Cross-Layer Optimization for MIMO-Based Wireless Ad Hoc Networks: Routing, Power Allocation, and Bandwidth Allocation

Jia Liu, *Student Member, IEEE,* Y. Thomas Hou, *Senior Member, IEEE,* Yi Shi, *Member, IEEE,* and Hanif D. Sherali

***Abstract*—MIMO-based communications systems have great potential to improve network capacity for wireless ad hoc networks. Due to unique physical layer characteristics associated with MIMO, network performance is tightly coupled with mechanisms at physical, link, and routing layers. So far, research on MIMO-based wireless ad hoc networks is still in its infancy and few results are available. In this paper, we consider the problem of jointly optimizing power and bandwidth allocation at each node and multi-hop/multi-path routing in a MIMO-based wireless ad hoc network. We develop a solution procedure to this cross-layer optimization problem and use simulations to validate the efficacy of this solution.**

***Index Terms*—Multiple-input multiple-output (MIMO), multi-hop ad hoc network, cross-layer optimization.**

I. Introduction

WIRELESS ad hoc network has been a focal research area in the research community for some years. A critical factor affecting the future prospect of such networks for practical deployment is *network capacity*. Multiple-input multiple-output (MIMO) system, which employs multiple antennas, is shown to be capable of increasing channel capacity substantially than conventional communication systems without the cost of additional spectrum [1]–[3]. As a result, MIMO has been recognized as an enabling technology for high capacity wireless ad hoc networks.

However, compared to the research on single-user MIMO, for which many results are available (see [4], [5] and references therein), research on multiuser MIMO systems is still in its infancy and many fundamental problems, particularly for multi-hop ad hoc networks, remain unsolved. Employing MIMO in an ad hoc network is far from trivial. As discussed by Winters [6], a MIMO-based ad hoc network with each node equipped with M antennas does not necessarily mean that the network capacity is also increased by M-fold. The potential network capacity gain with the use of MIMO depends on the coordinated mechanisms at the physical, link, and network layers. An improperly designed algorithm could diminish any potential capacity gain from MIMO. As a result, joint optimization across multiple layers is not only desirable, but also necessary.

In this paper, we investigate cross-layer optimization for MIMO-based ad hoc networks. The goal is to support a set of user communication sessions such that some network utility is maximized. However, to achieve high capacity for such networks, many challenging problems must be addressed. One problem is how to determine optimal power allocation at each transmitting node, optimal bandwidth allocation for each transmission, and optimal flow routing for the network. This problem is considerably more challenging than that for conventional single antenna-based wireless ad hoc networks. This is because, compared to the simple scalar channels in the single antenna case, power allocations are now performed over complex matrix channels. Also, compared to single-user MIMO systems, power allocation for multiple outgoing links at a node has to be jointly considered.

For this challenging cross-layer optimization problem, we show that it has some special structure which allows us to decompose the original problem into a set of subproblems in its dual domain. Specifically, our solution procedure first decouples the dual problem into a network layer subproblem and a link-physical layer subproblem. For the link-physical layer subproblem (corresponding to multi-antenna power allocation and bandwidth allocation), which is the most difficult part in the dual problem, we employ techniques in matrix differential calculus and develop an algorithm based on gradient projection (GP). By exploiting the piece-wise quadratic structure of the projection subproblem, our proposed GP method enjoys polynomial-time complexity.

Then, for the dual problem, we propose two strategies, i.e., a cutting-plane method based on outer-linearization and the subgradient-based scheme. Our proposed cutting-plane method allows an easy recovery of primal feasible and optimal solutions.

Finally, based on the subgradient-based approach, we design a distributed algorithm that achieves the same optimal solution as that of the centralized algorithm. We show that the excess link capacity of each link can be used for message exchange in our distributed algorithm.

The remainder of this paper is organized as follows. In Section II, we present the network model and problem formu-

Manuscript received July 1, 2007; revised December 1, 2007. This paper was presented in part at IEEE WCNC'07, Hong Kong, China, March 2007.

Jia Liu, Y. Thomas Hou, and Yi Shi are with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061 USA (e-mail: {kevinlau, thou, yshi}@vt.edu).

Hanif D. Sherali is with the Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061 USA (e-mail: hanifs@vt.edu).

Digital Object Identifier 10.1109/JSAC.2008.080808.

0733-8716/08/$25.00 © 2008 IEEE

lation. Section III presents our decomposition framework and key subproblems to be solved at each layer. In Section IV, we focus on the challenging physical-link layer subproblem and the design of gradient projection method. In Section V, we propose the cutting-plane and the subgradient methods for solving the dual problem, respectively. Section VI discusses the design of a distributed algorithm. Section VII presents numerical results. Section VIII reviews related work. Section IX concludes this paper.

II. Network Model

We consider an FDMA MIMO-based ad hoc network, where each node has been assigned non-overlapping (possibly reused) frequency bands for its incoming and outgoing links so that nodes can simultaneously transmit and receive, and cause no interference among each other. There is a vast amount of literature on how to perform channel assignments and its discussion is beyond the scope of this paper. In this paper, we focus on how to jointly optimize routing at the network layer, bandwidth allocation at the link layer, and power allocation at the physical layer.

We first introduce notation for matrices, vectors, and complex scalars in this paper. We use boldface to denote matrices and vectors. For a matrix $\mathbf{A}$, $\mathbf{A}^\dagger$ denotes the conjugate transpose. $\mathrm{Tr}\{\mathbf{A}\}$ denotes the trace of $\mathbf{A}$. $\mathrm{Diag}\{\mathbf{A}_1,\ldots,\mathbf{A}_n\}$ represents the block diagonal matrix with matrices $\mathbf{A}_1,\ldots,\mathbf{A}_n$ on its main diagonal. We denote $\mathbf{I}$ the identity matrix with dimension determined from the context. $\mathbf{A}\succeq 0$ represents that $\mathbf{A}$ is Hermitian and positive semidefinite (PSD). $\mathbf{1}$ and $\mathbf{0}$ denote vectors whose elements are all ones and zeros, respectively, and their dimensions are determined from the context. $(\mathbf{v})_m$ represents the m-th entry of vector $\mathbf{v}$. For a real vector $\mathbf{v}$ and a real matrix $\mathbf{A}$, $\mathbf{v}\geq\mathbf{0}$ and $\mathbf{A}\geq\mathbf{0}$ mean that all entries in $\mathbf{v}$ and $\mathbf{A}$ are nonnegative, respectively. We let $\mathbf{e}_i$ be the unit column vector where the i-th entry is 1 and all the other entries are 0. The dimension of $\mathbf{e}_i$ is determined from the context as well. The operator "$\langle\cdot,\cdot\rangle$" represents vector or matrix inner product operation.

A. MIMO Power Allocation

Let the matrix $\mathbf{H}_l\in\mathbb{C}^{n_r\times n_t}$ represent the wireless channel gain matrix from the transmitting node to the receiving node of link l, where n_t and n_r are the numbers of transmitting and receiving antenna elements at each node, respectively. Suppose that $\mathbf{H}_l$ is known at the transmitting node of link l. Although wireless channels in reality are time-varying, we consider a "constant" channel model in this paper, i.e., $\mathbf{H}_l$'s coherence time is larger than the transmission period we consider. This simplification is of much interest for it enables to find the ergodic capacity for block-wise fading channels [5]. The received complex base-band signal vector for MIMO link l with n_t transmitting antennas and n_r receiving antennas in a Gaussian channel is given by

$$\mathbf{r}_l=\sqrt{\rho_l}\mathbf{H}_l\mathbf{t}_l+\mathbf{n}_l, \tag{1}$$

where $\mathbf{r}_l$ and $\mathbf{t}_l$ represent the received and transmitted signal vectors, $\mathbf{n}_l$ is the normalized additive white Gaussian noise (AWGN) vector, ρ_j captures path-loss effect.

Let matrix $\mathbf{Q}_l$ represent the covariance matrix of a zero-mean Gaussian input symbol vector $\mathbf{t}_l$ at link l, i.e., $\mathbf{Q}_l=\mathbb{E}\left\{\mathbf{t}_l\cdot\mathbf{t}_l^\dagger\right\}$. The definition of $\mathbf{Q}_l$ implies that it is Hermitian and PSD. Physically, $\mathbf{Q}_l$ represents the power allocation for different antennas on link l's transmitting node. In this paper, we use matrix $\mathbf{Q}\triangleq\left[\begin{array}{cccc}\mathbf{Q}_1 & \mathbf{Q}_2 & \ldots & \mathbf{Q}_L\end{array}\right]\in\mathbb{C}^{n_t\times(n_t\cdot L)}$ to denote the collection of all input covariance matrices in the network.

We define $\mathcal{O}(n)$ and $\mathcal{I}(n)$ as the sets of links that are outgoing from and incoming to node n, respectively. At the physical layer, since the total transmit power of each node is subject to a maximum power constraint, we have that $\sum_{l\in\mathcal{O}(n)}\mathrm{Tr}\{\mathbf{Q}_l\}\leq P_{\max}^{(n)}$, $1\leq n\leq N$, where $P_{\max}^{(n)}$ represents the maximum transmit power of node n.

B. Link Capacity and Bandwidth Allocation

The capacity of a MIMO link l in a AWGN channel can be computed as

$$\Phi_l(W_l,\mathbf{Q}_l)\triangleq W_l\log_2\det\left(\mathbf{I}+\rho_l\mathbf{H}_l\mathbf{Q}_l\mathbf{H}_l^\dagger\right), \tag{2}$$

where W_l represents the communication bandwidth of link l. It can be readily verified that $\Phi_l(W_l,\mathbf{Q}_l)$ is a monotone increasing concave function for $W_l>0$ and $\mathbf{Q}_l\succeq 0$. It can be seen that the values of W_l and $\mathbf{Q}_l$, i.e., the allocation of bandwidth and power on link l, directly affect the capacity of link l. As a result, bandwidth allocation variables W_l together with power allocation variables $\mathbf{Q}_l$ play an important role in our cross-layer optimization problem.

Since the total bandwidth of all outgoing links at a node, say node n, cannot exceed its assigned bandwidth, denoted by B_n, we have $\sum_{l\in\mathcal{O}(n)}W_l\leq B_n$, $1\leq n\leq N$. We denote matrix $\mathbf{W}=\left[\begin{array}{cccc}W_1 & W_2 & \ldots & W_L\end{array}\right]^T\in\mathbb{R}^{L\times 1}$ the collection of all bandwidth variables.

C. Routing

In this paper, the topology of a MIMO-based wireless ad hoc network is represented by a directed graph, denoted by $\mathcal{G}=\{\mathcal{N},\mathcal{L}\}$, where $\mathcal{N}$ and $\mathcal{L}$ are the set of nodes and all possible MIMO links, respectively. We assume that $\mathcal{G}$ is always connected. Suppose that the cardinalities of the sets $\mathcal{N}$ and $\mathcal{L}$ are $|\mathcal{N}|=N$ and $|\mathcal{L}|=L$, respectively. In our model, a link (the line segment defined by a pair of nodes) exists if the link is shorter than or equal to the maximum transmission range R_T, i.e., $\mathcal{L}=\{(i,j):D_{ij}\leq R_T, i,j\in\mathcal{N}, i\neq j\}$, where D_{ij} represents the distance between node i and node j. R_T can be determined by a node's maximum transmission power.

It is worth pointing out that link does not physically exist in wireless networks and any node pair may be treated as a possible link. However, in practice, some nodes are separated so far away from each other that the channel gain is very low. In such a case, even if the transmitting node allocates all its power to this link, the received SNR remains so low that the link capacity is practically zero (i.e., impossible to decode due to the received low SNR). As a result, such node pairs can be safely removed from the set of possible links without affecting the accuracy of the final optimal solution. This is

because allocating power to "bad" links leads to virtually no capacity gain in these links while significantly penalizing the capacities of other "good" links, which obviously could not be optimal.

