Optimal Slotted Random Access in Coded Wireless Packet Networks

Maximilian Riemensberger*, Michael Heindlmaier†, Andreas Dotzler*, Danail Traskov†, and Wolfgang Utschick*

*Associate Institute for Signal Processing, Technische Universität München, 80290 Munich, Germany. {riemensberger,dotzler,utschick}@tum.de

†Institute for Communications Engineering, Technische Universität München, 80290 Munich, Germany. {michael.heindlmaier,danail.traskov}@tum.de

***Abstract*—We consider the problem of jointly optimizing random access and subgraph selection in coded wireless packet networks. As opposed to the corresponding scheduling approach, the problem cannot be formulated as a convex optimization problem and is thus difficult to solve. We propose a special adaptation of the branch and bound method which allows for computing the optimal strategy for small and medium networks. The key tool for efficiency is a random access specific convex relaxation bound. We compare these results to both optimal and suboptimal joint medium access and subgraph selection techniques, to conclude that there is room for further improvement of suboptimal yet practical random access schemes.**

I. Introduction

Optimized medium access is a decisive factor for the efficient operation of wireless networks. When resources are scarce and have to be shared, transmissions have to be controlled in order to avoid collisions. In principle there are two ways to overcome this problem: In a first approach a scheduler ensures orthogonality between transmissions. While this method can completely avoid collisions, it has to be coordinated by either a central entity that knows the global network state or by a distributed protocol. Therefore, in many practical systems random access mechanisms similar to slotted ALOHA are preferred. This approach is easier to implement in a distributed fashion as each node can decide independently if it should transmit or not, with the penalty of a nonzero probability of collisions.

In random access protocols, each transmitting node tries to adjust its attempt probability to optimize a criterion, such as maximum end-to-end throughput of a communication session, which is used in this work. Finding the optimal attempt probabilities for this problem is not trivial: Low probabilities cannot provide high throughput while too high probabilities will cause many collisions.

In the network layer, complementary to medium access, network coding can achieve higher throughput than routing. For the special case of a single multicast session, practical coding schemes are available and implementable in a distributed fashion [1]. Recent work [2], [3] showed that significant gains can be achieved by *jointly optimizing* medium access and network coding flow assignment. The joint subgraph selection and scheduling problem can be cast into a linear or convex optimization problem [3], [4]. However, the joint subgraph selection and random access problem, already considered in [5], turns out to be considerably more difficult to characterize. Unfortunately, it is a nonconvex problem, which, to the best of our knowledge, cannot be transformed into an equivalent convex problem, e.g. variable transformations exploiting log-convexity [6], [7] do not apply in this case, or solved by any other polynomial time algorithm.

We propose an algorithm that computes the globally optimal solution for joint random access and subgraph selection. As the problem is nonconvex, we use an appropriately adapted branch and bound approach for nonlinear programming [8]. Key tool for this algorithm is the monotonic structure of the joint optimization problem and the relation of the random access rate region and parts thereof to scheduling. The latter is exploited by explicitly parameterizing the convex hull of the rate regions and its subregions as a convex hull of finitely many points. At least for small and medium networks, we can compute the optimal solution and use it as a benchmark for the performance of random access in coded packet networks. We quantify the loss of random access w.r.t. scheduling [3] and show the potential gain for future random access schemes over the state of the art random access scheme for coded packet networks [5].

The key contributions of this paper are

- an algorithm that computes the optimal random access policy and subgraph selection for coded packet networks,
- an extensive comparison of optimal and suboptimal random access and scheduling policies,
- and an explicit parameterization of the convex hull of the random access rate region and its relation to scheduling.

The remainder of this paper is organized as follows: We formulate the network and interference model in Sec. II. A formal description of the achievable throughput region is given in Sec. III. Sec. IV presents the algorithm solving the joint random access and network coding problem, before we show simulation results of both optimal and suboptimal techniques in Sec. V. Sec. VI concludes the paper.

II. Network and Interference Model

We consider a wireless packet network with a set of nodes $\mathcal{N}$ corresponding to wireless devices and a neighborhood relation $N\colon \mathcal{N} \to 2^{\mathcal{N}}$ associating with each node i a set of nodes $N(i) \subset \mathcal{N}\backslash\{i\}$ that are in transmission range of i, excluding i. That is, only nodes $N(i)$ may receive packets or experience interference from i.[1] We assume time slotted communication with fixed length packets and backlogged queues at all nodes. We consider half-duplex transmission and secondary interference, i.e., in any time slot each node either transmits a packet or stays idle; an idle node successfully receives a packet if and only if it is within communication range of *exactly* one transmitting node.

