An Active Constraint Method for Optimal Multicast Message Transmission in Wireless Networks

Joe Hodgskiss[1], Armin Dekorsy[2] and Jörg Fliege[3]

[1] School of Mathematics, University of Birmingham, UK,
[2] Dept. of Communications Engineering, University of Bremen, Germany
[3] School of Mathematics, University of Southampton, UK

Abstract. Efficiently transmitting data in wireless networks requires joint optimization of routing, scheduling, and power control. Also, transmitting data to multiple destinations (multicasting) in an optimal way typically involves the solution of difficult combinatorial optimization problems. To avoid such difficulties, we use the concept of network coding to design an optimization problem for the problem of transmitting multiple multicast messages through a time-slotted multihop wireless network whilst allowing multipath routing. Apart from the mathematical model, we introduce a block co-ordinate descent algorithm that uses information about active constraints and the decoupled nature of the problem, and prove its convergence to a global optimum under certain assumptions. Numerical examples of wireless mesh backhaul networks with fixed nodes show that the algorithm rapidly provides optimal solutions in networks with no interference, and good solutions in a range of more challenging circumstances.
Keywords: multicast, network coding power control, resource allocation, wireless mesh networks.

1 Introduction

There is an increased interest in communication via wireless mesh networks such as ad-hoc, sensor or wireless mesh backhauling networks nowadays [1, 2]. In wireless networks the link capacities are variable quantities and can be adjusted by the allocation of resources such as bandwidth, power allocation, and time-slot length to fully exploit network performance. For efficient data transmission an integrated routing, time scheduling and power control optimization strategy is therefore required. This strategy has to take different transmission constraints into account, for example maximum available power level or limited buffer sizes at nodes. Also, the inherent decentralized nature of wireless mesh networks mandates that distributed algorithms should be developed to implement the joint routing, scheduling and power control optimization.

Cruz and Santhanam [3] have addressed the problem of finding an optimal link scheduling and power control policy while minimizing total average power consumption. Their algorithm is designed for single-path routing only, does not consider buffer limitations and has a worst case exponential complexity. Li and Ephremedis [4] solve at first power control and scheduling jointly. With the power values obtained, a routing distance is then calculated which in turn is used by Bellman-Ford routing. However, the proposed separation is performed by not considering the combinatorial structure of the entire routing, scheduling and power control problem. Although computationally inexpensive, the algorithm can end up in a suboptimal solution. Also, it neglects multiple path routing as well as buffer restrictions. Xiao et.al. [5] proposed dual decomposition as a promising decomposition approach. By dual decomposition the overall problem is split into several sub-problems while a master dual problem coordinates them.

Until recently, efficiently transmitting multicast messages through networks required the solution of difficult combinatorial optimization problems based around embedding Steiner trees into a particular graph [6]. For such problems, a centralized polynomial time algorithm has been introduced in [7], while for a certain class of networks, a distributed coding strategy has been described in [8]. However, using network coding, one can construct a multicast transmission from a group of unicast transmissions which do not compete for bandwidth. A big step forward in the practical use of network coding came with the idea of random linear network coding, discussed in [9], [10] and [11]. There, it has been shown that with high probability, a random linear network code will result in all receivers being able to retrieve the message. In terms of incorporating network coding into a nonlinear optimization problem for cross layer optimization, an important result can be found in [12] which demonstrates that with network coding, one can consider a multicast as a collection of unicasts which do not compete for channel capacity. This idea has then been taken forward in [13] with the introduction of "conceptual flows" that satisfy flow constraint laws but which do not compete for channel capacity. This is included in a joint power control and routing optimization problem on a non-timeslotted network by setting the actual flow along an edge as the maximum of all conceptual flows along that edge. The authors then go on to solve this problem using a dual decomposition approach similar to [14].

One possible strategy to design a distributed implementation is to break up the given problem into manageable sub-problems and solve these sub-problems by distributed iterative algorithms. This is the strategy that we will use in this paper. We will consider joint routing, time-scheduling and power control for single frequency wireless mesh networks. The wireless transmissions are arranged in time-slots. However, we take into account that simultaneously active transmissions suffer from multiple access interference. Further to this, we use the recently developed idea of network coding to incorporate multicast message transmissions into our optimization problem. This is achieved by forming linear combinations of data packets, creating encoded packets which can be useful to all multicast destinations. Using this idea, we will develop an optimization problem for the transmission of multiple unicast and multicast messages through a network.

While dual decomposition is a universal approach to solve such optimization problems [5, 19], it does not exploit the specific structures apparent here. By contrast, we propose a novel method that explicitly exploits the combinatorial structure of a joint routing, time-scheduling and power control problem by means of an active constraint method. In particular, the approach is as follows: We separate scheduling from routing and power allocation by including it in the constraint set of a simultaneous routing and power control problem. For scheduling, several well known approximations such as Greedy based approaches exist [15, Section 3.7] that we can leverage on. The constraints we use in the optimization problem are induced by a pre-calculated colouring of the network that, in turn, reflects the scheduling decisions of any arbitrary scheduler.

For solving the optimization problem considered, we introduce a block coordinate descent method to solve the simultaneous routing and power control problem and prove that, when the network does not suffer from interference, the algorithm converges to a global minimizer. The main points of the algorithm are as follows: (1) We re-write the optimization problem to an equivalent problem by applying the active constraint method. (2) We decouple the equivalent problem by solving a (convex) network and a (convex) power assignment problem separately. (3) Iterations are performed by switching between the two sub-problems for which network and power variables act as interchanging parameters.

Finally, we examine the performance of this algorithm when applied to networks suffering from interference. These numerical tests are performed by applying the algorithm, as well as a black box nonlinear solver run from multiple starting points, to a wireless cellular mesh backhauling network [1, 2]. The backhauling network describes a "regular" cellular network. This models the situation where, in order to save infrastructure expenses of laying cable or fiber to each node (base station), we try to extend the range of a given source node with wired backhaul connection by using several other nodes. These intermediate nodes have no wired connection and can only communicate with the backhaul via the source node by wireless mesh communications. The simulation set-up correctly models mobile radio channel characteristics such as path-loss and slow fading. Testing indicates that the algorithm provides optimal solutions in zero interference cases, and good, near optimal solutions in many other interference affected cases.

The rest of this paper is organized as follows. In Section 2 we describe the network model used for the wireless data network. In Section 3 we describe how we use the principle of network coding to incorporate multicast message transmission into our model. In section 4 we formulate the optimization problem. The co-ordinate descent algorithm for solving the joint routing and power control problem is presented in Section 5. Finally, in Section 6 we apply the algorithm to a wireless backhaul network and present the simulation results. We conclude the paper in Section 7.

2 Network Model

In this section we will introduce the model of the network through which the data is to be transmitted. Consider a set of V communication nodes, with $s \in V$ a source node and $D \subset V$ a set of receiver nodes. We assume $s \notin D$. Consider a transmission of identical data from s to all $d \in D$. If D consists of exactly one node, we term this transmission a **unicast**. If $D \cup s = V$, we term this transmission a **broadcast**. Otherwise, we term the transmission a **multicast**.