The network topology of $\mathcal{G}$ can be represented by a *node-arc incidence matrix* (NAIM) [7] $\mathbf{A} \in \mathbb{R}^{N\times L}$, whose entry a_{nl} associating with node n and arc (link) l is defined as

$$a_{nl} = \begin{cases} 1 & \text{if } n \text{ is the transmitting node of arc } l \\ -1 & \text{if } n \text{ is the receiving node of arc } l \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We use a multi-commodity flow model for the routing of data packets across a wireless ad hoc network. That is, in a wireless ad hoc network, source nodes may send different data to their intended destination nodes through *multi-path* and *multi-hop* routing. We assume that the flow conservation law at each node is satisfied, i.e., the network is a flow-balanced system. Suppose that there is a total of F sessions in the network. We denote the source and destination nodes of session f, $1 \le f \le F$ as $\mathrm{src}(f)$ and $\mathrm{dst}(f)$, respectively. For session f, we denote $\mathbf{s}_f \in \mathbb{R}^N$ the *source-sink vector*, whose entries, other than at the positions of $\mathrm{src}(f)$ and $\mathrm{dst}(f)$, are all zeros. In addition, from the flow conservation law, we must have $(\mathbf{s}_f)_{\mathrm{src}(f)} = -(\mathbf{s}_f)_{\mathrm{dst}(f)}$. Without loss of generality, we let $(\mathbf{s}_f)_{\mathrm{src}(f)} \ge 0$ and simply denote it as a scalar s_f. Therefore, we can further write the source-sink vector of session f in the form of

$$\mathbf{s}_f = s_f \left[\begin{array}{ccccccc} \cdots & 1 & \cdots & -1 & \cdots \end{array} \right]^T, \tag{4}$$

where the dots represent zeros, and 1 and -1 are in the positions of $\mathrm{src}(f)$ and $\mathrm{dst}(f)$, respectively. Note that for the source-sink vector of session f, 1 does not necessarily appear before -1 as in (4), which is only for illustrative purpose. Using the notation "$=_{x,y}$" to represent the component-wise equality of a vector except at the x-th and the y-th entries, we have $\mathbf{s}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0}$. In addition, denoting matrix $\mathbf{S} \triangleq \left[\begin{array}{cccc} \mathbf{s}_1 & \mathbf{s}_2 & \ldots & \mathbf{s}_F \end{array} \right] \in \mathbb{R}^{N\times F}$ the collection of all source-sink vectors $\mathbf{s}_f$, we have

$$\begin{aligned} \mathbf{S}\mathbf{e}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0}, &\quad 1 \le f \le F, \\ \langle \mathbf{1}, \mathbf{S}\mathbf{e}_f \rangle = 0, &\quad 1 \le f \le F, \\ (\mathbf{S}\mathbf{e}_f)_{\mathrm{src}(f)} = s_f, &\quad 1 \le f \le F, \end{aligned}$$

where $\mathbf{e}_f$ is the f-th unit column vector.

For link l, we let $x_l^{(f)} \ge 0$ be the amount of flow of session f on link l. We define $\mathbf{x}^{(f)} \in \mathbb{R}^L$ as the *flow vector* for session f. At each node n, components of the flow vector and source-sink vector for the same session satisfy the following flow conservation law:

$$\sum_{l\in\mathcal{O}(n)} x_l^{(f)} - \sum_{l\in\mathcal{I}(n)} x_l^{(f)} = (\mathbf{s}_f)_n, \quad 1 \le n \le N,\ 1 \le f \le F.$$

With NAIM, the flow conservation law for the entire network can be written as $\mathbf{A}\mathbf{x}^{(f)} = \mathbf{s}_f,\ 1 \le f \le F$. Denote matrix $\mathbf{X} \triangleq \left[\begin{array}{cccc} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \ldots & \mathbf{x}^{(F)} \end{array} \right] \in \mathbb{R}^{L\times F}$ the collection of all flow vectors $\mathbf{x}^{(f)}$. Then, the flow conservation law can be written as

$$\mathbf{A}\mathbf{X} = \mathbf{S}.$$

Since the network flow traversing a link cannot exceed the link's capacity limit, we have

$$\sum_{f=1}^{F} x_l^{(f)} \le \Phi_l(W_l, \mathbf{Q}_l), \quad 1 \le l \le L,$$

Using matrix-vector notation, this can be written compactly as

$$\langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l \rangle \le \Phi_l(W_l, \mathbf{Q}_l), \quad 1 \le l \le L.$$

Note that in a wireless network, link capacity $\Phi_l(W_l, \mathbf{Q}_l)$ is not fixed and varies as the power allocation and bandwidth allocation change.

In summary, the multicommodity network flow model imposes the following group of constraints:

$$\begin{cases} \mathbf{A}\mathbf{X} = \mathbf{S}, \\ \mathbf{X} \ge \mathbf{0}, \\ \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l \rangle \le \Phi_l(W_l, \mathbf{Q}_l), \quad \forall l, \end{cases}$$

where $\mathbf{S}$ satisfies $\mathbf{S}\mathbf{e}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0}$, $\langle \mathbf{1}, \mathbf{S}\mathbf{e}_f \rangle = 0$, and $(\mathbf{S}\mathbf{e}_f)_{\mathrm{src}(f)} = s_f$, for $f = 1, 2, \ldots, F$.

D. Problem Formulation

We adopt the proportional fairness utility function, i.e., $\ln(s_f)$ for session f [8]. The objective is to maximize the sum of utilities of all sessions. Putting together the physical layer constraints in Subsection II-A, the link layer constraints in Subsection II-B, and the network layer constraints in Subsection II-C, we have the following problem formulation.

$$\begin{array}{rll} \textbf{CRPBA}: \text{Maximize} & \sum_{f=1}^{F} \ln(s_f) & \\ \text{subject to} & \mathbf{A}\mathbf{X} = \mathbf{S} & \\ & \mathbf{X} \ge \mathbf{0} & \\ & \mathbf{S}\mathbf{e}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0} & \forall f \\ & \langle \mathbf{1}, \mathbf{S}\mathbf{e}_f \rangle = 0 & \forall f \\ & (\mathbf{S}\mathbf{e}_f)_{\mathrm{src}(f)} = s_f & \forall f \\ & \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l \rangle \le \Phi_l(W_l, \mathbf{Q}_l) & \forall l \\ & \sum_{l\in\mathcal{O}(n)} \mathrm{Tr}\{\mathbf{Q}_l\} \le P_{\max}^{(n)} & \forall n \\ & \sum_{l\in\mathcal{O}(n)} W_l \le B_n & \forall n \\ & \mathbf{Q}_l \succeq 0,\ W_l \ge 0 & \forall l \\ \text{Variables}: & \mathbf{S},\ \mathbf{X},\ \mathbf{Q}, \mathbf{W} & \end{array}$$

III. Problem Decomposition

We exploit the following special structure in CRPBA. The network layer variables and the link layer variables are coupled through the link capacity constraints $\langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l \rangle \le \Phi_l(W_l, \mathbf{Q}_l)$. Thus, we can employ Lagrangian dual decomposition to solve CRPBA efficiently. Generally, given a nonlinear programming problem, several different Lagrangian dual problems can be constructed depending on which constraints are associated with Lagrangian dual variables [9]. For CRPBA, we associate a Lagrangian multiplier u_l to link l's coupling constraint $\langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l \rangle \le \Phi_l(W_l, \mathbf{Q}_l)$. Let vector $\mathbf{u} \triangleq \left[\begin{array}{cccc} u_1 & u_2 & \ldots & u_L \end{array} \right]$ represent the collection of all dual variables. Hence, the Lagrangian can be written as

$$\Theta(\mathbf{u}) = \sup_{\mathbf{S},\mathbf{X},\mathbf{Q},\mathbf{W}} \{L(\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}, \mathbf{u}) | (\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}) \in \Gamma\},$$

where

$$L(\mathbf{S},\mathbf{X},\mathbf{Q},\mathbf{W},\mathbf{u}) = \sum_f \ln(s_f) + \sum_l u_l\left(\Phi_l(W_l,\mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l\rangle\right) \quad (5)$$

and Γ is defined as

$$\Gamma \triangleq \left\{ (\mathbf{S},\mathbf{X},\mathbf{Q},\mathbf{W}) \;\middle|\; \begin{array}{ll} \mathbf{AX}=\mathbf{S} & \\ \mathbf{X}\succeq\mathbf{0} & \\ \mathbf{Se}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0} & \forall f \\ \langle \mathbf{1}, \mathbf{Se}_f\rangle = 0 & \forall f \\ (\mathbf{Se}_f)_{\mathrm{src}(f)} = s_f & \forall f \\ \sum_{l\in\mathcal{O}(n)} \mathrm{Tr}\{\mathbf{Q}_l\} \le P_{\max}^{(n)} & \forall n \\ \mathbf{Q}_l \succeq 0, W_l \ge 0 & \forall l \\ \sum_{l\in\mathcal{O}(n)} W_l \le B_n & \forall n \end{array} \right\}$$

The Lagrangian dual problem of CRPBA can be written as:

$$\mathbf{D}^{\mathrm{CRPBA}}: \quad \begin{array}{ll} \text{Minimize} & \Theta(\mathbf{u}) \\ \text{subject to} & \mathbf{u}\ge\mathbf{0}. \end{array}$$

It is easy to recognize that, for any given Lagrangian multiplier $\mathbf{u}$, the Lagrangian can be separated into two terms:

$$\Theta(\mathbf{u}) = \Theta_{\mathrm{net}}(\mathbf{u}) + \Theta_{\mathrm{link-phy}}(\mathbf{u}),$$

where Θ_{net} and $\Theta_{\mathrm{link-phy}}$ are two subproblems respectively corresponding to the network layer and the link-physical layer:

$$\begin{array}{lll} \Theta_{\mathrm{net}}(\mathbf{u}) \triangleq & \text{Maximize} & \sum_f \ln(s_f) - \sum_l u_l\langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l\rangle \\ & \text{subject to} & \mathbf{AX}=\mathbf{S} \\ & & \mathbf{X}\succeq\mathbf{0} \\ & & \mathbf{Se}_f =_{\mathrm{src}(f),\mathrm{dst}(f)} \mathbf{0} \quad \forall f \\ & & \langle \mathbf{1}, \mathbf{Se}_f\rangle = 0 \quad \forall f \\ & & (\mathbf{Se}_f)_{\mathrm{src}(f)} = s_f \quad \forall f \\ & \text{Variables}: & \mathbf{S},\ \mathbf{X} \end{array}$$

$$\begin{array}{lll} \Theta_{\mathrm{link-phy}}(\mathbf{u}) \triangleq & \text{Maximize} & \sum_l u_l\Phi_l(W_l,\mathbf{Q}_l) \\ & \text{subject to} & \sum_{l\in\mathcal{O}(n)} \mathrm{Tr}\{\mathbf{Q}_l\} \le P_{\max}^{(n)} \quad \forall n \\ & & \sum_{l\in\mathcal{O}(n)} W_l \le B_n \quad \forall n \\ & & \mathbf{Q}_l \succeq 0, W_l \ge 0 \quad \forall l \\ & \text{Variables}: & \mathbf{Q},\ \mathbf{W} \end{array}$$

Then, the CRPBA Lagrangian dual problem can be transformed into the following master dual problem:

$$\mathbf{MD}^{\mathrm{CRPBA}}: \quad \begin{array}{ll} \text{Minimize} & \Theta_{\mathrm{net}}(\mathbf{u}) + \Theta_{\mathrm{link-phy}}(\mathbf{u}) \\ \text{subject to} & \mathbf{u}\ge\mathbf{0}. \end{array}$$

Now, the task of solving the decomposed Lagrangian dual problem boils down to solving the subproblems $\Theta_{\mathrm{net}}(\mathbf{u})$ and $\Theta_{\mathrm{link-phy}}(\mathbf{u})$, and the master problem $\mathbf{MD}^{\mathrm{CRPBA}}$. Note that in the network layer subproblem $\Theta_{\mathrm{net}}(\mathbf{u})$, the objective function is concave and all constraints are affine. Therefore, $\Theta_{\mathrm{net}}(\mathbf{u})$ can be readily solved by many polynomial-time convex programming methods. However, solving $\Theta_{\mathrm{link-phy}}(\mathbf{u})$ is not trivial because the objective function and constraints involve many complex matrices variables. Even though $\Theta_{\mathrm{link-phy}}(\mathbf{u})$ is a convex problem in nature, standard convex optimization methods without exploiting the special structure of $\Theta_{\mathrm{link-phy}}(\mathbf{u})$ are not efficient. In the following section, we propose a custom-designed method based on gradient projection to solve $\Theta_{\mathrm{link-phy}}(\mathbf{u})$.

IV. The Link-Physical Layer Subproblem

In this paper, we propose a gradient projection-based method (GP) to solve the link-physical layer subproblem. Gradient projection, originally proposed by Rosen [10], is a classical nonlinear programming method designed for solving constrained optimization problems. Its formal convergence proof, however, was not established until very recently in [9]. The framework of our proposed GP method is shown in Algorithm 1.