We construct a directed hypergraph $(\mathcal{N}, \mathcal{A})$ such that the hyperarc set $\mathcal{A}$ consists of all pairs (i, J) satisfying $i \in \mathcal{N}$ and $J \subset N(i)$. At any given time slot, if nodes $I \subset \mathcal{N}$ transmit and $\mathcal{N}\backslash I$ stay idle, each transmitter $i \in I$ injects one packet into the hyperarc $(i, J) \in \mathcal{A}$ where J is [2]

$$J = \{j \in N(i) : j \notin I \cup N(I\backslash\{i\})\}. \tag{1}$$

That is, J contains all neighbors j of i that are idle, $j \notin I$, and do not have an interferer in communication range, $j \notin N(I\backslash\{i\})$. $J = \emptyset$ (empty set) implies that the packet is lost since no neighbor of i can receive. Obviously, the active hyperarc (i, J) depends only on the decision of the neighbors of nodes in $N(i)$, that is the two-hop neighborhood of i. For ease of notation we define a characteristic function $c_{iJ}\colon 2^{\mathcal{N}} \to \{0, 1\}$ for each hyperarc (i, J) such that $c_{iJ}(I) = 1$ if I injects a packet into (i, J) and $c_{iJ}(I) = 0$ otherwise.

We model packet injection and reception over time by counting processes that have finite time averages. Let y_i denote the rate at which node i injects packets into the network and z_{iJ} the rate at which node i injects packets into hyperarc (i, J) with $y_i = \sum_{J \subset N(i)} z_{iJ}$. For ease of notation we define vectors $\boldsymbol{y} = (y_i)_{i \in \mathcal{N}}$ and $\boldsymbol{z} = (z_{iJ})_{(i,J) \in \mathcal{A}}$. Packet injection rates are bounded by $0 \leq y_i \leq 1$, where $y_i = 1$ means i transmits a packet in every time slot and $y_i = 0$ means i is always idle. Packets injected into any hyperarc (i, J) are subject to erasures,[3] i.e., with some probability a subset of nodes $K \subset J$ successfully receives a packet whereas nodes in $J\backslash K$ cannot recover it. Let z_{iJK} be the rate at which packets injected into (i, J) are received by all nodes in $K \subset J$, but not received by any node in $J\backslash K$, K possibly being the empty set, with $z_{iJ} = \sum_{K \subset J} z_{iJK}$. The distribution of erasures is implicitly modeled by the probability distribution $p_{iJK} = \frac{z_{iJK}}{z_{iJ}}$, namely, the probability that a packet injected into (i, J) is received exactly by all nodes $K \subset J$. We further introduce $b_{iJK} = \sum_{L \subset J : L \cap K \neq \emptyset} p_{iJL}$, i.e., the probability that a packet injected into (i, J) is received by at least one node in K. Based on these definitions many probabilistic erasure models can be considered, including statistical dependencies.

[1]The network model would be easily extended by considering separate neighborhood relations for transmission, reception, and interference. However, we use the simplified model to keep notation as clear as possible.

[2]We define $N(K) = \bigcup_{j \in K} N(j)$ for any $K \subset \mathcal{N}$.

[3]Erasures model impairments of wireless transmission other than interference and half-duplex devices, e.g. fading, and affect only idle nodes that are free of interference.

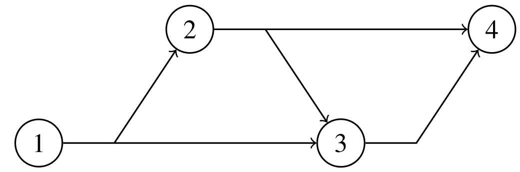

Figure 1. Example network with 4 nodes

Table I
INJECTION INTO HYPERARCS FOR EXAMPLE NETWORK

Transmitter set I	Injected Hyperarcs (i, J)
$\{1\}$	$(1, \{2, 3\})$
$\{2\}$	$(2, \{3, 4\})$
$\{3\}$	$(3, \{4\})$
$\{1,2\}$	$(1, \emptyset), (2, \{4\})$
$\{1,3\}$	$(1, \{2\}), (3, \{4\})$
$\{2,3\}$	$(2, \emptyset), (3, \emptyset)$
$\{1,2,3\}$	$(1, \emptyset), (2, \emptyset), (3, \emptyset)$

The injection rates $\boldsymbol{y}$ and $\boldsymbol{z}$ are governed by the mechanism which the nodes use to access the wireless medium. For slotted packet networks the most relevant mechanisms are scheduling and random access, which are described by their respective feasible hyperarc injection rate regions $\mathcal{Z}_{\text{s}}$ and $\mathcal{Z}_{\text{ra}}$.