The transmission problem we are facing is to transmit messages indexed by m, each of size S_m bits via a multiple hop wireless network. Each message has its source node s_m and a set of destination nodes D_m, where $s_m \notin D_m$. Let M be an index set for the set of messages, so $m \in M$. With *multiple path routing* each message can be transmitted via several paths from its source node to any of its destination nodes. Thus, intermediate nodes can send parts of messages to many receivers and receive parts of messages from many transmitters.

We denote the nodes by $v \in V$ where V is finite. At any time, each of these nodes $v \in V$ can map any parts of messages $m \in M$ onto a single link e for transmission. The set of all links is denoted by E. A wireless communication link corresponds to an edge $e = (u, v)$ between two nodes $u, v \in V$ and is described by the ordered pair $(u, v) \in V \times V$ such that u transmits information directly to v. Moreover, we assume that $(v, v) \notin E$ for all $v \in V$. We have that $G := (V, E)$ is a directed graph with node set V and edge set E. For an arbitrary node $v \in V$, denote by $E^+(v) := \{e \in E \mid e = (v, w) \in E\}$ and $E^-(v) := \{e \in E \mid e = (w, v) \in E\}$ the set of outgoing and incoming edges within E at the node v, respectively. A link represents a wireless resource characterized by a given bandwidth, time duration, space fraction, or by a given code assignment.

We assume a time-slotted single frequency network for which the time is divided into equal slots of length τ seconds while all nodes occupy the same frequency band of bandwidth B hertz. Time slots are indexed by $t \in T$, with T as an index set. We take time scheduling into account by assuming that there is given a coloring of the nodes such that adjacent nodes do not have the same color (half-duplex constraint) [4]. That is, we are given a number C and a function $co_V : V \longrightarrow \{1, ..., C\}$ such that $co_V(v) \neq co_V(w)$ for all nodes $v, w \in V$ with $(v, w) \in E$. Here, C is at least as large as the chromatic number of G. Computing such a coloring can be done by a Greedy approach [15] and is not the focus of this work.

To take delay constraints into account we introduce t_{max} as the maximum number of time slots a message is allowed before arriving at its destination, i.e $T := \{1, ..., t_{max}\}$.

The interference model we consider includes multiple access interference caused by simultaneously active transmissions that can not be perfectly separated by e.g. code- or space division multiple access (CDMA/SDMA) techniques. Thus, let $E_{e,t}$ be the set of edges interfering edge e at time t. The signal attenuation from node u to node v is $a_t(u, v)$ and we assume it remains unchanged within the duration of a time slot t. We further assume perfect knowledge of

$a_t(u, v)$ at the corresponding senders. Let $T(e)$ be the transmitting node and $R(e)$ be the receiving node of edge e. Hence, $a_t(T(l), R(e))$ denotes the attenuation a signal suffers that is transmitted from $T(l)$ but received by node $R(e)$. For link e such a signal represents multiple-access interference that is caused by link l.

Furthermore, with $p_{e,t}$ as the (transmit) power to be allocated to link e at time slot t, the received signal power at node $R(e)$ from the transmitter $T(e)$ is given by $a_t(T(e), R(e))p_{e,t}$. We simplify this interference model, and introduce shorter notation by assuming all signal attenuations to be time-invariant, and replace $a_t(T(l), R(e))$ with $a_{l,e}$ since it is clear we are referring to the transmitter of edge l and the receiver of edge e. We define the **signal-to-interference-plus-noise ratio** (SINR) of edge $e \in E$ at time slot $t \in T$ as

$$SINR_{e,t} = \frac{a_{e,e} p_{e,t}}{\sum\limits_{\substack{l \in E_{e,t} \\ l \neq e}} a_{l,e}\, p_{l,t} + \sigma_e^2} \tag{1}$$

with σ_e^2 as additive noise power of edge e. If we only assume thermal noise to be the same for all edges, we have $\sigma_e^2 = BN_0$ with noise spectral density N_0.

3 Handling Multicast Transmissions

We will now outline how we are able to build an optimization problem for the task of optimally transmitting multiple multicast and unicast messages. The standard approach is a combinatorial approach based on packing steiner trees, [6]. This method is computationally expensive and using this approach it is not always possible to meet the theoretical maximum rate for multicast transmissions, [11]. The approach we pursue is to use the novel concept of network coding. Here, intermediate nodes are given the ability to perform elementary operations on incoming data packets. Unlike the tree finding approach, this method requires no combinatorial optimization, and can also be shown to meet the min-mincut bound for multicast transmissions, [16].

3.1 The Network Coding Principle

The principle of network coding is best illustrated through a simple example as shown in Figure 1 on the next page. Consider two intermediate nodes in a network, u and v. Suppose packets A and B arrive at node u. Imagine multicast destination d_1 beyond node v requires only packet A in order to have the complete transmission, and destination d_2 only requires packet B (both destinations have received all the other packets they need through other paths). Without network coding, both packets, A and B need to be transmitted from u to v, and both need to be stored in a buffer at v. With network coding, a random combination of the two packets is made at u, $X = \lambda A + \mu B$. This encoded packet, along with the vector (λ, μ) is transmitted to, and stored at, node v before being forwarded on to the two destinations. d_1, with knowledge of B, X and (λ, μ) is able to recover A. Similarly, d_2 is able to recover B. With this simple example we were able to transmit and store half as many packets between u and v, at the cost of some simple linear algebra and the transporting of the encoding vector.

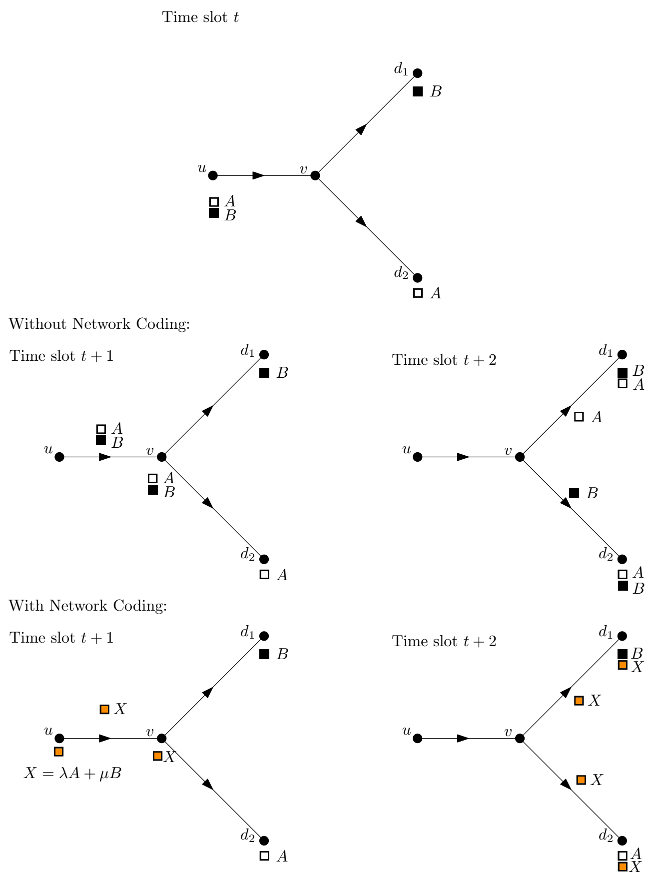

Fig. 1. *Using Network Coding prevents packets bound for different destinations from competing for bandwidth and storage space*

In traditional routing, intermediate nodes are only capable of storing, forwarding and replicating any packets they receive. The principle of Network Coding is to open a whole new set of operations at each node. As well as storing, forwarding and replicating, the nodes are given the ability to combine incoming packets, thus packets on a nodes outgoing edges can be formed of a mixture of packets that previously arrived on incoming edges. Rather than the original required packets, a destination node is presented with a set of recombined packets of data as well as the information required for retrieving the original information.

