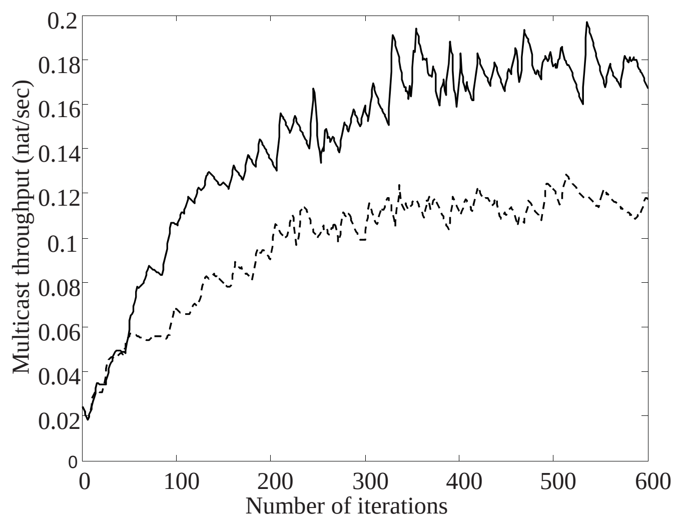

Fig. 2. Iterative multicast utility maximization for a network with 16 nodes and one multicast session with 3 destinations. Solid curve: achieved multicast throughput with hyperarc links. Dashed curve: achieved multicast throughput with unicast links.

References

[1] J. Wieselthier, G. Nguyen, and A. Ephremides, *Energy-efficient Broadcast and Multicast Trees in Wireless Networks,* Mobile Networks and Applications, Vol. 7, pp. 481-492, 2002.

[2] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, *Minimum-cost Multicast over Coded Packet Networks,* IEEE Trans. Inform. Theory, Vol. 52, pp. 2608-2623, Jun. 2006.

[3] R. Ahlswede, N. Cai, S. Li, and R. Yeung, *Network Information Flow,* IEEE Trans. Inform. Theory, Vol. 46, pp. 1204-1216, Jul. 2000.

[4] L. Wan, J. Luo, and A. Ephremides, *An Iterative Framework for Optimizing Multicast Throughput in Wireless Networks,* IEEE ISIT, Toronto, Canada, July 2008.

[5] Y. Polyanskiy, V. Poor, S. Verdú, *Channel Coding Rate in The Finite Blocklength Regime*, submitted to IEEE Trans. Inform. Theory.

[6] Y. Sagduyu and A. Ephremides, *On Joint MAC and Network Coding in Wireless Ad Hoc Networks,* IEEE Trans. Inform. Theory, Vol. 53, pp. 3697-3713, Oct. 2007.