In scheduled networks for each time slot all nodes agree on a policy, namely, a subset of hyperarcs where packets are injected, either using a central entity or a distributed coordination scheme. Optimal scheduling for coded packet networks has been studied in [3] using a conflict graph model to derive the feasible injection rate region $\mathcal{Z}_{\text{s}}$.

For random access in each time slot each node decides independently whether it transmits or stays idle. The average injection rate y_i equals the probability of transmission. For each hyperarc (i, J) the hyperarc injection rate z_{iJ} is a function of all nodes' packet injection rates $\boldsymbol{y}$ given by

$$z_{iJ}(\boldsymbol{y}) = \sum_{I \subset \mathcal{N}} c_{iJ}(I) \prod_{j \in I} y_j \prod_{j \in \mathcal{N}\backslash I} (1 - y_j), \tag{2}$$

i.e., the expected value of $c_{iJ}(I)$ w.r.t. the probability distribution on the joint node states. We define the feasible injection rate region of random access $\mathcal{Z}_{\text{ra}}$ as the set of all $\boldsymbol{z}$ satisfying (2). Note that $\mathcal{Z}_{\text{ra}}$ is nonconvex in general.

Example: Consider the 4-node network depicted in Fig. 1 with source 1 multicasting information to destinations 3 and 4. The hyperacrs in Fig. 1 indicate the neighborhood relations in the network; node 4 is assumed to be always idle. Table I gives the relation which hyperarcs are injected for all possible transmitter sets. For example the hyperarc injection rates of node 2 are given by

$$\begin{aligned} z_{2\emptyset} &= y_1 y_2 y_3 + (1 - y_1) y_2 y_3, \\ z_{2\{3\}} &= 0, \\ z_{2\{4\}} &= y_1 y_2 (1 - y_3), \\ z_{2\{34\}} &= (1 - y_1) y_2 (1 - y_3). \end{aligned} \tag{3}$$

Contrary to random access for routing [9], injection rates z_{iJ} of some hyperarcs (i, J) may increase with transmissions of interfering nodes, whereas injection rates into other hyperarcs decrease. In addition with the sum over multiple injecting transmitter sets $I \subset \mathcal{N}$, it implies that logarithmic transformations similar to [6] of (2) do not result in a convex formulation of the random access injection rates.

Also note that $z_{2\emptyset}$, namely the rate at which packets transmitted by node 2 are completely lost, is nonzero in general. However, only transmissions of node 3 can cause a total packet loss. This shows both the loss w.r.t. scheduling, which is free of total packet losses due to collisions, and the gain over random access in routed networks, where each packet has to be delivered to a specific node and a packet is totally lost if collision at this node occurs, although other nodes may have received the packet.

III. Achievable End-to-End Throughput

The achievable end-to-end throughput region for a multicast session using network coding, e.g. random linear network coding, is characterized in [3], [4]. The authors in [4] have shown that subgraph selection and network code construction are separable if only intra-session coding is considered.[4] For a single multicast session, with source $s \in \mathcal{N}$ and terminals $T \subset \mathcal{N}$, throughput of rate r is achievable given a fixed injection rate vector $\boldsymbol{z}$ if there exist flows $x_{ij}^{(t)}$ such that the following linear constraints are satisfied, cf. [3]:

$$r \geq 0,\ x_{ij}^{(t)} \geq 0, \quad \forall i \in \mathcal{N}, j \in N(i), t \in T, \tag{4}$$

$$\sum_{j \in N(i)} x_{ij}^{(t)} - \sum_{j: i \in N(j)} x_{ji}^{(t)} = \begin{cases} r, & \text{if } i = s, \\ -r, & \text{if } i = t, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$
$$\forall i \in \mathcal{N}, t \in T,$$

$$\sum_{j \in K} x_{ij}^{(t)} \leq \sum_{J \subset N(i)} z_{iJ} b_{iJK}, \quad \forall i \in \mathcal{N}, K \subset N(i), t \in T. \tag{6}$$

The maximal achievable throughput for any fixed $\boldsymbol{z}$ is given by

$$R(\boldsymbol{z}) = \max_{r, x_{ij}^{(t)}} r \quad \text{s.t. (4), (5), (6).} \tag{7}$$

The maximum throughput for scheduling [3] and random access is obtained by maximizing $R(\boldsymbol{z})$ w.r.t. all feasible hyperarc injection rates $\mathcal{Z}_{\text{s}}$ and $\mathcal{Z}_{\text{ra}}$, respectively. The scheduling region is defined as the convex hull of the incidence vectors of all conflicting-free hyperarc schedules.