The idea of Network coding is a relatively new one, first suggested in 2000 by Ahlswede et al, [17]. Over recent years, much research has focussed on ways in which Network Coding could be employed to improve many aspects of information networks, including improving throughput, reducing delays, and improving robustness. A good overview of network coding can be found in [18].

The benefit of using network coding is that, by forming linear combinations of data packets, nodes are able to transmit new packets containing information useful to multiple destinations, where otherwise this may not have been possible. This redistribution of data means that if the network is able to support a unicast flow through the network from the source to each of its destinations separately, it is able to support a multicast flow from the source to all the destinations. This can be formalized as follows:

Theorem 1. *Consider a multicast from a source node s to a destination set $d_1, \ldots, d_k$ in a fixed capacity network. If the network is able to support a* ***unicast*** *at rate α from s to d_i $\forall$ $i = 1, \ldots k$, using traditional routing with no other traffic in the network, the network is able to support a* ***multicast*** *from s to $d_1, \ldots, d_k$ at rate α using linear network coding.*

Proof: See [12].

Of course, if it is difficult to calculate the specific Network Code needed to achieve this multicast rate, the computational effort and time required to find the solution might render the whole method too expensive. With random linear network coding, the need to calculate a coding strategy is side-stepped by allowing each node to choose its encoding coefficients at random from a large enough finite field. In [9] it is shown that, for large enough finite fields, random linear network coding achieves the required multicast flow, with probability of a succesful code transmission converging to one as the field size increases. In [11], simulation shows that with fields as small as $\mathbb{F}_{2^8}$, the actual probability of decoding failure becomes negligible. This means intermediate nodes no longer need to be aware of an over-arcing network coding strategy and can simply form random linear combinations of any incoming packets.

3.2 Building a multicast from non-competing unicasts

The key thing to notice from this discussin is that flows bound for different destinations within the same multicast do not compete for communication bandwidth when sharing an edge. This means we can treat the multicast flow as a set of unicast flows, all of which safisfy standard routing flow constraints, but where separate unicast flows along the same edge do not compete for capacity. The actual number of coded packets that need to be transmitted down a given edge during a given time slot is simply the maximum of the number of packets that the separate unicast flows require. Since each multicast can be broken down into non-competing unicasts, it is clear that these unicasts do not represent the movement and storage of physical data, but represents the amount of encoded data each destination node requires on any given edge or buffer in a particular time slot. We therefore refer to these unicast flows used to build the multicast flow as **conceptual flows**, using the same naming convention as [13]. Similarly, in order to construct each conceptual flow, we require conceptual buffer variables to ensure that each of the unicast flows we are using satisfies flow constraints at each node. We then require that the physical buffer at each node contains enough encoded packets during any given time slot to satisfy the demands of any of the conceptual buffers for that time slot.

Having introduced how we use network coding to build each multicast from non-competing unicasts, we are now in a position to introduce the decision variables of our optimization problem.

Firstly, we have conceptual flow variables, $c_{e,t}^{m,d}$: The amount of data transmitted along edge e in timeslot t forming part of the conceptual flow of message m from the multicast source s_m to destination $d \in D_m$.

Secondly, we have conceptual buffer variables, $d_{n,t}^{m,d}$: The amount of data stored at node n in timeslot t forming part of the conceptual flow of message m from the multicast source s_m to destination $d \in D_m$.

In order to model the flow of the actual encoded data through the network, we also need physical flow and buffer variables, $f_{e,t}^m$, $b_{n,t}^m$, representing the amount of encoded data from message m transmitted along a given edge in a given time slot, and the amount of encoded data from message m stored at a given buffer in a given time slot.

Communication variable $p_{e,t} \in \mathbb{R}$ is the transmit power allocated to edge e at time slot t to transmit the total traffic on edge e (in Watt). If we stack the different variables to vectors we obtain $\mathbf{c} = (c_{e,t}^{m,d})$, $\mathbf{d} = (d_{e,t}^{m,d})$, $\mathbf{f} = (f_{v,t}^m)$, $\mathbf{b} = (b_{v,t}^m)$, and $\mathbf{p} = (p_{e,t})$. We further use the following parameters. Let $S_m \in \mathbb{R}^+$ be the size of message m (in bits) and $B_v \in \mathbb{R}^+$ be the maximum total buffer size at node v (in bits). Power constraints are $P_v^{max} \in \mathbb{R}^+$ as the maximum transmission power of a node (in Watt) assumed to be the same for all nodes and $P_e^{max} \in \mathbb{R}^+$ as the maximum transmission power per edge (in Watt).

4 Optimization Problem

4.1 Problem Description

Let us consider the operation of a wireless data network with the objective to minimize a convex cost function $\Phi(\mathbf{p}, \mathbf{f}, \mathbf{b})$ (or to maximize a concave utility function) which we assume to be monotone increasing in $\mathbf{p}$. The design variables $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{f}$ and $\mathbf{p}$ are subject to a number of constraints. With $(e \in E, v \in V, t \in T)$ we require the power constraints

$$p_{e,t} \geq 0, \tag{2}$$

$$p_{e,t} \leq P_e^{max}, \tag{3}$$

$$\sum_{e \in E^+(v)} p_{e,t} \leq P_v^{max}. \tag{4}$$

forming the polyhedral set

$$C_p := \{\mathbf{p}|\mathbf{p} \text{ fulfills (2), (3) and (4)}\}.$$

Similar to power constraints, we require that flow constraints form a polyhedral set C_c. We assume that given source nodes s_m have to transmit messages of sizes S_m to destination sets D_m in a given time t_{max}, and so the polyhedral set C_c is defined by the equalities and inequalities

$$c_{e,t}^{m,d} \geq 0 \quad (e \in E, m \in M, d \in D_m, t \in T) \tag{5}$$
$$d_{v,t}^{m,d} \geq 0 \quad (v \in V, m \in M, t \in T) \tag{6}$$
$$b_{v,t}^{m} \leq B_{v,m} \quad (v \in V, m \in M, t \in T) \tag{7}$$
$$b_{v,1}^{m} = 0 \quad (m \in M, v \in V \setminus \{s_m\}) \tag{8}$$
$$b_{v,t_{max}}^{m} = 0 \quad (m \in M, v \in V \setminus \{D_m\}) \tag{9}$$
$$d_{s_m,1}^{m,d} = S_m \quad (m \in M, d \in D_m), \tag{10}$$
$$d_{d,t_{max}}^{m,d} = S_m \quad (m \in M, d \in D_m) \tag{11}$$
$$f_{e,t}^{m} = 0 \quad (m \in M, co_E(e) \neq co_T(t)) \tag{12}$$
$$d_{v,t+1}^{m,d} - d_{v,t}^{m,d} = \sum_{e \in E^-(v)} c_{e,t}^{m,d} - \sum_{e \in E^+(v)} c_{e,t}^{m,d},$$
$$(m \in M, d \in D_m, v \in V, t \in T \setminus \{t_{max}\}) \tag{13}$$
$$\max_{d \in D_m} c_{e,t}^{m,d} \leq f_{e,t}^{m} \quad (e \in E, m \in M, d \in D_m, t \in T) \tag{14}$$
$$\max_{d \in D_m} d_{e,t}^{m,d} \leq b_{e,t}^{m} \quad (v \in V, m \in M, d \in D_m, t \in T). \tag{15}$$