Algorithm 1 Gradient Projection Algorithm

Initialization:
Choose the initial conditions $\mathbf{W}^{(0)} = [W_1^{(0)}, W_2^{(0)}, \ldots, W_L^{(0)}]^T$, $\mathbf{Q}^{(0)} = [\mathbf{Q}_1^{(0)}, \mathbf{Q}_2^{(0)}, \ldots, \mathbf{Q}_L^{(0)}]^T$. Let $k=0$.

Main Loop:

1. Calculate the gradients $G_{W_l}^{(k)} = \nabla_{W_l}\Theta_{\mathrm{link-phy}}(\mathbf{u},\mathbf{W}^{(k)},\mathbf{Q}^{(k)})$ and $\mathbf{G}_{\mathbf{Q}_l}^{(k)} = \nabla_{\mathbf{Q}_l}\Theta_{\mathrm{link-phy}}(\mathbf{u},\mathbf{W}^{(k)},\mathbf{Q}^{(k)})$, for $l=1,2,\ldots,L$.
2. Choose an appropriate step size s_k. Let $W_l^{(k)'} = W_l^{(k)} + s_k G_{W_l}^{(k)}$, $\mathbf{Q}_l^{(k)'} = \mathbf{Q}_l^{(k)} + s_k\mathbf{G}_{\mathbf{Q}_l}^{(k)}$, for $l=1,2,\ldots,L$.
3. Let $[\bar{\mathbf{W}}_n^{(k)}, \bar{\mathbf{Q}}_n^{(k)}]^T$ be the projection of $[\mathbf{W}_n^{(k)'}, \mathbf{Q}_n^{(k)'}]^T$ onto $\Omega_+(n)$, where $\Omega_+(n) \triangleq \{(W_l,\mathbf{Q}_l) : l\in\mathcal{O}(n), W_l\ge 0, \mathbf{Q}_l\succeq 0, \sum_{l\in\mathcal{O}(n)} W_l \le B_n, \sum_{l\in\mathcal{O}(n)}\mathrm{Tr}\{\mathbf{Q}_l\} \le P_{\max}^{(n)}\}$.
4. Choose an appropriate step size α_k. Let $W_l^{(k+1)} = W_l^{(k)} + \alpha_k(\bar{W}_l^{(k)} - W_l^{(k)})$, $\mathbf{Q}_l^{(k+1)} = \mathbf{Q}_l^{(k)} + \alpha_k(\bar{\mathbf{Q}}_l^{(k)} - \mathbf{Q}_l(k))$, $l = 1,2,\ldots,L$.
5. Let $k\leftarrow k+1$. If $\|\mathbf{Q}_l^{(k)} - \mathbf{Q}_l^{(k-1)}\| < \epsilon$ and $|W_l^{(k)} - W_l^{(k-1)}| < \epsilon$, for $l=1,2,\ldots,L$, then stop; else go to Step 1.

Due to the complexity of the objective function, performing an exact line search is onerous as it calls for excessive evaluations of the objective function. Therefore, we adopt the "Armijo rule" inexact line search method [9], which still enjoys provable convergence.

The basic idea of Armijo rule is that at each step of line search, we sacrifice accuracy for efficiency as long as we have sufficient improvement. According to Armijo rule, we choose $s_k = 1$ and $\alpha_k = \beta^{m_k}$ (same as in [11]), where m_k is the first non-negative integer that satisfies

$$\Theta_{\mathrm{link-phy}}(\mathbf{Q}^{(k+1)}) - \Theta_{\mathrm{link-phy}}(\mathbf{Q}^{(k)}) \ge \sigma\beta^{m_k}\sum_{l=1}^{L}\mathrm{Tr}\left[\nabla_{\mathbf{Q}_l}\Theta_{\mathrm{link-phy}}(\mathbf{Q}^{(k)})^\dagger\left(\bar{\mathbf{Q}}_l^{(k)} - \mathbf{Q}_l^{(k)}\right)\right], \quad (6)$$

where $0<\beta<1$ and $0<\sigma<1$ are some fixed constants.

In order to make the GP algorithm work, we have to address two problem specific questions, i.e., during the k-th duration, how to compute the gradients and how to project $\mathbf{Q}_l^{(k)'}$ and $W_l^{(k)'}$ onto $\Omega_+(n) \triangleq \{(W_l,\mathbf{Q}_l) : l\in\mathcal{O}(n), W_l\ge 0, \mathbf{Q}_l\succeq 0, \sum_{l\in\mathcal{O}(n)} W_l \le B_n, \sum_{l\in\mathcal{O}(n)}\mathrm{Tr}\{\mathbf{Q}_l\} \le P_{\max}^{(n)}\}$.

A. Computing the Gradients

For the gradient with respect to W_l, it is not difficult to see that $G_{W_l} \triangleq \nabla_{W_l}\Theta_{\mathrm{link-phy}} = u_l\log_2\det(\mathbf{I} + \rho_l\mathbf{H}_l\mathbf{Q}_l\mathbf{H}_l^\dagger)$. The gradient $\mathbf{G}_{\mathbf{Q}_l} \triangleq \nabla_{\mathbf{Q}_l}\Theta_{\mathrm{link-phy}}(\mathbf{Q})$ depends on the partial derivative of $\Theta_{\mathrm{link-phy}}(\mathbf{Q})$ with respect to $\mathbf{Q}_l$. Before computing the partial derivative of $\Theta_{\mathrm{link-phy}}(\mathbf{Q})$, we need the following lemma [12].

Lemma 1: For matrices $\mathbf{B} \in \mathbb{C}^{p\times m}$, $\mathbf{X} \in \mathbb{C}^{m\times n}$, and $\mathbf{C} \in \mathbb{C}^{n\times p}$, if $(\mathbf{BXC})$ is invertible, then we have $\frac{\partial \det(\mathbf{BXC})}{\partial \mathbf{X}} = \det(\mathbf{BXC})\left[\mathbf{C}(\mathbf{BXC})^{-1}\mathbf{B}\right]^T$.

With Lemma 1, we have the following corollary.

Corollary 1: For matrices $\mathbf{A} \in \mathbb{C}^{p\times p}$, $\mathbf{B} \in \mathbb{C}^{p\times m}$, $rank(\mathbf{B}) = p$, $\mathbf{X} \in \mathbb{C}^{m\times n}$, $\mathbf{C} \in \mathbb{C}^{n\times p}$, and $rank(\mathbf{C}) = p$, if $(\mathbf{A} + \mathbf{BXC})$ is invertible, then we have $\frac{\partial}{\partial \mathbf{X}} \ln\det(\mathbf{A} + \mathbf{BXC}) = \left[\mathbf{C}(\mathbf{A} + \mathbf{BXC})^{-1}\mathbf{B}\right]^T$.

Proof: By chain rule, we have

$$\begin{aligned}\frac{\partial}{\partial \mathbf{X}} \ln\det(\mathbf{BXC}) &= \frac{\partial \ln\det(\mathbf{BXC})}{\partial \det(\mathbf{BXC})} \cdot \frac{\partial \det(\mathbf{BXC})}{\partial \mathbf{X}} \\ &= \frac{1}{\det(\mathbf{BXC})} \cdot \det(\mathbf{BXC})\left[\mathbf{C}(\mathbf{BXC})^{-1}\mathbf{B}\right]^T \\ &= \left[\mathbf{C}(\mathbf{BXC})^{-1}\mathbf{B}\right]^T. \end{aligned} \quad (7)$$

Since $rank(\mathbf{B}) = p$ and $rank(\mathbf{C}) = p$, we have that $\mathbf{B}$ and $\mathbf{C}$ have right and left inverses, respectively. Let $\mathbf{Y}$ be such that $\mathbf{BYC} = \mathbf{A} + \mathbf{BXC}$, i.e., $\mathbf{Y} = \mathbf{X} + \mathbf{B}_R\mathbf{A}\mathbf{C}_L$, where $\mathbf{B}_R$ and $\mathbf{C}_L$ are the right and left inverses of $\mathbf{B}$ and $\mathbf{C}$, respectively. It then follows that $\partial\mathbf{Y}/\partial\mathbf{X} = \mathbf{I}$. Thus, we have

$$\begin{aligned}&\frac{\partial \ln\det(\mathbf{A} + \mathbf{BXC})}{\partial \mathbf{X}} = \frac{\partial \ln\det(\mathbf{BYC})}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} = \\ &\left[\mathbf{C}(\mathbf{BYC})^{-1}\mathbf{B}\right]^T \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} = \left[\mathbf{C}(\mathbf{A} + \mathbf{BXC})^{-1}\mathbf{B}\right]^T \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \\ &= \left[\mathbf{C}(\mathbf{A} + \mathbf{BXC})^{-1}\mathbf{B}\right]^T.\end{aligned}$$

■

We can now compute the partial derivative of $\Theta_{\mathrm{link-phy}}(\mathbf{Q})$ with respect to $\mathbf{Q}_l$, which is given by

$$\frac{\partial \Theta_{\mathrm{link-phy}}(\mathbf{Q})}{\partial \mathbf{Q}_l} = W_l \frac{\partial}{\partial \mathbf{Q}_l}\left[\log_2\det\left(\mathbf{I} + \rho_j \mathbf{H}_j \mathbf{Q}_j \mathbf{H}_j^\dagger\right)\right].$$

Assuming that the channel gain matrices $\mathbf{H}$ are of full row rank (if not, we can always delete the linearly dependent rows). Applying Corollary 1 by letting $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = \rho_l\mathbf{H}_l$, $\mathbf{X} = \mathbf{Q}_l$, and $\mathbf{C} = \mathbf{H}_l^\dagger$, we have

$$\frac{\partial \Theta_{\mathrm{link-phy}}(\mathbf{Q})}{\partial \mathbf{Q}_l} = \frac{W_l u_l \rho_l}{\ln 2}\left[\mathbf{H}_l^\dagger\left(\mathbf{I} + \rho_l \mathbf{H}_l \mathbf{Q}_l \mathbf{H}_l^\dagger\right)^{-1}\mathbf{H}_l\right]^T,$$

where we have used the fact that $\mathbf{R}_l$ does not depend on $\mathbf{Q}_l$.

Recall that, for a function $f(z)$, where $z = x + jy$ is a complex variable, its derivative is defined as $\frac{\partial f(z)}{\partial z} = \frac{1}{2}\left(\frac{\partial f(z)}{\partial x} - j\frac{\partial f(z)}{\partial y}\right)$, and the gradient is defined as $\nabla_z f(z) = \frac{\partial f(z)}{\partial x} + j\frac{\partial f(z)}{\partial y}$ [11], [13]. Hence, we have $\nabla_z f(z) = 2\left(\frac{\partial f(z)}{\partial z}\right)^*$. Therefore,

$$\mathbf{G}_{\mathbf{Q}_l} = \frac{2W_l u_l \rho_l}{\ln 2}\mathbf{H}_l^\dagger\left(\mathbf{I} + \rho_l \mathbf{H}_l \mathbf{Q}_l \mathbf{H}_l^\dagger\right)^{-1}\mathbf{H}_l. \quad (8)$$

Note that in (8), since $\left(\mathbf{I} + \rho_j \mathbf{H}_l \mathbf{Q}_l \mathbf{H}_l^\dagger\right)^{-1}$ is a Hermitian matrix, $\mathbf{G}_{\mathbf{Q}_l}$ is also Hermitian.

B. A Polynomial-Time Algorithm for Performing Projection

Since $\mathbf{G}_{\mathbf{Q}_l}$ is Hermitian, it then follows that $\mathbf{Q}_l'(k) = \mathbf{Q}_l(k) + s_k\mathbf{G}_{\mathbf{Q}_l}(k)$ is Hermitian. On the other hand, since $\Omega_+(n)$ contains a constraint on power sum for each node n having $|\mathcal{O}(n)|$ outgoing links, we need to project the $|\mathcal{O}(n)|$ W-scalars and $|\mathcal{O}(n)|$ $\mathbf{Q}$-covariance matrices onto $\Omega_+(n)$ simultaneously.