Example: Consider again of the network depicted in Fig. 1. At node 2 the flow constraint (5) for terminal 4 is given by

$$x_{23}^{(4)} + x_{24}^{(4)} - x_{12}^{(4)} = 0 \tag{8}$$

and the capacity constraints (6) are given by

$$\begin{aligned} x_{23}^{(4)} &\leq z_{2\{3\}} b_{2\{3\}\{3\}} + z_{2\{34\}} b_{2\{34\}\{3\}}, \\ x_{24}^{(4)} &\leq z_{2\{4\}} b_{2\{4\}\{4\}} + z_{2\{34\}} b_{2\{34\}\{4\}}, \\ x_{23}^{(4)} + x_{24}^{(4)} &\leq z_{2\{3\}} b_{2\{3\}\{34\}} + z_{2\{4\}} b_{2\{4\}\{34\}} \\ &\quad + z_{2\{34\}} b_{2\{34\}\{34\}}. \end{aligned} \tag{9}$$

Note that the constraint corresponding to $K = \emptyset$ is redundant and therefore omitted. The flow constraint basically states that the amount of information that enters a node needs to leave the node, except for sources and terminals. Because of network coding, any packet can serve multiple flows for multiple destinations. The capacity constraint bounds the amount of information that is transmitted from a node i (example $i = 2$) to any subset of its neighbors. That is, the average amount of information from i to $K \subset N(i)$ intended for any terminal $t \in T$, LHS of (6) and (9), may not exceed the average amount of packets that reaches at least one node in K, RHS of (6) and (9). The latter corresponds to the value of the cut in the subset $\{i\} \cup N(i)$ that separates $\{i\} \cup (N(i) \backslash K)$ from K.

Using the hyperarc injection rates of random access (3) in the capacity constraint (9) yields

$$\begin{aligned} x_{23}^{(4)} &\leq (1 - y_1) y_2 (1 - y_3) b_{2\{34\}\{3\}}, \\ x_{24}^{(4)} &\leq y_2 (1 - y_3) \left(y_1 b_{2\{4\}\{4\}} + (1 - y_1) b_{2\{34\}\{4\}} \right), \\ x_{23}^{(4)} + x_{24}^{(4)} &\leq y_2 (1 - y_3) \left(y_1 b_{2\{4\}\{34\}} + (1 - y_1) b_{2\{34\}\{34\}} \right). \end{aligned} \tag{10}$$

Eq. (8) and (10) for all nodes and terminals constitute the achievable throughput region with random access for the example network.

A. Convex Hull of the Injection Rate Region of Random Access

For routed packet networks the inefficiency of random access w.r.t. scheduling is well-known. The random access link throughput region is generally strictly contained in the scheduling region, but its convex hull coincides with the scheduling region for single carrier networks. We derive a similar characterization for coded packet networks in this section, which is also essential for our random access algorithm, cf. Sec. IV.

Define a box $M = [\boldsymbol{a}, \boldsymbol{b}] = \{\boldsymbol{y} \in \mathbb{R}^{|\mathcal{N}|} : \boldsymbol{a} \leq \boldsymbol{y} \leq \boldsymbol{b}\}$ for any $\boldsymbol{0} \leq \boldsymbol{a} \leq \boldsymbol{b} \leq \boldsymbol{1}$ and its vertex set $\text{vert}(M) = \{\boldsymbol{y} \in \mathbb{R}^{|\mathcal{N}|} : y_i \in \{a_i, b_i\}, \forall i \in \mathcal{N}\}$.

Theorem 1: The convex hull $\text{conv}\, \mathcal{Z}(M)$ of the partial injection rate region $\mathcal{Z}(M) = \{\boldsymbol{z}(\boldsymbol{y}) : \boldsymbol{y} \in M\}$ with $\boldsymbol{z}(\boldsymbol{y})$ as defined in (2) is given by

$$\text{conv}\, \mathcal{Z}(M) = \text{conv}\{\boldsymbol{z}(\boldsymbol{y}) : \boldsymbol{y} \in \text{vert}(M)\}, \tag{11}$$

i.e., the convex hull of a finite set of points.

The theorem is an immediate consequence of the supporting hyperplane theorem, cf. [10], and the following lemma:

Lemma 1: For any $\boldsymbol{w} \in \mathbb{R}^{|\mathcal{A}|}$ there exists a vertex $\hat{\boldsymbol{y}} \in \text{vert}(M)$ that solves

$$\max_{\boldsymbol{y}} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y}) \quad \text{s.t. } \boldsymbol{y} \in M. \tag{12}$$

[4] If inter-session network coding is not considered, the problem formulation can be readily extended to multiple separate multicast sessions.