Equation (5) forbids negative conceptual data flows, and with (14) also forbids negative physical flows. Equation (6) forbids negative conceptual buffer variables, and with (15) also forbids negative physical buffer variables. Equation (7) avoids buffer overload, while (9) and (8) ensure all buffers except the source are empty at the start of the transmission, and all buffers except the destinations are empty at the end of the transmission.

Equation (10) forces the full message to be at the source of each conceptual flow at the start of the transmission. To account for delay constraints (11) ensures that messages reach their destinations completely at t_{max} at latest.

Coloring is ensured by (12), and (13) is a modified Kirchhoffs Law [19] for each individual conceptual flow.

Equation (14) ensures that the physical encoded data travelling along any given edge is enough to satisfy all the conceptual flows using that edge. Similary, (15) ensures that each buffer contains enough physical encoded data to satisfy all conceptual buffers at that node. We can now formalize the optimization problem we are interested in solving:

$$
\begin{aligned}
\text{minimize} \quad & \Phi(\mathbf{p}, \mathbf{f}, \mathbf{b}) \\
\text{subject to} \quad & (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in C_c \\
& \mathbf{p} \in C_p \\
& \sum_{m \in M} (1 + \kappa_m) f^m_{e,t} \leq R_{e,t}(\mathbf{p}) \quad e \in E^+(v), \quad t \in T.
\end{aligned} \tag{16}
$$

By the last constraints we model that the amount of information (in bits) we can transmit on a single wireless link e at time slot t is bounded from above by a maximum mutual information bound $R_{e,t}(\mathbf{p})$ that itself depends on the power setting. The last constraints of (16) are the only constraints coupling network flow variables $(\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f})$ with communication variables $\mathbf{p}$, and so we call them **coupling constraints** [5]. They represent the most challenging constraints of the problem as they are also the only non-linear constraints and are indeed non-convex in $\mathbf{f}$ and $\mathbf{p}$.

The factor $(1 + \kappa_m)$ is required to model the transmitting of the encoding coefficients along with the encoded data in order to enable decoding at the destination nodes. We refer to κ_m as the **encoding information overhead**, and in typical applications it can be expected to be ≈ 0.05.

All the other constraints are either constraints for the network flow variables or for the communication variables only. Assuming time-invariant channel conditions within the duration of a single time slot t, function $R_{e,t}(\mathbf{p})$ describes the amount of information of edge e and can be expressed by the well-known Shannon formula

$$
R_{e,t}(\mathbf{p}) = B \cdot \tau \cdot \log_2\big(1 + \frac{1}{\Omega_e} SINR_{e,t}\big) \quad (e \in E, \ t \in T). \tag{17}
$$

Here, $SINR_{e,t}$ is as described in 1. For each edge e, the factor $\Omega_e \in \mathbb{R}^+_0$ represents any implementation margin relative to the maximum mutual information given by the Shannon formula [20]. In practice, achieving this mutual information requires adaptive modulation and coding.

Before going on to outline an algorithm for solving the optimization problem, we present a useful result regarding the location of solutions to this problem:

Theorem 2. *Suppose that the objective function $\Phi : (\mathbf{p}, \mathbf{b}, \mathbf{f}) \mapsto \Phi(\mathbf{p}, \mathbf{b}, \mathbf{f})$ is strictly monotone in $\mathbf{p}$, and that we want to solve the optimization problem detailed in (16). Then all coupling constraints are active at each locally optimal solution of this problem.*

Proof. Let $\mathbf{x} = (\mathbf{p}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f})$ be a feasible point of the problem (16) such that not all coupling constraints are active. Let $I \subseteq E \times T$ be the set of all edge-timeslot pairs for which the coupling constraint is inactive, that is for $(e, t) \in I$,

$$
\sum_{m \in M} (1 + \kappa_m) f^m_{e,t} < B\tau \log_2 \left(1 + \frac{a_{e,e} p_{e,t}}{\sum_{l=1 \ldots E, l \neq e} a_{l,e} p_{l,t} + BN_0} \right) \tag{18}
$$

If we can show there is a feasible descent direction from this arbitrary point for which not all coupling constraints are active, then we have shown this point is not a local minimizer and therefore any local minimizer must have all coupling constraints active.

Pick any $(e,t) \in I$ and consider the vector $\mathbf{d} = (0,0,\ldots,0,-1,0,\ldots,0)^T$ where the -1 is in the position corresponding to power variable $p_{e,t}$. Clearly this is a descent direction since the objective function is strictly monotone in $\mathbf{p}$. We now need to check that for suitably small $\alpha \in \mathbb{R}^+$, $\mathbf{x}_{\text{new}} := \mathbf{x} + \beta\mathbf{d}$ is feasible for all $\beta \in [0,\alpha]$.

Clearly $\mathbf{x}_{\text{new}}$ will remain in $\mathbf{C_c}$ since we have not altered $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ or $\mathbf{f}$. Since we have not increased any power variables, there is no chance we have violated the maximum node power or maximum edge power constraints by moving to $\mathbf{x}_{\text{new}}$.

Since $\sum_{m\in M}(1+\kappa_m) f^m_{e,t} \geq 0$, by (18) we have that $p_{e,t} > 0$ and so $p_{e,t} - \beta \geq 0$ for $\beta \leq p_{e,t}$ and so for suitably small β we do not violate the non-negative power constraint.

In constraint (18), by continuity there exists some $q < p_{e,t}$ such that

$$\sum_{m\in M}(1+\kappa_m) f^m_{e,t} = B\tau \log_2\left(1 + \frac{a_{e,e}q}{\sum_{l=1\ldots E, l\neq e} a_{l,e}p_{l,t} + BN_0}\right).$$

As long as $p_{e,t} - \beta \geq q$, constraint (18) remains feasible at $\mathbf{x}_{\text{new}}$. For all other coupling constraints, $p_{e,t}$ appears on the denominator in the log term on the right hand side, and so decreasing $p_{e,t}$ increases the right hand side and therefore cannot compromise feasibility. $\mathbf{x}_{\text{new}} = \mathbf{x} + \beta\mathbf{d}$ is therefore feasible $\forall\, \beta \in [0, p_{e,t} - q]$. We have found a feasible descent direction from $\mathbf{x}$ and so x was not a local minimizer.