Toward this end, we construct a block diagonal matrix $\mathbf{D}_n$ as follows:

$$\mathbf{D}_n = \left[\begin{array}{c|c} \mathbf{W}_n & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{Q}_n \end{array}\right] \in \mathbb{C}^{|\mathcal{O}(n)|(n_t+1)\times|\mathcal{O}(n)|(n_t+1)},$$

where $\mathbf{W}_n$ is defined as $\mathbf{W}_n \triangleq \mathrm{Diag}\{W_l : l \in \mathcal{O}(n)\} \in \mathbb{C}^{|\mathcal{O}(n)|\times|\mathcal{O}(n)|}$, and $\mathbf{Q}_n$ is defined as $\mathbf{Q}_n \triangleq \mathrm{Diag}\{\mathbf{Q}_l : l \in \mathcal{O}(n)\} \in \mathbb{C}^{|\mathcal{O}(n)|n_t\times|\mathcal{O}(n)|n_t}$. We introduce two more matrices $\mathbf{E}_1^{(n)}$ and $\mathbf{E}_2^{(n)}$ as follows:

$$\mathbf{E}_1^{(n)} = \left[\begin{array}{c|c} \mathbf{I}_{|\mathcal{O}(n)|} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \in \mathbb{C}^{|\mathcal{O}(n)|(n_t+1)\times|\mathcal{O}(n)|(n_t+1)},$$

$$\mathbf{E}_2^{(n)} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{|\mathcal{O}(n)|n_t} \end{array}\right] \in \mathbb{C}^{|\mathcal{O}(n)|(n_t+1)\times|\mathcal{O}(n)|(n_t+1)}.$$

It is easy to recognize that if $\mathbf{D}_n \in \Omega_+(n)$, we have $\mathrm{Tr}(\mathbf{E}_1^{(n)}\mathbf{D}_n) = \sum_{l\in\mathcal{O}(n)} W_l \le B_n$, $\mathrm{Tr}(\mathbf{E}_2^{(n)}\mathbf{D}_n) = \sum_{l\in\mathcal{O}(n)} \mathrm{Tr}(\mathbf{Q}_l) \le P_{\max}^{(n)}$, and $\mathbf{D}_n \succeq 0$. In this paper, Frobenius norm, which is the counterpart of the Euclidean norm in the vector space, is used as the matrix distance metric. By the definition of Frobenius norm, the distance between two matrices $\mathbf{A}$ and $\mathbf{B}$ is $\|\mathbf{A} - \mathbf{B}\|_F = \left(\mathrm{Tr}\left[(\mathbf{A} - \mathbf{B})^\dagger(\mathbf{A} - \mathbf{B})\right]\right)^{\frac{1}{2}}$. Thus, given the block diagonal matrix $\mathbf{D}_n$, we wish to find a matrix $\tilde{\mathbf{D}}_n \in \Omega_+(n)$ such that $\tilde{\mathbf{D}}_n$ minimizes $\|\tilde{\mathbf{D}}_n - \mathbf{D}_n\|_F$, i.e.,

$$\begin{array}{ll} \mathsf{Minimize} & \|\tilde{\mathbf{D}}_n - \mathbf{D}_n\|_F \\ \mathsf{subject\ to} & \mathrm{Tr}(\mathbf{E}_1^{(n)}\tilde{\mathbf{D}}_n) \le B_n \\ & \mathrm{Tr}(\mathbf{E}_2^{(n)}\tilde{\mathbf{D}}_n) \le P_{\max}^{(n)} \\ & \tilde{\mathbf{D}}_n \succeq 0 \end{array} \quad (9)$$

For more convenient algebraic manipulations, we instead consider the following equivalent optimization problem:

$$\begin{array}{ll} \mathsf{Minimize} & \frac{1}{2}\left\|\tilde{\mathbf{D}}_n - \mathbf{D}_n\right\|_F^2 \\ \mathsf{subject\ to} & \mathrm{Tr}(\mathbf{E}_1^{(n)}\tilde{\mathbf{D}}_n) \le B_n \\ & \mathrm{Tr}(\mathbf{E}_2^{(n)}\tilde{\mathbf{D}}_n) \le P_{\max}^{(n)} \\ & \tilde{\mathbf{D}}_n \succeq 0. \end{array} \quad (10)$$

Note that the objective function of this minimization problem is convex in $\tilde{\mathbf{D}}_n$, the constraint $\tilde{\mathbf{D}}_n \succeq 0$ represents the convex cone of positive semidefinite matrices, and the constraints $\mathrm{Tr}(\mathbf{E}_1^{(n)}\tilde{\mathbf{D}}_n) \le B_n$ and $\mathrm{Tr}(\mathbf{E}_2^{(n)}\tilde{\mathbf{D}}_n) \le P_{\max}^{(n)}$ are linear constraints. Therefore, this problem is a convex optimization problem, and we can solve this minimization problem by solving its Lagrangian dual problem.

Associating Hermitian matrix $\mathbf{\Pi}$ to the constraint $\tilde{\mathbf{D}}_n \succeq 0$, ν to the constraint $\mathrm{Tr}(\mathbf{E}_1^{(n)}\tilde{\mathbf{D}}_n) \le B_n$, and μ to the constraint $\mathrm{Tr}(\mathbf{E}_2^{(n)}\tilde{\mathbf{D}}_n) \le P_{\max}^{(n)}$, we can write the Lagrangian as

$$\begin{aligned} g(\mathbf{\Pi}, \nu, \mu) = \min_{\tilde{\mathbf{D}}_n}\Bigg\{&\frac{1}{2}\left\|\tilde{\mathbf{D}}_n - \mathbf{D}_n\right\|_F^2 - \mathrm{Tr}(\mathbf{\Pi}^\dagger\tilde{\mathbf{D}}_n) \\ &+\nu\left(\mathrm{Tr}[\mathbf{E}_1^{(n)}\tilde{\mathbf{D}}_n] - B_n\right) + \mu\left(\mathrm{Tr}[\mathbf{E}_2^{(n)}\tilde{\mathbf{D}}_n] - P_{\max}^{(n)}\right)\Bigg\}. \end{aligned} \quad (11)$$

Since $g(\mathbf{\Pi}, \nu, \mu)$ is a convex quadratic function in $\tilde{\mathbf{D}}_n$ and $\tilde{\mathbf{D}}_n$ becomes unconstrained after moving the positive semidefinite constraint to the objective function, we can compute the

minimizer of (11) by simply setting the derivative of (11) to zero, i.e.,

$$(\tilde{\mathbf{D}}_n - \mathbf{D}_n) - \mathbf{\Pi}^\dagger + \nu\mathbf{E}_1^{(n)} + \mu\mathbf{E}_2^{(n)} = 0.$$

Noting that $\mathbf{\Pi}^\dagger = \mathbf{\Pi}$, we have

$$\tilde{\mathbf{D}}_n = \mathbf{D}_n + \mathbf{\Pi} - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}. \tag{12}$$

Substituting $\tilde{\mathbf{D}}_n$ into (11), we have

$$\begin{aligned} g(\mathbf{\Pi},\nu,\mu) &= \frac{1}{2}\left\|\mathbf{\Pi} - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}\right\|_F^2 - \nu B_n - \mu P_{\max}^{(n)} \\ &+\mathrm{Tr}\left[\left(\nu\mathbf{E}_1^{(n)} + \mu\mathbf{E}_2^{(n)} - \mathbf{\Pi}\right)\left(\mathbf{D}_n + \mathbf{\Pi} - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}\right)\right] \\ &= -\frac{1}{2}\left\|\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} + \mathbf{\Pi}\right\|_F^2 - \nu B_n - \mu P_{\max}^{(n)} \\ &+\frac{1}{2}\|\mathbf{D}_n\|_F^2. \end{aligned}$$

Therefore, the Lagrangian dual problem can be written as

$$\begin{array}{ll} \text{Maximize} & -\frac{1}{2}\left\|\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} + \mathbf{\Pi}\right\|_F^2 \\ & -\nu B_n - \mu P_{\max}^{(n)} + \frac{1}{2}\|\mathbf{D}_n\|_F^2 \\ \text{subject to} & \mathbf{\Pi} \succeq 0, \nu \geq 0, \mu \geq 0. \end{array} \tag{13}$$

After solving (13) and by saddle-point optimality condition, we have the optimal solution to the primal problem as:

$$\tilde{\mathbf{D}}_n^* = \mathbf{D}_n - \nu^*\mathbf{E}_1^{(n)} - \mu^*\mathbf{E}_2^{(n)} + \mathbf{\Pi}^*, \tag{14}$$

where ν^*, μ^* and $\mathbf{\Pi}^*$ are the optimal solutions to Lagrangian dual problem in (13).

From matrix theory, we know that the eigenvalues of a Hermitian matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ are real. Suppose that we sort these eigenvalues of $\mathbf{A}$, denoted by λ_i, $i = 1,\ldots,p$, in non-increasing order, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, and perform eigenvalue decomposition on $\mathbf{A}$ yielding $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathrm{Diag}\{\lambda_i : i = 1,\ldots,p\}\mathbf{U}_\mathbf{A}^\dagger$. In this decomposition, $\mathbf{U}_\mathbf{A}$ is the unitary matrix formed by the eigenvectors corresponding to the non-increasing eigenvalues. Then, we have the positive semidefinite and negative semidefinite projections of $\mathbf{A}$ as follows:

$$\mathbf{A}_+ = \mathbf{U}_\mathbf{A}\mathrm{Diag}\{\max\{\lambda_1,0\},\ldots,\max\{\lambda_p,0\}\}\mathbf{U}_\mathbf{A}^\dagger, \tag{15}$$

$$\mathbf{A}_- = \mathbf{U}_\mathbf{A}\mathrm{Diag}\{\min\{\lambda_1,0\},\ldots,\min\{\lambda_p,0\}\}\mathbf{U}_\mathbf{A}^\dagger. \tag{16}$$

The proof of the results in (15) and (16) is a straightforward application of Moreau decomposition [14] by noting that $\mathbf{A}_+ \succeq 0$, $\mathbf{A}_- \preceq 0$, $\langle \mathbf{A}_+, \mathbf{A}_- \rangle = 0$, $\mathbf{A}_+ + \mathbf{A}_- = \mathbf{A}$, and the positive semidefinite cone and negative semidefinite cone are polar cones to each other.

Now we consider the term $\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} + \mathbf{\Pi}$, which is the only term involving $\mathbf{\Pi}$ in the dual objective function. We can rewrite it as $\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} - (-\mathbf{\Pi})$, where we note that $-\mathbf{\Pi} \preceq 0$. Finding a negative semidefinite matrix $-\mathbf{\Pi}$ such that $\|\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} - (-\mathbf{\Pi})\|_F$ is minimized is equivalent to finding the projection of $\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}$ onto the negative semidefinite cone. From our previous discussion, we have

$$-\mathbf{\Pi} = \left(\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}\right)_- . \tag{17}$$

Substituting (17) back to the Lagrangian dual objective function, we have

$$\begin{aligned} &\min_{\mathbf{\Pi}} \left\|\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)} + \mathbf{\Pi}\right\|_F \\ &= \left(\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)}\right)_+ . \end{aligned}$$

Thus, the matrix variable $\mathbf{\Pi}$ in the Lagrangian dual problem can be solved and the Lagrangian dual problem (13) can be simplified to

$$\begin{array}{ll} \text{Maximize} & \psi(\nu,\mu) \triangleq -\frac{1}{2}\|(\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+\|_F^2 \\ & -\nu B_n - \mu P_{\max}^{(n)} + \frac{1}{2}\|\mathbf{D}_n\|_F^2 \\ \text{subject to} & \nu \geq 0, \mu \geq 0. \end{array}$$