Proof: Let $\boldsymbol{y} \in M = [\boldsymbol{a}, \boldsymbol{b}]$ be an optimal solution to (12) that is not a vertex of M, i.e., there exists k with $a_k < y_k < b_k$. Using (2) and grouping w.r.t. y_k and $1 - y_k$ we get

$$\begin{aligned}
&\boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y}) = \\
&= y_k \sum_{iJ \in \mathcal{A}} w_{iJ} \sum_{\substack{I \subset \mathcal{N} \\ I \ni k}} c_{iJ}(I) \prod_{\substack{j \in I \\ j \neq k}} y_j \prod_{j \in \mathcal{N} \setminus I} (1 - y_j) + \\
&\quad + (1 - y_k) \sum_{iJ \in \mathcal{A}} w_{iJ} \sum_{\substack{I \subset \mathcal{N} \\ I \not\ni k}} c_{iJ}(I) \prod_{j \in I} y_j \prod_{\substack{j \in \mathcal{N} \setminus I \\ j \neq k}} (1 - y_j) \\
&= \alpha_k y_k + \beta_k (1 - y_k),
\end{aligned} \tag{13}$$

where $\alpha_k \geq 0$ and $\beta_k \geq 0$ depend on some y_i, $i \in \mathcal{N} \setminus \{k\}$, but not on y_k. If $\alpha_k > \beta_k$, then $\boldsymbol{y}'$ defined by $y'_k = b_k$ and $y'_i = y_i$, $i \in \mathcal{N} \setminus \{k\}$, satisfies $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y}') > \boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y})$ contradicting the optimality of $\boldsymbol{y}$. A similar contradiction occurs for $\alpha_k < \beta_k$. If $\alpha_k = \beta_k$, then $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y}') = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{z}(\boldsymbol{y})$ is also optimal. Successive application of this argument to all k where $a_k < y_k < b_k$ yields the result. ∎

Theorem 1 implies that using the convex hull of the entire random access rate region $\mathcal{Z}_{\mathrm{ra}} = \mathcal{Z}([\mathbf{0}, \mathbf{1}])$ is essentially equivalent to scheduling with the restriction that only hyperarcs satisfying (1) can be activated. This may allow for new efficient solution approaches to the scheduling problem [3] using the random access parameterization for scheduling. Additionally, further heuristic approaches to the scheduling problem using $\operatorname{conv} \mathcal{Z}_{\mathrm{ra}}$ with Lagrangian duality instead of the explicit conflict graph formulation [11] can be developed. However, both applications are beyond the scope of this work.

Nevertheless, Theorem 1 also provides an upper bound on the end-to-end throughput with random access for all probability assignments $\boldsymbol{y} \in M$ for any box $M \subset [\mathbf{0}, \mathbf{1}]$. This upper bound can be used in our global optimization algorithm for random access.

IV. Monotonic Optimization Approach to Maximum Throughput Random Access

We propose a solution to the maximum throughput problem with random access given by

$$\max_{\boldsymbol{y}} \; R(\boldsymbol{z}(\boldsymbol{y})) \quad \text{s.t. } \boldsymbol{y} \in [\mathbf{0}, \mathbf{1}]. \tag{14}$$

The objective function includes the maximization of $x_{ij}^{(t)}$ and r subject to linear constraints (4)–(6) and the polynomial expressions of the hyperarc injection rates $\boldsymbol{z}$ in terms of $\boldsymbol{y}$. To the best of our knowledge there is no way to incorporate these polynomials in a convex manner. Incidentally, the maximal throughput problem w.r.t. flows and attempt probabilities is in general a signomial program [12] that is not a geometric program. Since signomial problems are generally NP hard, we conjecture that there is no polynomial time algorithm, including standard and distributed nonlinear programming algorithms, that finds the global optimum of the maximum throughput problem with random access (14).

Despite lack of convexity, we can exploit that (14) is partially monotonic, namely, $R(\boldsymbol{z})$ is increasing in $\boldsymbol{z}$ and $z_{iJ}(\boldsymbol{y})$ can be expressed as difference of monotone (d.m.) functions[5] of $\boldsymbol{y}$. Since there is no analytic expression for $R(\boldsymbol{z}(\boldsymbol{y}))$, conversion of $R(\boldsymbol{z}(\boldsymbol{y}))$ into a d.m. function of $\boldsymbol{y}$ is intractable. Furthermore, reformulation as a d.m. function and application of the polyblock algorithm for monotonic problems [13] seems not efficient as this aggregates the detailed d.m. structure of $|\mathcal{A}|$ polynomial constraints (2) into a single d.m. function, which may lead to poor numerical performance as compared to other d.m. branch and bound mechanisms [14].