With this result we are able to rewrite the problem in a number of equivalent formulations. Using this along with the largely decoupled nature of the problem we can develop a block co-ordinate descent algorithm.

5 A Primal Co-ordinate Descent Algorithm

The key motivating reason for employing a block co-ordinate approach to solving the optimization problem is as follows: The variables of this problem can be split into flow variables $(\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f})$ and physical communication resource variables $\mathbf{p}$, and if we fix either set of variables the problem we are left with closely resembles, or is equivalent to an existing well studied optimization problem. Specifically, for fixed power variables, we can reformulate the problem as a nonlinear minimum cost multi-commodity flow problem, and for fixed flow variables we have a standard power control problem. Both of these subproblems can be formulated as convex optimization problems, and many solution methods exist for solving these subproblems in a distributed manner.

To make use of the apparent simplicity of the subproblems, we employ a primal block co-ordinate descent method where we alternately fix the physical communication variables, $\mathbf{p}$, and solve the problem over the network flow variables $(\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f})$, and then fix the $(\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f})$ variables and solve over the $\mathbf{p}$ variables. The idea would be to toggle between these two problems until an optimal point is found.

By Theorem 2, we are able to reformulate the optimization problem with equality in the coupling constraints,

$$\sum_{m\in M}(1+\kappa_m)f_{e,t}^m = R_{e,t}(\mathbf{p}),$$

and then rearranging for $p_{e,t}$,

$$p_{e,t} = \frac{1}{a_{e,e}}(2^{\frac{1}{B\tau}\sum_{m\in M}(1+\kappa_m)f_{e,t}^m} - 1)(\sum_{k\neq e} a_{k,e}p_{k,t} + \sigma^2) := J_{e,t}(\mathbf{f}, \mathbf{p}).$$

Stacking up the components $J_{e,t}$ into a vector, we can then substitute $\mathbf{J}(\mathbf{f}, \mathbf{p})$ for $\mathbf{p}$ in the objective function of our optimization problem, before relaxing the equality constraint in the coupling constraint to $p_{e,t} \geq J_{e,t}(\mathbf{f}, \mathbf{p})$ to obtain another formulation,

$$\begin{aligned} \text{minimize} \quad & \Phi(\mathbf{J}(\mathbf{f}, \mathbf{p}), \mathbf{f}, \mathbf{b}) \\ \text{subject to} \quad & (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in C_c \\ & \mathbf{p} \in C_p \\ & p_{e,t} \geq J_{e,t}(\mathbf{f}, \mathbf{p}), \quad e \in E, t \in T. \end{aligned} \tag{19}$$

which is almost equivalent to the original formulation, as described in the following lemma:

Lemma 1. *Assume that $\Phi(\mathbf{p}, \mathbf{b}, \mathbf{f})$ is strictly monotone in $\mathbf{p}$. For each local minimizer $(\mathbf{b}^0, \mathbf{c}^0, \mathbf{d}^0, \mathbf{f}^0, \mathbf{p}^0)$ of (19) there exists a local minimizer $(\mathbf{b}^0, \mathbf{c}^0, \mathbf{d}^0, \mathbf{f}^0, \mathbf{p}^1)$ of (19) with:*

- *All coupling constraints are active at $(\mathbf{b}^0, \mathbf{c}^0, \mathbf{d}^0, \mathbf{f}^0, \mathbf{p}^1)$.*
- *$(\mathbf{b}^0, \mathbf{c}^0, \mathbf{d}^0, \mathbf{f}^0, \mathbf{p}^0)$ and $(\mathbf{b}^0, \mathbf{c}^0, \mathbf{d}^0, \mathbf{f}^0, \mathbf{p}^1)$ have the same objective function value.*

Proof. Consider a local minimizer $\mathbf{x} = (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}, \mathbf{p})$ of (19).
Case 1: All coupling constraints are active and we are done.
Case 2: $\exists\, I \neq \emptyset, I \subseteq E \times T$ such that $p_{e,t} > J_{e,t}(\mathbf{f}, \mathbf{p})\ \forall\ (e,t) \in I$.
Consider $(e,t) \in I$.
Case 2a: $a_{e,l} \neq 0$ for some $l \neq e$, s.t. $\sum_{m\in M} f_{l,t}^m > 0$, and so $p_{e,t}$ appears in $J_{l,t}(\mathbf{f}, \mathbf{p})$ since

$$J_{l,t}(\mathbf{f}, \mathbf{p}) = \frac{1}{a_{l,l}}(2^{\frac{1}{B\tau}\sum_{m\in M} f_{l,t}^m} - 1)(\sum_{k\neq l} a_{k,l}p_{k,t} + \sigma^2)$$

Since $\Phi(\mathbf{b}, \mathbf{f}, \mathbf{p})$ is strictly monotone in $\mathbf{p}$,

$$\hat{\mathbf{p}} \leq \bar{\mathbf{p}} \Rightarrow \Phi(\mathbf{b}, \mathbf{f}, \hat{\mathbf{p}}) \leq \Phi(\mathbf{b}, \mathbf{f}, \bar{\mathbf{p}})$$

Here the inequality is a vector inequality, ie $\hat{\mathbf{p}} \leq \bar{\mathbf{p}} \Leftrightarrow \hat{p}_{e,t} \leq \bar{p}_{e,t} \forall\ e \in E, t \in T$. Further, by strict monotonicity, if the vector inequality is strict in at least one co-ordinate, then the inequality in the function values is strict also. Similarly,

$$\mathbf{J}(\mathbf{f}, \hat{\mathbf{p}}) \leq \mathbf{J}(\mathbf{f}, \bar{\mathbf{p}}) \Rightarrow \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f}, \hat{\mathbf{p}})) \leq \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f}, \bar{\mathbf{p}})).$$

Now consider $\mathbf{p}_{\text{new}}$ identical to $\mathbf{p}$ but with $p_{e,t}$ replaced by $(p_{e,t} - \delta)$ for some $\delta > 0$. Then,

$$J_{l,t}(\mathbf{f}, \mathbf{p}_{\text{new}}) = \frac{1}{a_{l,l}}(2^{\frac{1}{B\tau}\sum_{m \in M} f_{e,t}^m} - 1)(\sum_{k \neq l} a_{k,l} p_{k,t} + a_{e,l}(p_{e,t} - \delta) + \sigma^2) < J_{l,t}(\mathbf{f}, \mathbf{p}). \quad (20)$$

Clearly, decreasing $p_{e,t}$ to $p_{e,t} - \delta$ cannot increase $J_{k,t}$ for any $(k,t) \in E \times T$, and so $\mathbf{J}(\mathbf{f}, \mathbf{p}_{\text{new}}) \leq \mathbf{J}(\mathbf{f}, \mathbf{p})$, with strict inequality in at least one co-ordinate. By strict monotonicity, $\Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f}, \mathbf{p}_{\text{new}})) < \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f}, \mathbf{p}))$, and so the descent vector used in the proof of Theorem 2 is again a descent direction in this case, and for the same reason as in that proof it is feasible. $\mathbf{x}$ is therefore not a local minimizer, so Case 2a cannot occur.
Case 2b: $p_{e,t}$ does not appear with a non-zero coefficient in any co-ordinate of $\mathbf{J}(\mathbf{f}, \mathbf{p})$, and therefore does not appear in the objective function. We can therefore reduce $p_{e,t}\ \forall\ (e,t) \in I$ until $p_{e,t} = J_{e,t}(\mathbf{f}, \mathbf{p})\ \forall\ (e,t) \in I$, without compromising feasibility of any other constraint, to get a new point with the same objective function value, with all coupling constraints active, without altering $\mathbf{b}, \mathbf{c}, \mathbf{d}$ or $\mathbf{f}$.