Suppose that after performing eigenvalue decomposition on $\mathbf{D}_n$, we have $\mathbf{D}_n = \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^\dagger$, where $\mathbf{\Lambda}_n$ is the diagonal matrix formed by the eigenvalues of $\mathbf{D}_n$, $\mathbf{U}_n$ is the unitary matrix formed by the corresponding eigenvectors. From the fact that $\mathbf{E}_1^{(n)} = \mathbf{U}_n\mathbf{E}_1^{(n)}\mathbf{U}_n^\dagger$ and $\mathbf{E}_2^{(n)} = \mathbf{U}_n\mathbf{E}_2^{(n)}\mathbf{U}_n^\dagger$, we have $(\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+ = \mathbf{U}_n(\mathbf{\Lambda}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+\mathbf{U}_n^\dagger$. Since $\mathbf{U}_n$ is unitary, we have $\|(\mathbf{D}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+\|_F^2 = \|(\mathbf{\Lambda}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+\|_F^2$. In particular, we denote the eigenvalues in $\mathbf{\Lambda}_n$ corresponding to $\mathbf{W}_n$ and $\mathbf{Q}_n$ by $\lambda_i^{(\mathbf{W}_n)}$ and $\lambda_j^{(\mathbf{Q}_n)}$, respectively, and sort the eigenvalues in these two groups in non-increasing order as follows:

$$\mathbf{\Lambda}_n = \mathrm{Diag}\{\lambda_1^{(\mathbf{W}_n)},\ldots,\lambda_{|\mathcal{O}(n)|}^{(\mathbf{W}_n)},\lambda_1^{(\mathbf{Q}_n)},\ldots,\lambda_{|\mathcal{O}(n)|\times n_t}^{(\mathbf{Q}_n)}\},$$

where $\lambda_1^{(\mathbf{W}_n)} \geq \ldots \geq \lambda_{|\mathcal{O}(n)|}^{(\mathbf{W}_n)}$ and $\lambda_1^{(\mathbf{Q}_n)} \geq \ldots \geq \lambda_{|\mathcal{O}(n)|\times n_t}^{(\mathbf{Q}_n)}$. It then follows that

$$\begin{aligned} &\|(\mathbf{\Lambda}_n - \nu\mathbf{E}_1^{(n)} - \mu\mathbf{E}_2^{(n)})_+\|_F^2 \\ &= \sum_{i=1}^{|\mathcal{O}(n)|}\left(\max\left\{0,\lambda_i^{(\mathbf{W}_n)} - \nu\right\}\right)^2 \\ &\quad + \sum_{j=1}^{|\mathcal{O}(n)|n_t}\left(\max\left\{0,\lambda_j^{(\mathbf{Q}_n)} - \mu\right\}\right)^2. \end{aligned} \tag{18}$$

From (18), we have

$$\begin{aligned} \psi(\nu,\mu) &= -\frac{1}{2}\sum_{i=1}^{|\mathcal{O}(n)|}\left(\max\left\{0,\lambda_i^{(\mathbf{W}_n)} - \nu\right\}\right)^2 \\ &\quad -\nu B_n - \frac{1}{2}\sum_{j=1}^{|\mathcal{O}(n)|n_t}\left(\max\left\{0,\lambda_j^{(\mathbf{Q}_n)} - \mu\right\}\right)^2 \\ &\quad -\mu P_{\max}^{(n)} + \frac{1}{2}\|\mathbf{D}_n\|_F^2 \\ &= \psi(\nu) + \psi(\mu) + \frac{1}{2}\|\mathbf{D}_n\|_F^2, \end{aligned} \tag{19}$$

where $\psi(\nu) \triangleq -\frac{1}{2}\sum_{i=1}^{|\mathcal{O}(n)|}(\max\{0,\lambda_i^{(\mathbf{W}_n)} - \nu\})^2 - \nu B_n$ and $\psi(\mu) \triangleq -\frac{1}{2}\sum_{j=1}^{|\mathcal{O}(n)|n_t}(\max\{0,\lambda_j^{(\mathbf{Q}_n)} - \mu\})^2 - \mu P_{\max}^{(n)}$, i.e., we separate $\psi(\nu,\mu)$ into two parts. It can be readily verified that $\psi(\nu,\mu)$ is continuous and piece-wise concave in ν and μ. Generally, piece-wise concave maximization problems can be solved by subgradient method. In this problem, it is easy to

derive the subgradients with respect to ν and μ as follows:

$$\frac{\partial \psi}{\partial \nu} = \sum_{i=1}^{|\mathcal{O}(n)|} \max\left\{0, \lambda_i^{(\mathbf{W}_n)} - \nu\right\} - B_n,$$

$$\frac{\partial \psi}{\partial \mu} = \sum_{j=1}^{|\mathcal{O}(n)|\times n_t} \max\left\{0, \lambda_j^{(\mathbf{Q}_n)} - \mu\right\} - P_{\max}^{(n)}.$$

However, due to the heuristic nature of its step size selection strategy, subgradient algorithm usually does not perform well. In fact, since $\psi(\nu, \mu)$ is piece-wise quadratic and separable, we can solve $\psi(\nu, \mu)$ by exploiting this special structure.

For example, we can start searching the optimal value of ν as follows. We use $\hat{I}$ to index the pieces of $\psi(\nu)$, $\hat{I} = 0, 1, \ldots, |\mathcal{O}(n)|$. Initially we set $\hat{I} = 0$ and increase $\hat{I}$ subsequently. Also, we introduce $\lambda_0^{(\mathbf{W}_n)} = \infty$ and $\lambda_{|\mathcal{O}(n)|+1}^{(\mathbf{W}_n)} = -\infty$. If $\hat{I} = 0$, we let endpoint objective value $\psi_{\hat{I}}\left(\lambda_0^{(\mathbf{W}_n)}\right) = 0$, $\phi^* = \psi_{\hat{I}}\left(\lambda_0^{(\mathbf{W}_n)}\right)$, and let $\nu^* = \lambda_0^{(\mathbf{W}_n)}$. If $\hat{I} > |\mathcal{O}(n)|$, the search stops. For a particular index $\hat{I}$, suppose that $\nu \in \left[\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}, \lambda_{\hat{I}}^{(\mathbf{W}_n)}\right] \cap \mathbb{R}_+$, where $\mathbb{R}_+$ denotes the set of non-negative real numbers. Solve $\nu_{\hat{I}}^*$ by setting

$$\frac{\partial}{\partial \nu}\psi_{\hat{I}}(\nu) \triangleq \frac{\partial}{\partial \nu}\left(-\frac{1}{2}\sum_{i=1}^{\hat{I}}\left(\lambda_i^{(\mathbf{W}_n)} - \nu\right)^2 - \nu B_n\right) = 0,$$

and we have

$$\nu_{\hat{I}}^* = \frac{\sum_{i=1}^{\hat{I}} \lambda_i^{(\mathbf{W}_n)} - B_n}{\hat{I}}. \tag{20}$$

Now we consider the following two cases:

1) If $\nu_{\hat{I}}^* \in [\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}, \lambda_{\hat{I}}^{(\mathbf{W}_n)}] \cap \mathbb{R}_+$, then we have already found the optimal solution for ν because $\psi(\nu, \mu)$ is continuous concave quadratic in ν, and the point with zero-value first derivative, if exists, must be the unique global maximum solution. Thus, we can let $\nu^* = \nu_{\hat{I}}^*$ and the search is done.
2) If $\nu_{\hat{I}}^* \notin [\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}, \lambda_{\hat{I}}^{(\mathbf{W}_n)}] \cap \mathbb{R}_+$, we must have that the local maximum in the interval $[\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}, \lambda_{\hat{I}}^{(\mathbf{W}_n)}] \cap \mathbb{R}_+$ is achieved at one of the two end points. We note that the objective value $\psi_{\hat{I}}\left(\lambda_{\hat{I}}^{(\mathbf{W}_n)}\right)$ has been computed in the previous iteration. This is because from the continuity of the objective function, we have $\psi_{\hat{I}}\left(\lambda_{\hat{I}}^{(\mathbf{W}_n)}\right) = \psi_{\hat{I}-1}\left(\lambda_{\hat{I}}^{(\mathbf{W}_n)}\right)$. Thus, we only need to compute the objective value $\psi_{\hat{I}}\left(\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}\right)$ of another endpoint. If $\psi_{\hat{I}}\left(\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}\right) < \psi_{\hat{I}}\left(\lambda_{\hat{I}}^{(\mathbf{W}_n)}\right) = \phi^*$, then we know ν^* is the optimal solution; else let $\nu^* = \lambda_{\hat{I}+1}^{(\mathbf{W}_n)}$, $\phi^* = \psi_{\hat{I}}\left(\lambda_{\hat{I}+1}^{(\mathbf{W}_n)}\right)$, let $\hat{I} \leftarrow \hat{I} + 1$ and continue.

Since there are $|\mathcal{O}(n)|+1$ intervals, the search process takes at most $|\mathcal{O}(n)| + 1$ steps to find the optimal solution ν^*. Likewise, the search process for μ can be done in a similar fashion.

After finding ν^* and μ^*, we have

$$\begin{aligned}\tilde{\mathbf{D}}_n^* &= \left(\mathbf{D}_n - \nu^*\mathbf{E}_1^{(n)} - \mu^*\mathbf{E}_2^{(n)}\right)_+ \\ &= \mathbf{U}_n\left(\mathbf{\Lambda}_n - \nu^*\mathbf{E}_1^{(n)} - \mu^*\mathbf{E}_2^{(n)}\right)_+\mathbf{U}_n^\dagger, \end{aligned} \tag{21}$$

That is, the projection $\tilde{\mathbf{D}}_n$ can be computed by adjusting the eigenvalues of $\mathbf{D}_n$ using ν^* and μ^* and keeping the eigenvectors unchanged.

The projection of $\mathbf{D}_n$ onto $\Omega_+(n)$ is summarized in Algorithm 2 and Algorithm 3.

Algorithm 2 Projection onto $\Omega_+(n)$

1. Construct a block diagonal matrix $\mathbf{D}_n$. Perform eigenvalue decomposition $\mathbf{D}_n = \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^\dagger$; separate the eigenvalues in two groups corresponding to $\mathbf{W}_n$ and $\mathbf{Q}_n$; sort them in non-increasing order within each group, respectively.
2. For each group of eigenvalues, use Algorithm 3 to find the optimal dual variables ν^* and μ^*.
3. Compute $\tilde{\mathbf{D}}_n = \mathbf{U}_n(\mathbf{\Lambda}_n - \nu^*\mathbf{E}_1^{(n)} - \mu^*\mathbf{E}_2^{(n)})_+\mathbf{U}_n^\dagger$.

Algorithm 3 Search the Optimal Dual Variable ν^* and μ^*

Initiation:
Introduce $\lambda_0 = \infty$ and $\lambda_K = -\infty$. Let $\hat{I} = 0$. Let endpoint objective $\psi_{\hat{I}}(\lambda_0) = 0$, $\phi^* = \psi_{\hat{I}}(\lambda_0)$, and $\mu^* = \lambda_0$.
Main Loop:
1. If $\hat{I} > K$, return μ^*; else let $\mu_{\hat{I}}^* = (\sum_{j=1}^{\hat{I}} \lambda_j - P)/\hat{I}$.
2. If $\mu_{\hat{I}}^* \in [\lambda_{\hat{I}+1}, \lambda_{\hat{I}}] \cap \mathbb{R}_+$, then let $\mu^* = \mu_{\hat{I}}^*$ and return μ^*.
3. Compute $\psi_{\hat{I}}(\lambda_{\hat{I}+1})$. If $\psi_{\hat{I}}(\lambda_{\hat{I}+1}) < \phi^*$, then return μ^*; else let $\mu^* = \lambda_{\hat{I}+1}$, $\phi^* = \psi_{\hat{I}}(\lambda_{\hat{I}+1})$, $\hat{I} \leftarrow \hat{I} + 1$ and go to Step 1.

V. Solving the Lagrangian Dual Problem

A. A Cutting-Plane Method Based on Outer-Linearization

We briefly review the basic idea of cutting-plane method as follows. Let $z = \Theta(\mathbf{u})$. The inequality $z \geq \sum_f \ln(s_f) + \sum_l u_l\left(\Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l\rangle\right)$ must hold for all $(\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}) \in \Gamma$. Thus, the dual problem is equivalent to

$$\begin{array}{ll}\text{Minimize} & z \\ \text{subject to} & z \geq \sum_f \ln(s_f) + \\ & \sum_l u_l\left(\Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l\rangle\right) \\ & \mathbf{u} \geq 0, \end{array} \tag{22}$$

where $(\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}) \in \Gamma$. However, (22) is a linear program with infinite constraints not known explicitly. Instead, we can consider the following *approximating problem*:

$$\begin{array}{ll}\text{Minimize} & z \\ \text{subject to} & z \geq \sum_f \ln\left(s_f^{(j)}\right) + \\ & \sum_l u_l\left(\Phi_l(W_l^{(j)}, \mathbf{Q}_l^{(j)}) - \langle \mathbf{1}, \mathbf{X}^{(j)T}\mathbf{e}_l\rangle\right) \\ & \mathbf{u} \geq 0, \end{array} \tag{23}$$

where the points $(\mathbf{S}^{(j)}, \mathbf{X}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{W}^{(j)}) \in \Gamma$, for $j = 1, \ldots, k-1$. The problem in (23) is a linear program with a finite number of constraints and can be solved efficiently. Let $(z^{(k)}, \mathbf{u}^{(k)})$ be an optimal solution to the approximating problem, which we refer to as the *master program*. If the solution is feasible to (22), then it is an optimal solution to the Lagrangian dual problem. To check feasibility, we consider the following *subproblem*:

$$\begin{array}{ll}\text{Maximize} & \sum_f \ln(s_f) + \\ & \sum_l u_l^{(k)}\left(\Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T\mathbf{e}_l\rangle\right) \\ \text{subject to} & (\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}) \in \Gamma. \end{array} \tag{24}$$

Suppose that $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$ is an optimal solution to the subproblem (24) and $\Theta^*(\mathbf{u}^{(k)})$ is the corresponding optimal objective value. If $z_k \geq \Theta^*(\mathbf{u}^{(k)})$, then $\mathbf{u}^{(k)}$ is an optimal solution to the Lagrangian dual problem. Otherwise, for $\mathbf{u} = \mathbf{u}^{(k)}$, the inequality constraint in (22) is not satisfied for $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$. Thus, we can add the constraint

$$z \geq \sum_f \ln\left(s_f^{(k)}\right) + \sum_l u_l \left(\Phi_l(W_l^{(k)}, \mathbf{Q}_l^{(k)}) - \langle \mathbf{1}, \mathbf{X}^{(k)T}\mathbf{e}_l\rangle\right) \quad (25)$$

to (23), and solve the master linear program again. Obviously, $(z^{(k)}, \mathbf{u}^{(k)})$ violates (25) and will be cut off by (25). The cutting plane algorithm is summarized in Algorithm 4.