A. Branch and Bound Mechanism on Packet Injection Rates

We use a rectangular branch and bound procedure on the nonconvex variables $\boldsymbol{y}$ of the maximum throughput problem with monotonic and problem specific convex relaxation bounds. Feasible solutions for the injection rates are contained in the unit hypercube $[\mathbf{0}, \mathbf{1}] \subset \mathbb{R}^{|\mathcal{N}|}$. For notational simplicity we define $f(\boldsymbol{y}) = R(\boldsymbol{z}(\boldsymbol{y}))$.

Branch and bound is a generic method for global nonconvex optimization, e.g. integer programming, concave minimization [8], and monotonic optimization [14]. It basically consists of relaxing the feasible set,[6] subdividing the relaxed feasible set, and bounding the optimal function value over each part. Parts where the optimum cannot be located are discarded and some other parts are chosen for further refinement of the subdivision and the bounds.

Since the maximal throughput problem (14) is a d.m. optimization problem over a box region, a rectangular branch and bound procedure can be defined. That is, the feasible region $[\mathbf{0}, \mathbf{1}]$ is partitioned into a finite number of boxes. For each box M an upper bound $f_{\mathrm{u}}(M)$ on the objective function f such that $f_{\mathrm{u}}(M) \geq \max\{f(\boldsymbol{y}) : \boldsymbol{y} \in M\}$ is computed, cf. Sec. IV-B and Sec. IV-C. The box with the largest bound is then partitioned into two new boxes using rectangular bisection, namely, it is cut into two equally large boxes along its longest side.

Given the initial set of boxes $\mathcal{M}^0 = \{[\mathbf{0}, \mathbf{1}]\}$ and a feasible solution $\boldsymbol{y}^0 = \mathbf{0}$, at each iteration k perform the following operations:

Step 1 Find $M^k = \arg\max\{f_{\mathrm{u}}(M) : M \in \mathcal{M}^{k-1}\}$ and update upper bound $f_{\mathrm{u}}^k = f_{\mathrm{u}}(M^k)$.

Step 2 Take some feasible point $\boldsymbol{y}^k \in M^k$ and update lower bound $f^k = \max(f^{k-1}, f(\boldsymbol{y}^k))$.

Step 3 Subdivide M^k into two boxes M_1^k and M_2^k by rectangular bisection.

Step 4 Update relevant boxes according to $\mathcal{M}^k = \{M \in \{M_1^k, M_2^k\} \cup \mathcal{M}^{k-1} \setminus \{M^k\} : f_{\mathrm{u}}(M) \geq f^k\}$.

Convergence in the sense $f_{\mathrm{u}}^k - f^k \to 0$ is guaranteed provided that the upper bound is consistent [14], i.e., for any nested sequence of boxes M_i that shrinks to a singleton $\boldsymbol{y}$, $f_{\mathrm{u}}(M_i) \to f(\boldsymbol{y})$ as $i \to \infty$.

[5] A function f is d.m. if it can be expressed as the difference of two increasing functions.

[6] For the random access problem the feasible set of $\boldsymbol{y}$ is the unit hypercube, which is sufficiently simple. Therefore, relaxation is not necessary.

B. Monotonic Throughput Upper Bound

Given a box $M = [\boldsymbol{a}, \boldsymbol{b}] \subset [\boldsymbol{0}, \boldsymbol{1}]$ we observe that

$$z_{iJ}^{\text{dm}}(M) = \sum_{I \subset \mathcal{N}} c_{iJ}(I) \prod_{i \in I} b_i \prod_{i \in \mathcal{N} \setminus I} (1 - a_i) \qquad (15)$$

is an upper bound on (2) since $\boldsymbol{b} \geq \boldsymbol{y}$ and $\boldsymbol{1} - \boldsymbol{a} \geq \boldsymbol{1} - \boldsymbol{y}$ for any $\boldsymbol{y} \in M$. That is, we underestimate the probability of collision at all receivers provided that the packet injection rates $\boldsymbol{y}$ lie in M. As the maximal throughput $R(\boldsymbol{z})$ is increasing w.r.t. the hyperarc injection rates $\boldsymbol{z}$, we conclude that $f_{\text{dm}}(M) = R(\boldsymbol{z}^{\text{dm}}(M)) \geq f(\boldsymbol{y})$ for all $\boldsymbol{y} \in M$. Consistency of this bound follows from [14]. $f_{\text{dm}}(M)$ can be evaluated by solving a linear program.