We are now able to use these equivalent, and near equivalent formulations to decompose the optimization problem into two convex subproblems.

Network flow (routing) subproblem We assume feasible fixed power variables $\mathbf{p} \in C_p$. Using formulation (19) we need to solve the optimization problem

$$\begin{array}{ll} \text{minimize} & \Phi(\mathbf{J}(\mathbf{p}, \mathbf{f}), \mathbf{f}, \mathbf{b}) \\ \text{subject to} & (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in C_c, \\ & \sum_{m \in M} (1 + \kappa_m) f_{e,t}^m \leq R_{e,t}(\mathbf{p}). \end{array} \quad (21)$$

where $\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}$ are the optimization variables.

Power Control subproblem We assume feasible network variables $(\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in C_c$. We need to solve the optimization problem

$$\begin{array}{ll} \text{minimize} & \Phi(\mathbf{p}, \mathbf{f}, \mathbf{b}) \\ \text{subject to} & \mathbf{p} \in C_p, \\ & \mathbf{p} \succeq \mathbf{J}(\mathbf{p}, \mathbf{c}). \end{array} \quad (22)$$

where $\mathbf{p}$ are the optimization variables.

As mentioned above, both of these subproblems are convex. The block co-ordinate descent algorithm is now described in Algorithm 1 below.

Algorithm 1: Block Co-ordinate Descent Algorithm

1. Input: All parameters for problem (16).
2. Choose $\hat{\mathbf{p}}^{(0)} \in C_p$ so that problem (21) is feasible;
3. Choose $(\hat{\mathbf{b}}^{(0)}, \hat{\mathbf{c}}^{(0)}, \hat{\mathbf{d}}^{(0)}, \hat{\mathbf{f}}^{(0)})$ arbitrarily
4. $i := 0$
5. `While` stopping criterion for $(\hat{\mathbf{b}}^{(i)}, \hat{\mathbf{c}}^{(i)}, \hat{\mathbf{d}}^{(i)}, \hat{\mathbf{f}}^{(i)}, \hat{\mathbf{p}}^{(i)})$ not fulfilled
   (a) Set $\hat{\mathbf{p}} := \hat{\mathbf{p}}^{(i)}$ and solve problem (21).
   (b) Denote the result by $(\hat{\mathbf{b}}^{(i+1)}, \hat{\mathbf{c}}^{(i+1)}, \hat{\mathbf{d}}^{(i+1)}, \hat{\mathbf{f}}^{(i+1)})$.
   (c) Set $(\hat{\mathbf{b}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}) := (\hat{\mathbf{b}}^{(i+1)}, \hat{\mathbf{c}}^{(i+1)}, \hat{\mathbf{d}}^{(i+1)}, \hat{\mathbf{f}}^{(i+1)})$ and solve problem (22). Denote the result by $\hat{\mathbf{p}}^{(i+1)}$.
   (d) $i := i + 1$.
6. Output: $(\hat{\mathbf{b}}^{(i)}, \hat{\mathbf{c}}^{(i)}, \hat{\mathbf{d}}^{(i)}, \hat{\mathbf{f}}^{(i)}, \hat{\mathbf{p}}^{(i)})$

Since (16) is a nonconvex problem, one cannot expect this algorithm to converge to a globally optimal solution in general. As the following results show, the algorithm stops within two complete iterations, and in the zero interference case, the coupling constraints become convex and the algorithm converges to a globally optimal solution under certain starting conditions.

Lemma 2. *After one solution of each of the subproblems (21) and (22), the block co-ordinate algorithm terminates at an iteration point at which all coupling constraints are active.*

Proof. We prove this constructively, working through the algorithm from an arbitrary starting point: Let $\mathbf{p}^0$ be our initial choice of $\mathbf{p}$, and $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ the solution to (21) with $\mathbf{p}$ fixed at $\mathbf{p}^0$.

Let $\mathbf{p}^1$ be the solution of (22) with the routing variables fixed at $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$.

We know, from the proof of Thm 2 that $\sum_{m \in M}(1+\kappa_m)(f^m_{e,t})^1 = \Psi_{e,t}(\mathbf{p}^1) \; \forall \, e \in E, t \in T$.

Consider now subproblem (21) with $\mathbf{p}$ fixed at $\mathbf{p}^1$, particularly the coupling constraints $p^1_{e,t} \geq J_{e,t}(\mathbf{f}, \mathbf{p}^1)$, which we rearrange back to $\sum_{m \in M}(1 + \kappa_m) f^m_{e,t} = R_{e,t}(\mathbf{p}^1) \; \forall \, e \in E, t \in T$.

We already know we have a feasible point $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ which uses all available capacity on all edges, and in solving the network flow subproblem we are seeking another feasible point which uses **less** capacity on some edges than $\mathbf{f}^1$, but never any **more** capacity on any edges than $\mathbf{f}^1$. Clearly no such point exists and so the feasible set for this subproblem is a singleton, namely $\{(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)\}$, so $(\mathbf{b}^2, \mathbf{c}^2, \mathbf{d}^2, \mathbf{f}^2)$, the solution of this subproblem must be $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ and therefore $\mathbf{p}^2 = \mathbf{p}^1$ and so on.

By definition, the optimal function value of the power control subproblem is $\Phi(\mathbf{b}^1, \mathbf{f}^1, \mathbf{p}^1)$, and the optimal function value of the network flow subproblem is $\Phi(\mathbf{b}^2, \mathbf{f}^2, \mathbf{J}(\mathbf{f}^2, \mathbf{p}^1)) = \Phi(\mathbf{b}^1, \mathbf{f}^1, \mathbf{J}(\mathbf{f}^1, \mathbf{p}^1))$.

Since $\sum_{m\in M}(f_{e,t}^m)^1 = R_{e,t}(\mathbf{p}^1)\ \forall\ e \in E, t \in T$, we see that $\mathbf{J}(\mathbf{f}^1, \mathbf{p}^1) = \mathbf{p}^1$, and so the optimal function values to the two subproblems are equal for all but SP1($\mathbf{p}^0$).

Note that Lemma 2 does not say a great deal about the quality of solutions provided by the algorithm, other than that they lie on the surface where all coupling constraints are active. At least this is a sensible place to be since we know all solutions lie on this surface also.