Algorithm 4 Cutting Plane Algorithm for Solving $\mathbf{D}^{\mathrm{CRPBA}}$

Initialization:
Find a point $(\mathbf{S}^{(0)}, \mathbf{X}^{(0)}, \mathbf{Q}^{(0)}, \mathbf{W}^{(0)}) \in \Gamma$. Let $k = 1$.
Main Loop:
1. Solve the master program in (23). Let $(z^{(k)}, \mathbf{u}^{(k)})$ be an optimal solution.
2. Solve the subproblem in (24). Let $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$ be an optimal point, and let $\Theta^*(\mathbf{u}^{(k)})$ be the corresponding optimal objective value.
3. If $z^{(k)} \geq \Theta(\mathbf{u}^{(k)})$, then stop with $\mathbf{u}^{(k)}$ as the optimal dual solution. Otherwise, add the constraint (25) to the master program, replace k by $k+1$, and go to Step 1.

B. Subgradient-Based Mechanism

Since the dual objective function is piece-wise differentiable [9], subgradient method can be used to solve the master dual problem. Subgradient algorithm for minimization problems is a generalization of steepest descent algorithm in which the negative gradient direction is replaced by a suitable negative subgradient direction. For $\Theta(\mathbf{u})$, starting with an initial $\mathbf{u}^{(0)}$ and after evaluating subproblems $\Theta_{\mathrm{net}}(\mathbf{u})$ and $\Theta_{\mathrm{link-phy}}(\mathbf{u})$ for $\mathbf{u}^{(k)}$ in the k-th iteration, we update the dual variables by

$$\mathbf{u}^{(k+1)} = \left[\mathbf{u}^k - s_k \mathbf{d}^{(k)}\right]_+. \quad (26)$$

In (26), the operator $[\cdot]_+$ projects a vector onto the nonnegative orthant, and s_k denotes a positive scalar step size. $\mathbf{d}^{(k)}$ is a subgradient of the Lagrangian at point $\mathbf{u}^{(k)}$. Although it has been shown in [9] that the subgradient algorithm converges if the step size s_k satisfies $s_k \to 0$ as $k \to \infty$, $\sum_{k=0}^{\infty} s_k = \infty$, and $\sum_{k=0}^{\infty} s_k^2 < \infty$, one has to carefully select the step size to avoid stalling and accelerate the convergence. It is shown in [9] that the best choice of the step size s_k is

$$s_k = \frac{\beta_k[\Theta(\mathbf{u}^{(k)}) - \hat{\Theta}]}{\|\mathbf{d}_k\|}, \quad (27)$$

where $\beta_k > 0$ and $\hat{\Theta}$ is an estimate of the optimal value of Θ. However, this optimal step size selection strategy requires global information and the estimation of Θ, which is difficult in implementation. Another possible step size selection strategy is the divergent harmonic series

$$\beta \sum_{k=1}^{\infty} \frac{1}{k} = \infty. \quad (28)$$

For the dual master problem, the subgradient for the Lagrangian dual problem is

$$d_l^{(k)} = \Phi_l(\mathbf{Q}^*(\mathbf{u})) - \langle \mathbf{1}, \mathbf{X}^*(\mathbf{u})^T \mathbf{e}_l\rangle, \quad l = 1, 2, \ldots, L. \quad (29)$$

It is worth pointing out that the dual variables $\mathbf{u}^{(k)}$ can be economically interpreted as "prices" of the links during the k-th iteration. The subgradients $\mathbf{d}^{(k)}$ indicates the excess capacities of the links during the k-th iteration. The dual updating scheme of the subgradient algorithm can be viewed as a pricing scheme. When a link, say link k, is under-utilized, then $d_l^{(k)} > 0$. From (26), we can see that the price of link k will be reduced. On the other hand, when link k is over-utilized, then $d_l^{(k)} < 0$. Again, from (26), it can be seen that the price of that link k will be increased. The subgradient algorithm is summarized in Algorithm 5.

Algorithm 5 A Subgradient Algorithm for Solving MWSR

Initialization:
Choose the initial starting points $\mathbf{u}^{(0)}$. Let $k = 0$.
Main Loop:
1. Compute $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$ by solving the network layer and link layer subproblems.
2. Choose an appropriate step size s_k. Compute the subgradient $\mathbf{d}^{(k)}$ using (29) with $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$.
3. Update dual variables $\mathbf{u}^{(k)}$ using (26) with $\mathbf{d}^{(k)}$.
4. If $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\| < \epsilon$, then return $(\mathbf{S}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{W}^{(k)})$ as the final optimal solution and stop. Otherwise, let $k \leftarrow k+1$ and go to Step 1.

C. Recovering Primal Optimal Solution

Thus far, we have investigated the procedures for solving the Lagrangian dual problem. During the course of solving the dual problem, the following problem, which is used to evaluate $\Theta(\mathbf{u})$ at $\mathbf{u}$, needs to be solved:

$$\begin{array}{ll} \mathsf{Maximize} & \sum_f \ln(s_f) + \\ & \sum_l u_l \left(\Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l\rangle\right) \\ \mathsf{subject\ to} & (\mathbf{S}, \mathbf{X}, \mathbf{Q}, \mathbf{W}) \in \Gamma. \end{array} \quad (30)$$

Suppose that $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Q}^*, \mathbf{W}^*)$ is an optimal solution to (30). If $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Q}^*, \mathbf{W}^*)$ is also feasible to the primal problem and $(\mathbf{u}^*)^T \left(\Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l\rangle\right) = 0$, then it is clear that $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Q}^*, \mathbf{W}^*, \mathbf{u}^*)$ is a saddle point and $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Q}^*, \mathbf{W}^*)$ solves the primal problem. However, $(\mathbf{S}^*, \mathbf{X}^*, \mathbf{Q}^*, \mathbf{W}^*)$ may not be feasible to the primal problem in general since it only solves (30), a problem related but different to the primal problem. Therefore, extra effort has to be taken to recover a primal optimal solution when the saddle point conditions do not hold.

Toward this end, suppose that $(\mathbf{S}^{(j)}, \mathbf{X}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{W}^{(j)})$, for $j = 1, 2, \ldots, k$ are the optimal solution of (30) for $\mathbf{u} = \mathbf{u}^{(j)}$. Now, let us consider the following linear programming problem:

$$\begin{array}{ll} \mathsf{Maximize} & \sum_{j=0}^{k} \tau_j \sum_{f=1}^{F} \log(s_f^{(j)}) \\ \mathsf{subject\ to} & \sum_{j=0}^{k} \tau_j \left(\langle \mathbf{1}, \mathbf{X}^{(j)T}\mathbf{e}_l\rangle - \Phi_l(W_l^{(j)}, \mathbf{Q}_l^{(j)})\right) \leq 0, \forall l \\ & \sum_{j=0}^{k} \tau_j (\mathbf{S}^{(j)}, \mathbf{X}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{W}^{(j)}) \in \Gamma \\ & \sum_{j=0}^{k} \tau_j = 1 \\ & \tau_j \geq 0, \forall j. \end{array} \quad (31)$$

We have the following theorem for recovering a primal feasible near-optimal solution.

Theorem 1: Let τ_j^*, for $j = 1, \ldots, k$ be an optimal solution to (31). Then $(\bar{\mathbf{S}}, \bar{\mathbf{X}}, \bar{\mathbf{Q}}, \bar{\mathbf{W}}) = \sum_{j=1}^{k} \tau_j (\mathbf{S}^{(j)}, \mathbf{X}^{(j)}, \mathbf{Q}^{(j)}, \mathbf{W}^{(j)})$ is a feasible solution to the primal problem. Furthermore, let $\bar{V}$ be the objective value corresponding to $(\bar{\mathbf{S}}, \bar{\mathbf{X}}, \bar{\mathbf{Q}}, \bar{\mathbf{W}})$, $V_k = \sum_{j=0}^{k} \tau_j^* \sum_{f=1}^{F} \log(s_f^{(j)})$, and V^* be the true primal optimal objective value. If $\Theta(\mathbf{u}) - \bar{V} \leq \epsilon$ for some $\mathbf{u} \geq \mathbf{0}$, then $\bar{V} \geq V^* - \epsilon$.

Proof: Since the function $\langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l \rangle - \Phi_l(W_l, \mathbf{Q}_l)$ is convex, we have that $\langle \mathbf{1}, \bar{\mathbf{X}}^T \mathbf{e}_l \rangle - \Phi_l(\bar{W}_l, \bar{\mathbf{Q}}_l) \leq \sum_{j=0}^{k} \tau_j^* (\langle \mathbf{1}, \mathbf{X}^{(j)T} \mathbf{e}_l \rangle - \Phi_l(W_l^{(j)}, \mathbf{Q}_l^{(j)})) \leq 0$. Thus, $(\bar{\mathbf{S}}, \bar{\mathbf{X}}, \bar{\mathbf{Q}}, \bar{\mathbf{W}})$ is feasible to the primal problem. Noting the concavity of $\sum \log(\cdot)$, we have

$$\bar{V} \geq \sum_{j=0}^{k} \tau_j^* \sum_{f=1}^{F} \log(s_{f,j}) = V_k \geq \Theta(\mathbf{u}) - \epsilon \geq V^* - \epsilon.$$

This completes the proof. ∎

It is worth pointing out that there is no need to solve (31) separately when we use the cutting-plane method to solve the Lagrangian dual problem. This is because (31) is precisely the linear programming dual of (23). As a result, the values of τ_j are immediately available after we solve (23).

VI. Distributed Implementation

As mentioned earlier, the Lagrangian dual problem is solvable by subgradient algorithm. Although the original motivation in applying subgradient algorithm is the non-differentiability of Lagrangian dual objective function (piecewise concave/convex function in general), we find that the subgradient method has the additional advantage of being amenable to distributed implementation. Specifically, the subgradient method has the following properties.