C. Convex Relaxation Upper Bound

Given the same box $M = [\boldsymbol{a}, \boldsymbol{b}] \subset [\boldsymbol{0}, \boldsymbol{1}]$ and its vertex set $\operatorname{vert}(M)$, we consider the convex hull $\operatorname{conv} \mathcal{Z}(M)$ of all $\boldsymbol{z}(\boldsymbol{y})$ as defined in (2) with $\boldsymbol{y} \in M$. Then

$$f_{\text{cr}}(M) = \max\{R(\boldsymbol{z}) : \boldsymbol{z} \in \operatorname{conv} \mathcal{Z}(M)\} \qquad (16)$$

is an upper bound on f over M given by a convex optimization problem. As a consequence of the explicit parametrization of $\operatorname{conv} \mathcal{Z}(M)$ as convex hull of a finite set of points due to Theorem 1, f_{cr} can be evaluated by solving a single linear program. In particular, f_{cr} corresponds to the maximal throughput by using random access with a finite set of probability vectors in M each for some fraction of time. That is, convex relaxation corresponds to time sharing or scheduling of random access probability vectors.

D. Joint Upper Bound

For the algorithm we use a joint upper bound $f_{\text{u}}(M) = \min(f_{\text{dm}}(M), f_{\text{cr}}(M))$ and evaluate it on M_1^k and M_2^k in Step 4. The convex relaxation bound is typically much tighter than the monotonic bound, which is essential for the convergence speed of the branch and bound procedure. However, complexity of evaluating f_{cr} is considerably higher since the size of the associated linear program increases with the cardinality of $\operatorname{vert}(M)$, which is exponential in the dimension of M. Therefore, we compute f_{dm} first and check whether we remove the corresponding block, and only evaluate f_{cr} if this is not the case. Finally, consistency of the joint bound is ensured by f_{dm}.

V. Numerical Comparison

We compare the algorithm described in Section IV-A to different joint medium access and subgraph selection techniques: In [3], the authors present the optimal solution to the joint scheduling and subgraph selection problem. The greedy scheduling algorithm [11] is based on Lagrangian dual decomposition where a greedy method is used to solve the maximum weighted stable set problem. The dual problem is solved by a subgradient method. A suboptimal, yet practical technique for random access is presented in [5]. The subgraph selection problem is decoupled from the medium access problem and both are iteratively updated for increasing target throughput as long as the obtained coding subgraph is feasible with respect to random access.

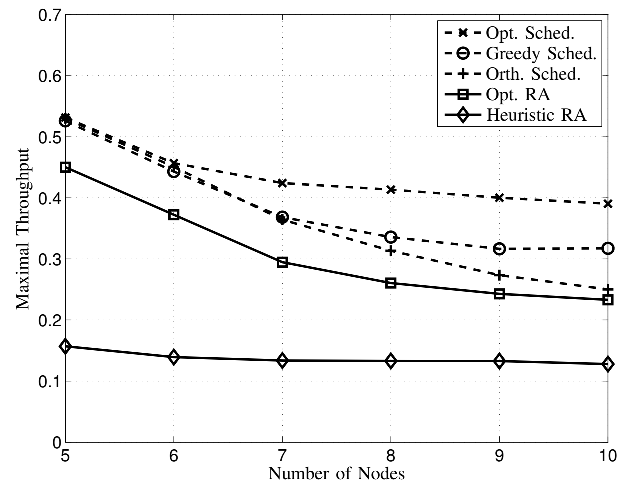

Figure 2. Maximal throughput rates vs. number of nodes. The number of neighbors is limited to 5 in this simulation.

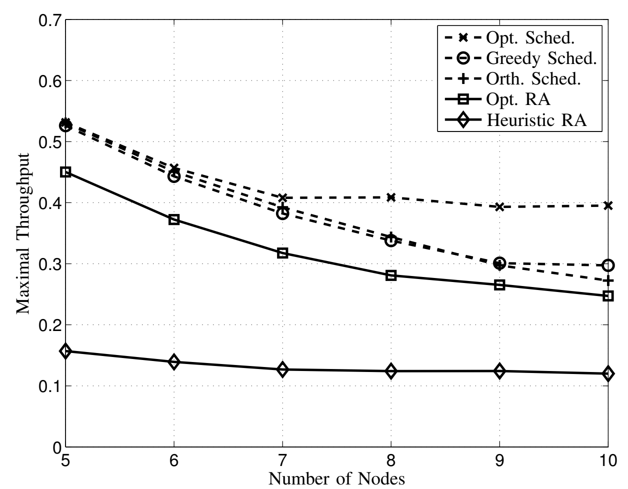

Figure 3. Maximal throughput rates vs. number of nodes. The number of neighbors is limited to 6 in this simulation.