Lemma 3. *Suppose we wish to solve optimization problem, (16), using the coordinate descent algorithm 1. Assume the network suffers from no interference, that is*

$$R_{e,t}(\mathbf{p}) = B\tau \log_2(1 + \frac{a_{e,e}}{\sigma^2} p_{e,t}).$$

Assume further that at $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$*, the minimizer of the network flow subproblem, no coupling constraints are active, that is*

$$\sum_{m\in M} (1 + \kappa_m)(f_{e,t}^m)^1 < R_{e,t}(\mathbf{p}^0) \forall e \in E, t \in T.$$

Then, setting $\mathbf{p}^1$ *as the global minimizer of the power control subproblem applied to* $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$*,* $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^1)$ *is a global minimizer of the original optimization problem (16).*

Proof. Note that, in the zero interference case, power variables no longer appear in the objective function of problem formulation(19) since

$$J_{e,t}(\mathbf{f}, \mathbf{p}) = \frac{\sigma^2}{a_{e,e}}(2^{\frac{1}{B\tau}\sum_{m\in M}(1+\kappa_m)f_{e,t}^m} - 1) = J_{e,t}(\mathbf{f}).$$

The network flow subproblem is therefore:

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t. } (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \\ &\qquad p_{e,t}^0 \geq J_{e,t}(\mathbf{f}) \end{aligned}$$

This is a convex optimization problem, and so $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ is a global minimizer for the subproblem. Now, by assumption $p_{e,t}^0 > J_{e,t}(\mathbf{f}^1)\ \forall e \in E, t \in T$, and so $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ is also a global minimizer for the same problem without any coupling constraints:

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t. } (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \end{aligned}$$

By extension, for any $\mathbf{p} \in \mathbf{C}_{\mathbf{p}}$, $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p})$ is a global minimizer of:

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t. } (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \\ &\qquad \mathbf{p} \in \mathbf{C}_{\mathbf{p}} \end{aligned}$$

Adding additional constraints cannot decrease the optimal objective function value for a problem, so if $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p})$ remains feasible for a problem with additional constraints, it will be a global minimizer for that problem as well.

Now, since we assumed $p^0_{e,t} > J_{e,t}(\mathbf{f}^1) \forall\; e \in E, t \in T$ in the statement of the lemma, $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^0)$ is clearly feasible for

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t.}\;\; (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \\ &\qquad \mathbf{p} \in \mathbf{C}_{\mathbf{p}}, \\ &\qquad \textstyle\sum_{m \in M} f^m_{e,t} \leq B\tau \log_2(1 + a_{e,e} p_{e,t}) \;\forall e \in E, t \in T. \end{aligned}$$

$(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^0)$ is therefore a global minimzer for this problem, which is exactly formulation (19) of the original problem. Solving the power control subproblem applied to $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ we obtain $\mathbf{p}^1$ for which all coupling constraints are active. $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^1)$ is thus a global minimizer for (19) for which all coupling constraints are active, and therefore a global minimizer of the original problem (16).

Lemma 3 provides us with a simple test to perform on the solution of the network flow problem applied to $(\mathbf{p}^0)$ to tell us if we will reach a global minimizer. The following theorem provides a condition on the choice of $\mathbf{p}^0$ to guarantee convergence to the global minimizer in the zero interference case.

Theorem 3. *Assume* $\mathbf{C}_{\mathbf{p}}$ *contains only edge power bounds and not node power bounds and that the networks suffers no interference. Then the co-ordinate descent algorithm terminates at the global minimizer of problem (16) if we choose* $p^0_{e,t} = P^{\max}_e \;\forall e \in E, t \in T$.

Proof. Consider $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$, the minimizer of the network flow subproblem applied to $(\mathbf{p}^0)$:

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t.}\;\; (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \\ &\qquad P^{\max}_e \geq J_{e,t}(\mathbf{f}) \;\forall e \in E, t \in T. \end{aligned} \tag{23}$$

Since this problem is convex, $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ is a global minimizer for this problem, and therefore $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^0)$ is a global minimizer for the problem:

$$\begin{aligned} &\min \Phi(\mathbf{b}, \mathbf{f}, \mathbf{J}(\mathbf{f})) \\ &\text{s.t.}\;\; (\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{f}) \in \mathbf{C}_{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{f}} \\ &\qquad 0 \leq p_{e,t} \leq P^{\max}_e \;\forall e \in E, t \in T, \\ &\qquad p_{e,t} \geq J_{e,t}(\mathbf{f}) \;\forall e \in E, t \in T. \end{aligned} \tag{24}$$

since the feasible region for (24) is a subset of the feasible region for (23). Solving the power control subproblem applied to $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1)$ we attain $\mathbf{p}^1$ for which all coupling constraints are active. $(\mathbf{b}^1, \mathbf{c}^1, \mathbf{d}^1, \mathbf{f}^1, \mathbf{p}^1)$ is thus a global minimizer of (19) for which all coupling constraints are active and is therefore a global minimizer of (16).

The greatest difficulty with solving this optimization problem algorithmically in cases where we have cross channel interference is the non-convex coupling constraint between the edge flows and the transmit powers. The standard way around this problem is to apply an approximation and change of variables to convert these constraints to convex constraints [14], [21]. This convexified problem is then solved using a dual decomposition algorithm. One drawback of this approach is that the approximation is often a very bad approximation, and the dual algorithm converges at a sublinear rate. Without convexifying the problem, there is often a duality gap making it difficult or impossible to get optimal primal variables from optimal dual variables.

In networks with interference, we can still expect the algorithm to considerably outperform the dual approach in terms of speed, since at every iteration of the dual decomposition algorithm, the solutions to two subproblems of similar difficulty to the subproblems in the co-ordinate descent algorithm are required. In the co-ordinate descent algorithm, these subproblems need only be solved once each, whereas in the dual decomposition subgradient approach, each subproblem was solved tens or hundreds of times during the course of the algorithm. In the next section we will present numerical results, backing up the theoretical results shown in this chapter as well as showing the value of the co-ordinate descent algorithm, even if only treated as a heuristic with no guarantee of convergence to a local minimizer.

6 Simulation Results

In this section, we present some numerical results of the co-ordinate descent algorithm as applied to a multicast transmission in a wireless mesh backhaul network. The network under consideration is a typical cellular network with hexagonal cell structure. The cells are arranged around a center cell by rings and a node is located in the center of a hexagon.This models the situation where, to save infrastructure expenses like laying cable or fiber to each node in a network, we try to extend the range of given source node (center node) by intermediate nodes being wireless connected. The source node has wired backhaul connection only, while all other nodes have no wired backhaul connection and can only communicate with the wireless mesh backhaul via the source node. We require that wireless links can only be formed between nodes in adjacent rings. This means, (1) a node can not transmit to any node that is more than one ring away, and, (2) intra-ring communication is not allowed so that nodes belonging to the same ring have no wireless link established. We consider the case where the first ring consists of 3 nodes, the second ring 5 nodes, and the third ring consists of the three multicast destination nodes. The objective function we assume is to minimize total transmitted power with $\Phi(\mathbf{p}) = \sum_{e\in E,t\in T} p_{e,t}$.