- Subgradient computation only requires local traffic information $\langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l \rangle$ and the available link capacity information $\Phi_l(W_l, \mathbf{Q}_l)$ at each link l. The subgradient can be computed as $\partial\Theta(\mathbf{u})/\partial u_l = \Phi_l(W_l, \mathbf{Q}_l) - \langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l \rangle$. In this case, subgradient only involves local variables W_l, $\mathbf{Q}_l$, and $\mathbf{X}^T \mathbf{e}_l$ at each link and thus can be computed locally.
- The choice of step sizes can be chosen as $\lambda_k = \beta/k$, $k = 1, 2, \ldots$, where $0 < \beta \leq 1$ is a predefined constant. This step size choice obviously satisfies the convergence conditions. This choice of step size depends only on the iteration index k (can be defined as some function of elapsed time), and does not require any other global knowledge. In conjunction with the first property, the dual variable, in the iterative form of $u_l^{(k+1)} = u_l^{(k)} + \lambda_k(\partial\Theta_l(\mathbf{u})/\partial u_l)$, can also be computed locally.
- The objective functions $\Theta_{\text{link-phy}}$ and Θ_{net} can be decomposed such that each node in the network can perform the computation locally. Recall that the link layer subproblem is: Max $\Theta_{\text{link-phy}} \triangleq \sum_{l=1}^{L} u_l C_l(\mathbf{Q}_l)$ s.t. $\sum_{l \in \mathcal{O}(n)} \text{Tr}\{\mathbf{Q}_l\} \leq P_{\max}, \mathbf{Q}_l \succeq 0, \forall l$, where $\mathcal{O}(n)$ denotes the set of outgoing links from node n. For this subproblem, it is not hard to see that it can be decomposed into a set of new subproblems at each node n of the following form:

$$\begin{array}{ll} \text{Maximize} & \Theta_{\text{link-phy}}^{(n)} \triangleq u_l C_l(\mathbf{Q}_l) \\ \text{subject to} & \sum_{l \in \mathcal{O}(n)} \text{Tr}\{\mathbf{Q}_l\} \leq P_{\max} \\ & \mathbf{Q}_l \succeq 0, l \in \mathcal{O}(n). \end{array} \tag{32}$$

The original dual link layer subproblem can then be transformed to $\Theta_{\text{link-phy}} = \sum_{n=1}^{N} \Theta_{\text{link-phy}}^{(n)}$. This suggests that the optimization of each problem $\Theta_{\text{link-phy}}^{(n)}$ in (32) only requires local information of channel gains (all outgoing links from node n) and the locally-computed dual variable u_l. Thus, the link layer subproblem can be solved distributively. Likewise, for the network layer subproblem (i.e., Max $\Theta_{\text{net}} \triangleq \sum_{f=1}^{F} \ln(s_f) - \sum_l u_l \langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l \rangle$, s.t. flow balance constraints for all flows), we may decompose it into a set of subproblems based on the source node of each session f:

$$\begin{array}{ll} \text{Maximize} & \Theta_{\text{net}}^{(f)} \triangleq \ln(s_f) - \sum_l u_l \langle \mathbf{1}, \mathbf{X}^T \mathbf{e}_l \rangle \\ \text{subject to} & \text{flow balance constraints for flow } f. \end{array} \tag{33}$$

The original network layer subproblem can then be simply transformed to $\Theta_{\text{net}} = \sum_{f=1}^{F} \Theta_{\text{net}}^{(f)}$. Again, this suggests that the optimization of each problem $\Theta_{\text{net}}^{(f)}$ in (33) only requires the locally-computed dual variable u_l for the links. In each iteration, all links can send the locally-computed dual information back to the source node of each session. As a result, the network layer subproblem can be solved in a distributed fashion.

Algorithm 6 Distributed Implementation

Initialization:
Initialize the iteration index $k = 0$, and choose initial values for $u_l^{(0)}$, for all l.

Main Loop:
1. Each node independently solves the decomposed link layer subproblem in (32). After that, each node independently updates dual variables u_l for all its outgoing links using (26) (where the step size and subgradient computation follow (28) and (29)) and broadcast them to its next hop neighbor.
2. Upon receiving some dual information from other neighbors, each node relays it to its next hop neighbor excluding the node where this information comes from.
3. Upon receiving all links' dual information $\mathbf{u}$, each source node solves the decomposed network layer subproblem in (33). After that, each source node f performs source routing and store these routing information $x_l^{(f)}, \forall l$ in the header.
4. Upon receiving the source routing information, each intermediate node routes the packets according to the routing information $x_l^{(f)}$ in the header.
5. Based on current values of dual variables $u_l^{(k)}$ and the iteration number k, compute $u_l^{(k+1)}$. If $u_l^{(k+1)} - u_l^{(k)} < \epsilon$, or k has reached a predefined number of iterations, the algorithm stops. Otherwise, let $k \leftarrow k + 1$ and go to Step 1.

The key steps in this distributed algorithm include:

1) Initialize the iteration index $k = 0$, and choose initial values for $u_l^{(0)}$, for all l.
2) At node n, based on the current dual information u_l, where $l \in \mathcal{O}(n)$, node n can solve the decomposed link layer subproblem in (32). After that, node n updates dual variables u_l for all its outgoing links using (26) (where the step size and subgradient computation follow

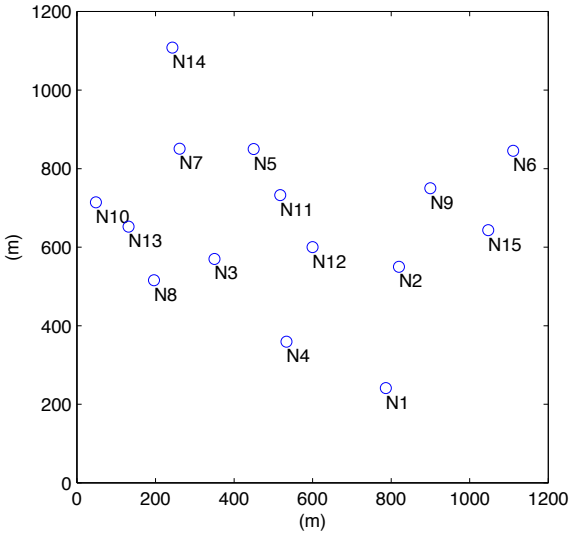

Fig. 1. Network topology of a 15-node ad hoc network.

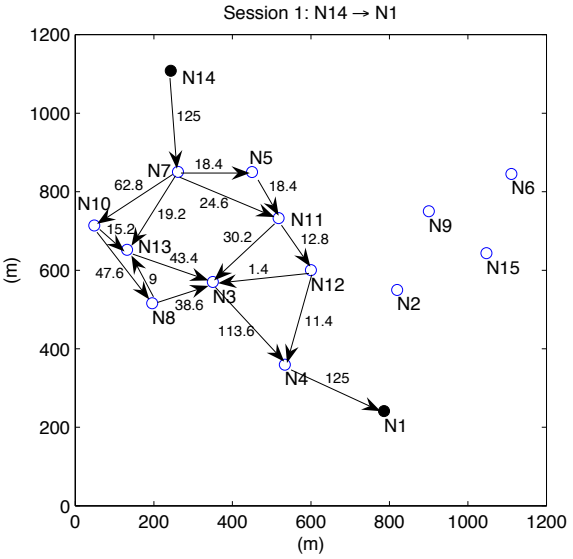

Fig. 2. Routing and flow rates of session 1 (in Mb/s).

(28) and (29)) and broadcast them to its next h neighbor. In the meantime, upon receiving some du information from other neighbors, node n relays it its next hop neighbor excluding the node where tl information comes from. Since the network is assum to be connected, these dual information will eventua reach each source node.

3) At source node f, upon receiving all links' dual infe mation $\mathbf{u}$, node s solves the decomposed network lay subproblem in (33). After that, source node f has updated flow rate s_f and updated routing informati $x_l^{(f)}$, $\forall l$. Then, source n shall be able to do sour routing and store these routing information $x_l^{(f)}$, $\forall l$ the packet headers.
4) Upon receiving the source routing information, ea intermediate node n will route the packets accordi to the routing information $x_l^{(f)}$ in the packet header.
5) Based on current values of dual variables $u_l^{(k)}$ and the iteration number k, compute $u_l^{(k+1)}$. If $u_l^{(k+1)} - u_l^{(k)} < \epsilon$, or k has reached a predefined number of iterations, the algorithm stops. Otherwise, let $k = k + 1$ and continue.

The distributed implementation of subgradient algorithm is summarized in Algorithm 6.

VII. Numerical Results

In this section, we present some pertinent numerical results for our solution procedure. We first describe our simulation settings. As shown in Fig. 1, we have 15 nodes uniformly distributed in a square region of 1200m × 1200m. Each node in the network is equipped with two antennas. The maximum transmit power for each node is set to $P_{\max} = 20$ dBm (100 mW). The path loss index is 3. The total bandwidth at each node is 20 MHz. There are three sessions in the network: node 14 to node 1, node 6 to node 10, and node 5 to node 4.

After executing our solution procedure, we find that the optimal rates for these three sessions are $s_1 = 125$ Mb/s,

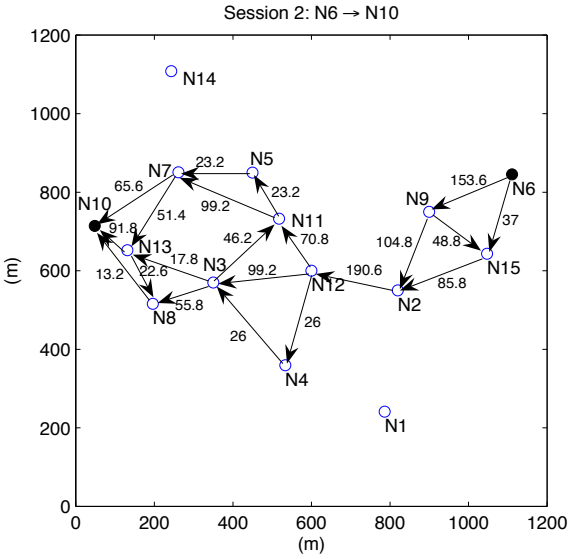

Fig. 3. Routing and flow rates of session 2 (in Mb/s).

$s_2 = 190.6$ Mb/s, and $s_3 = 258$ Mb/s. The routings and flow rates of sessions 1, 2, and 3 are shown in Fig. 2, Fig. 3, and Fig. 4, respectively. These figures show that flow routings for sessions 1, 2, and 3 are all multi-path and multi-hop. It can easily be verified that the flow rates in Figures 2, 3, and 4 satisfy flow conservation.

Denote $W_{(i,j)}$ and $\mathbf{Q}_{(i,j)}$ the bandwidth allocation and power allocation for the transmission from node i to node j. Table I shows the optimal bandwidth allocation of the network. Table II shows the optimal power allocation of the network. In Table I, the value in each cell represents the fraction of total bandwidth of the link's transmitting node. For example, $W_{(11,7)} = 0.23$ means that 0.23 of N11's total bandwidth is allocated to the transmission from N11 to N7. In Table II, each cell with four entries corresponds to a 2×2 $\mathbf{Q}$ matrix, which represents a power allocation. Take the transmission from N11 to N7 for example, $\mathbf{Q}_{(11,7)}$ in Table II

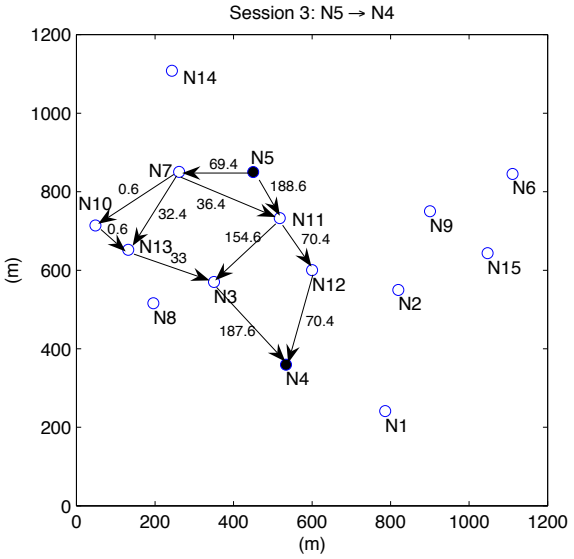

Fig. 4. Routing and flow rates of session 3 (in Mb/s).

says that the power allocations to the two antennas at N11 are 11.83 mW and 12.10 mW. Also, the signals sent through these two antennas, denoted by x_1 and x_2, should also be correlated with power $\mathbb{E}[x_1 x_2^\dagger] = (-0.25 + 0.01i)$ mW and $\mathbb{E}[x_2 x_1^\dagger] = (-0.25 - 0.01i)$ mW.

It can be observed from Table I and Table II that not every node allocates its full power and total bandwidth for its outgoing links. For example, N14 has only one outgoing link $(14, 7)$. We can see that $W_{(14,7)} = 0.71$ and $\mathrm{Tr}(\mathbf{Q}_{(14,7)}) = 79.47$ mW, which shows N14 does not transmit at full power and utilize all of its assigned bandwidth. This is due to the existence of bottleneck nodes in the network. Even if the non-bottleneck nodes increase the total transmit power and bandwidth, it will not help increase its session rate because the end-to-end session rate is bounded by the minimum bottleneck link along the path. In this network, it can be verified that N3, N7, N11, and N12 are bottleneck nodes. For example, at N11, $W_{(11,3)} + W_{(11,5)} + W_{(11,7)} + W_{(11,12)} = 1$ and $\mathrm{Tr}(Q_{(11,3)}) + \mathrm{Tr}(\mathbf{Q}_{(11,5)}) + \mathrm{Tr}(\mathbf{Q}_{(11,7)}) + \mathrm{Tr}(\mathbf{Q}_{(11,12)}) = 100$ mW. This means that the power and bandwidth at N11 has been fully utilized and cannot be further increased.