We simulate these algorithms on random network topologies supporting one multicast session with two terminals, where the source is the leftmost node and the terminals the two rightmost nodes. Nodes are uniformly scattered over a square with unit node density. We assume that two nodes are in communication range if their distance is smaller than the radius of connectivity $d_c = 1.8$. Transmissions are subject to erasures which are due to distance attenuation and Rayleigh fading. The SNR thus depends on the distance d to the transmitter, the path loss exponent α and the fading coefficient γ, which is the realization of a unit mean exponentially distributed random variable. A packet is erased if the SNR is below a certain threshold β, i.e., $\gamma d^{-\alpha} \leq \beta$. We use $\alpha = 2$ and $\beta = 0.25$. In the simulations, we assume that erasures of packets at different receivers are independent, though this is not required by the model.

Fig. 2 and Fig. 3 show the maximum throughput as a function of the nodes in the network averaged over 500 random networks. The maximal number neighbors is restricted to 5 and 6, respectively. We compare optimal scheduling (non-conflicting simultaneous transmissions), suboptimal greedy scheduling [11], orthogonal scheduling [4] (one active node per time slot), optimal random access, and heuristic random access [5]. Optimal random access performs worse than all scheduling approaches for small to moderate size networks. The gap to optimal scheduling increases with the network size, whereas the gap to orthogonal scheduling decreases. Interestingly, there is a quite large gap between optimal and the suboptimal random access algorithm presented in [5]. The results for maximal node degree 5 and 6 are similar. However, the gap between optimal random access and the heuristic scheme increases with the maximal node degree and also optimal random access performance is slightly closer to optimal scheduling.

VI. Conclusions

We presented an algorithm that jointly optimizes random access attempt probabilities and the network coding subgraph for coded packet networks, extending similar results for scheduling [3]. Although we cannot expect to achieve optimal scheduling performance with random access, we can quantify the gap due to packet collisions. Nevertheless, simulation results show that there is some potential for new low complexity random access techniques to outperform existing approaches [5].

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft through the focus project on Communications in Interference Limited Networks (COIN) and in part by the European Commission through NEWCOM++.

References

[1] T. Ho, M. Médard, R. Kötter, D. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[2] Y. E. Sagduyu and A. Ephremides, "On Joint MAC and Network Coding in Wireless Ad Hoc Networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3697–3713, Oct. 2007.

[3] D. Traskov, M. Heindlmaier, M. Médard, R. Kötter, and D. S. Lun, "Scheduling for Network Coded Multicast: A Conflict Graph Formulation," in *IEEE GLOBECOM Workshops*, Nov. 2008, pp. 1–5.

[4] D. S. Lun, N. Ratnakar, M. Médard, R. Kötter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost Multicast over Coded Packet Networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2608–2623, Jun. 2006.

[5] D. Traskov, D. S. Lun, R. Kötter, and M. Médard, "Network Coding in Wireless Networks with Random Access," in *IEEE International Symposium on Information Theory, 2007. ISIT 2007.*, Jun. 2007, pp. 2726 –2730.

[6] X. Wang and K. Kar, "Cross-Layer Rate Optimization for Proportional Fairness in Multihop Wireless Networks with Random Access," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1548–1559, Aug. 2006.

[7] J.-W. Lee, M. Chiang, and A. R. Calderbank, "Utility-Optimal Random-Access Control," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2741 –2751, Jul. 2007.

[8] R. Horst and H. Tuy, *Global Optimization*. Springer, 1996.

[9] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1992.

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

[11] M. Heindlmaier, D. Traskov, M. Médard, and R. Kötter, "Scheduling for Network Coded Multicast: A Distributed Approach," in *IEEE GLOBECOM Workshops*, Nov. 2009.

[12] M. Chiang, "Geometric Programming for Communication Systems," *Foundations and Trends in Communications and Information Theory*, vol. 2, pp. 1–154, 2005.

[13] H. Tuy, "Monotonic Optimization: Problems and Solution Approaches," *SIAM Journal on Optimization*, vol. 11, no. 2, pp. 464–494, 2000.

[14] H. Tuy, F. Al-Khayyal, and P. Thach, *Essays and Surveys in Global Optimization*. Springer, 2005, ch. Monotonic Optimization: Branch and Cut Methods, pp. 39–78.