Since both subproblems have been shown to have unique solutions, we did not deem it important to solve them using methods one would practically use in wireless settings. We want to get a feel for the algorithms overall performance and were not concerned with implementation at subproblem level. For this reason, both subproblems where solved with state-of-the-art solvers, namely the e04mf dense linear solver from NAG, and it's sparse nonlinear counterpart, e04ug [22].

We initialize all starting powers to maximum, since this is known to lead to optimal solutions in the zero interference case. In some high interference cases this means the starting point is infeasible and so the algorithm fails.

In line with Theorem 3, it was found that for all problem instances tested, the algorithm terminated after one solution of each subproblem at a point where all coupling constraints were active.

Supporting Lemma 3 and Theorem 3, we tested both networks in the zero interference case over a range of values of bandwidth, timeslot length, number of available timeslots, message size and background noise, using both Network Coding and simplistic routing, and found that in each case the block co-ordinate descent algorithm returned a global minimizer.

The main results of interest came when looking at problem instances with interference. In many instances, the algorithm finds feasible solutions that are close to optimal.

In figure 2 on the next page we investigate the backhaul network where we have used realistic parameter values and gain matrices. We compare the results found by our algorithm to the best found results of a strong black box solver started from multiple starting points, namely the NAG solver `e04ug` [22]. We see that for small messages, the co-ordinate descent algorithm returns solutions near global minimizers. As the message size increases, the solutions become suboptimal, and for messages too large, our choice of $\mathbf{p}^0$ results in an infeasible network flow subproblem. We also compare results using network coding to model the multicast, to the naive approach of treating a multicast as multiple independent unicasts. We see that using the network coding model, the algorithm is able to handle larger messages before the starting point becomes infeasible. We also see that, as expected, in solutions from the black-box and from the co-ordinate descent algorithm, the total transmit power required when using the network coding model is significantly reduced compared with the naive approach of treating the multicast as separate unicasts.

7 Conclusions

We have designed an optimization problem for the transmission of unicast and multicast messages through a time-slotted multipath, multihop wireless network. Through the idea of network coding we were able to incorporate the multicast transmissions in a simple way, resulting in large performance gains when compared to a naive approach whilst avoiding the complex combinatorial optimization problem associated with a tree finding approach. The optimization problem lends itself to being solved by a block co-ordinate descent approach, and in the zero interference case, we are able to guarantee convergence to a global optimum. Although this method does not guarantee to find global optima in cases where we consider cross channel interference, in many cases it provides feasible, near optimal solutions.

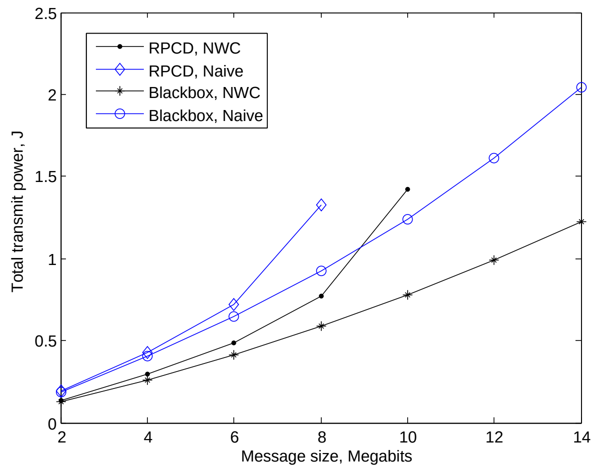

Fig. 2. *Total multicast transmit energy for varying message sizes, with inter-cell interference only in the backhaul network over 20 timeslots, comparing the co-ordinate descent algorithm to a blackbox solver.*

References

1. Viswanathan, H. and Mukherjee, S. Throughput-range tradeoff of wireless mesh backhaul networks, IEEE Journal on Selected Areas in Communications **24** (2006) 593602
2. The IEEE 802.16 Working Group on Broadband Wireless Access Standards, `http://www.wirelessman.org/`.
3. Cruz, R. L. and Santhanam, A. V. Optimal routing, link scheduling and power control in multi-hop wireless networks, Proc. of INFOCOM 2003, San Francisco, (2003) 702711
4. Li, Y. and Ephremides, A. Joint scheduling, power control, and routing algorithm for ad-hoc wireless networks, Proc. of the 38th Annual Hawaii International Conference on System Sciences, (2005) 322
5. Xiao, L., Johansson, M. and Boyd, S. P. Simultaneous routing and resource allocation via dual decomposition, IEEE Transactions on Communications **52** no. 7 (2004) 11361144
6. Winter, P. Steiner Problem in Networks: A Survey, Networks **17** (1987) 129-167
7. Sanders, P., Egner, S. and Tolhuizen, L. Polynomial time algorithms for network information flow, Proc. 15th ACM Symposium on Parallel Algorithms and Architectures (2003) 286-294
8. Fragouli, C. and Soljanin, E. Decentralized Network Coding, Information Theory Workshop (2004) 310-314

9. Ho, T., Koetter, R., Medard, M., Karger, D., and Effros, M. The benefits of Coding over routing in a randomized setting, Proceedings of ISIT (2003) 442
10. Ho, T., Medard, M., Shi, J., Effros, M., and Karger, D. On randomized network coding, Proc. 41st Annual Allerton Conference on Communication, Control, and Computing (2003)
11. Wu, Y., Chou, P. A., and Jain, K. A comparison of network coding and tree packing, Proceedings of ISIT (2004) 143
12. Koetter, R. and Medard, M. An algebraic approach to network coding, IEEE/ACM Transactions on Networking **11** (2003) 782-795
13. Yuan, J., Li, Z., Lu, W., and Li, B. A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks, IEEE Journal on Selected Areas in Communication **24** (2006) 2092-2103
14. Johansson, M., Xiao, L., and Boyd, S. Simultaneous Routing and Power Allocation in CDMA Wireless Data Networks, Proc. IEEE International Conference on Communications (2003) 51-55
15. Maurer, S. B. and Ralston, A. Discrete Algorithmic Mathematics, Addison-Wesley, Reading, Mass, USA, (1991)
16. Li, S., Yeung, R. and Cai, N. Linear Network Coding, IEEE Transactions on Information Theory **49** (2003) 371-381
17. Ahlswede, R., Cai, N., Li, S. R., and Yeung, R. W. Network information flow, IEEE Transactions on Information Theory **46** (2000) 1204-1216
18. Fragouli, C., Katabi, D., Markopoulou, A., Medard, M., and Rahul, H. Wireless Network Coding: Opportunities and Challenges, Proc. MILCOM (2007)
19. Boyd, S. and Vandenberghe, L. Convex Optimization, Cambridge University Press, Cambridge, UK, (2004)
20. Eyuboglu, M. V. and Forney Jr., G. D. Trellis precoding: combined coding, precoding and shaping for intersymbol interference channels, IEEE Transactions on Information Theory, **38** (1992) 301-314
21. Johansson, M. and Xiao, L. Scheduling, Routing and Power Allocation for Fairness in Wireless Networks, Proc. IEEE 59th Vehicular Technology Conference (2004) 1355-1360
22. National Algorithm Group, Documentation for the NAG MATLAB Toolbox, `http://www.nag.co.uk/`