We plot the convergence behavior of gradient projection algorithm for the link-physical layer subproblem (with dual variable $\mathbf{u} = \mathbf{1}$) in Fig. 5. It can be seen that GP takes 28 iterations to converge to the maximum objective value of $\Theta_{\mathrm{link-phy}}(\mathbf{u})$ with $\mathbf{u} = \mathbf{1}$.

The convergence process of the Lagrangian dual problem is illustrated in Fig. 6. The step size selection strategy for the distributed subgradient is $\lambda_k = \frac{0.1}{k}$. In this figure, "Dual UB" denotes the current objective value of the Lagrangian dual function, which can be thought of as an upper bound of the primal objective value. "Primal Feasible Solution" denotes the current primal feasible solution recovered from the Lagrangian dual, which can be thought of as a lower bound of the optimal primal objective value. During each iteration, the cutting-plane and subgradient methods each solve the Lagrangian dual problem. The upper bounds of the optimal objective value are non-increasing with iterations. Meanwhile, the primal feasible objective values keep increasing with iterations. As expected, the upper bounds and the lower bound converge and give the optimal solution, as shown in this figure. We find that for this 15-node ad hoc network, centralized cutting-plane algorithm and distributed subgradient algorithm converge in approximately 115 iterations and 160 iterations, respectively. The optimal value of the network utility function is 6.64 (in $\log(\mathrm{b/s/Hz})$).

For performance gain comparison, we compare the heuristic strategy discussed in [6], which uses equal power allocation and equal bandwidth allocation to all the outgoing links at each node, which yields an objective value of 3.01. In comparison, the performance gap to our optimal value (6.64) is significant.

VIII. Related Work

In this section, we give a synopsis of related work on MIMO for single and multi-hop wireless networks. For single-hop, there is much research on MIMO-based systems for cellular networks [15]–[20]. In this setting, there are two types of channels: the MIMO multiple access channel (MIMO-MAC), which is associated with "uplinks," and the MIMO broadcast channel (MIMO-BC), which is associated with "downlinks." In [20], Yu *et al.* showed that the maximum sum rate of MIMO-MAC can be solved via the so-called "Iterative Water-Filling" method by exploiting the nice convexity structure of the problem. Very recently, Weingarten *et al.* [21] showed that the capacity region for MIMO-BC coincides with "dirty-paper coding" (DPC) rate region [22]. In particular, it was shown that there is a nice duality between MIMO-BC and MIMO-MAC [23]–[25]. By exploiting this duality, the non-convex capacity region problem of MIMO-BC can be computed by solving an equivalent MIMO-MAC problem under a power sum constraint [16]–[18], [26], [27].

For multiuser MIMO-based single-hop ad hoc networks (i.e., without any infrastructure), the maximum sum rate problem becomes much more challenging due to its non-convex nature. In [28], Jorswieck and Boche analyzed the worst-case performance of a multiuser MIMO system with interference. In [29], [30], Demirkol and Ingram introduced an iterative (trial-and-error) method based on stream control for some simple network topologies. In [31], Sundaresan *et al.* proposed a MAC control scheme along ideas similar to those in [29], [30]. In [32], Chen and Gans analyzed the asymptotic behavior of network spectral efficiency with L simultaneous co-channel transmission pairs. They showed that, in the absence of channel state information (CSI) at the transmitters, the network asymptotic spectral efficiency is limited by n_r nats/s/Hz as $L \to \infty$, and at least $n_t + n_r + 2\sqrt{n_t \cdot n_r}$ nats/s/Hz when CSI is available at the transmitters. In contrast to asymptotic analysis, in [11], Ye and Blum studied a finite-sized network and designed algorithms to find local optimal solutions.

For MIMO-based multi-hop ad hoc networks, research results remain limited. In [33], Hu and Zhang studied the joint problem of MAC and routing, with a consideration of optimal hop distance to minimize end-to-end delay. In [34], Sundaresan and Sivakumar used simulations to study various characteristics and tradeoffs (multiplexing gain vs.

TABLE I
BANDWIDTH ALLOCATION OF THE 15-NODE NETWORK ($\times$20 MHz).

$W_{(1,4)}$	0.71	$W_{(4,1)}$	0.44	$W_{(2,9)}$	0.24	$W_{(9,2)}$	0.35	$W_{(2,12)}$	0.41	$W_{(12,2)}$	0.18	$W_{(2,15)}$	0.24
$W_{(15,2)}$	0.40	$W_{(3,4)}$	0.17	$W_{(4,3)}$	0.24	$W_{(3,8)}$	0.14	$W_{(8,3)}$	0.24	$W_{(3,11)}$	0.23	$W_{(11,3)}$	0.22
$W_{(3,12)}$	0.20	$W_{(12,3)}$	0.24	$W_{(3,13)}$	0.25	$W_{(13,3)}$	0.21	$W_{(4,12)}$	0.24	$W_{(12,4)}$	0.18	$W_{(5,7)}$	0.38
$W_{(7,5)}$	0.20	$W_{(5,11)}$	0.53	$W_{(11,5)}$	0.21	$W_{(6,9)}$	0.37	$W_{(9,6)}$	0.29	$W_{(6,15)}$	0.45	$W_{(15,6)}$	0.24
$W_{(7,10)}$	0.30	$W_{(10,7)}$	0.24	$W_{(7,11)}$	0.22	$W_{(11,7)}$	0.23	$W_{(7,13)}$	0.15	$W_{(13,7)}$	0.18	$W_{(7,14)}$	0.14
$W_{(14,7)}$	0.71	$W_{(8,10)}$	0.27	$W_{(10,8)}$	0.24	$W_{(8,13)}$	0.26	$W_{(13,8)}$	0.18	$W_{(9,15)}$	0.34	$W_{(15,9)}$	0.26
$W_{(10,13)}$	0.24	$W_{(13,10)}$	0.33	$W_{(11,12)}$	0.34	$W_{(12,11)}$	0.39						

TABLE II
POWER ALLOCATION IN THE 15-NODE AD HOC NETWORK (MW).

$\mathbf{Q}_{(1,4)}$	35.46 $0.00+0.00i$	$0.00-0.00i$ 35.46	$\mathbf{Q}_{(4,1)}$	21.84 $-0.09-0.02i$	$-0.09+0.02i$ 21.87	$\mathbf{Q}_{(2,9)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82
$\mathbf{Q}_{(9,2)}$	17.23 $-0.09-0.10i$	$-0.09+0.10i$ 17.47	$\mathbf{Q}_{(2,12)}$	24.51 $-0.08+0.08i$	$-0.08-0.08i$ 23.98	$\mathbf{Q}_{(12,2)}$	8.87 $0.00+0.00i$	$0.00-0.00i$ 8.87
$\mathbf{Q}_{(2,15)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82	$\mathbf{Q}_{(15,2)}$	20.58 $-0.01-0.16i$	$-0.01+0.16i$ 20.50	$\mathbf{Q}_{(3,4)}$	8.90 $0.01-0.00i$	$0.01+0.00i$ 8.91
$\mathbf{Q}_{(4,3)}$	11.82 $0.00+0.00i$	$0.000-0.000i$ 11.82	$\mathbf{Q}_{(3,8)}$	7.36 $0.10-0.01i$	$0.10+0.01i$ 7.23	$\mathbf{Q}_{(8,3)}$	12.18 $0.00+0.00i$	$0.00-0.00i$ 12.15
$\mathbf{Q}_{(3,11)}$	11.37 $0.00-0.33i$	$0.00+0.33i$ 12.04	$\mathbf{Q}_{(11,3)}$	10.76 $0.00-0.00i$	$0.00+0.00i$ 10.70	$\mathbf{Q}_{(3,12)}$	9.60 $0.13+0.18i$	$0.13-0.18i$ 10.27
$\mathbf{Q}_{(12,3)}$	11.75 $0.00+0.00i$	$0.00-0.00i$ 11.73	$\mathbf{Q}_{(3,13)}$	12.22 $0.35-0.69i$	$0.35+0.69i$ 12.09	$\mathbf{Q}_{(13,3)}$	11.09 $-0.03+0.05i$	$-0.03-0.05i$ 11.08
$\mathbf{Q}_{(4,12)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82	$\mathbf{Q}_{(12,4)}$	9.22 $0.04+0.09i$	$0.04-0.09i$ 9.25	$\mathbf{Q}_{(5,7)}$	19.62 $-0.01+0.08i$	$-0.01-0.08i$ 19.68
$\mathbf{Q}_{(7,5)}$	9.54 $0.00+0.04i$	$0.00-0.04i$ 9.51	$\mathbf{Q}_{(5,11)}$	26.56 $-0.10+0.14i$	$-0.10-0.14i$ 26.90	$\mathbf{Q}_{(11,5)}$	10.42 $0.01+0.12i$	$0.01-0.12i$ 10.39
$\mathbf{Q}_{(6,9)}$	21.22 $0.00+0.00i$	$0.00-0.00i$ 21.22	$\mathbf{Q}_{(9,6)}$	14.78 $-0.10+0.08i$	$-0.10-0.08i$ 14.55	$\mathbf{Q}_{(6,15)}$	22.84 $0.21-0.14i$	$0.21+0.14i$ 23.24
$\mathbf{Q}_{(15,6)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82	$\mathbf{Q}_{(7,10)}$	14.19 $-0.05+0.09i$	$-0.05-0.09i$ 13.82	$\mathbf{Q}_{(10,7)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82
$\mathbf{Q}_{(7,11)}$	12.48 $1.15+0.43i$	$1.15-0.43i$ 11.21	$\mathbf{Q}_{(11,7)}$	11.83 $-0.25+0.01i$	$-0.25-0.01i$ 12.10	$\mathbf{Q}_{(7,13)}$	7.54 $0.12+0.02i$	$0.12-0.02i$ 7.51
$\mathbf{Q}_{(13,7)}$	8.87 $0.00+0.00i$	$0.00-0.00i$ 8.87	$\mathbf{Q}_{(7,14)}$	7.10 $0.00+0.00i$	$0.00-0.00i$ 7.10	$\mathbf{Q}_{(14,7)}$	39.76 $0.19-0.04i$	$0.19+0.04i$ 39.71
$\mathbf{Q}_{(8,10)}$	15.00 $0.05+0.05i$	$0.05-0.05i$ 14.95	$\mathbf{Q}_{(10,8)}$	11.82 $0.016+0.00i$	$0.0-0.00i$ 11.82	$\mathbf{Q}_{(8,13)}$	12.80 $0.03+0.04i$	$0.03-0.04i$ 12.82
$\mathbf{Q}_{(13,8)}$	10.00 $0.54+0.59i$	$0.54-0.59i$ 9.69	$\mathbf{Q}_{(9,15)}$	17.21 $0.32-0.28i$	$0.32+0.28i$ 16.77	$\mathbf{Q}_{(15,9)}$	13.09 $0.10+0.08i$	$0.10-0.08i$ 12.95
$\mathbf{Q}_{(10,13)}$	11.82 $0.00+0.00i$	$0.00-0.00i$ 11.82	$\mathbf{Q}_{(13,10)}$	15.70 $0.15-0.01i$	$0.15+0.01i$ 15.71	$\mathbf{Q}_{(11,12)}$	16.76 $-0.04-0.02i$	$-0.04+0.02i$ 17.04
$\mathbf{Q}_{(12,11)}$	20.85 $0.20-0.11i$	$0.20+0.11i$ 19.47						

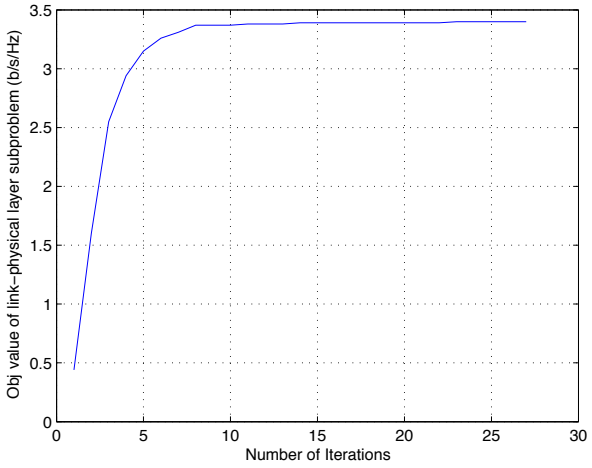

Fig. 5. Convergence behavior of gradient projection algorithms for link-physical layer subproblem (for $\mathbf{u}=\mathbf{1}$).

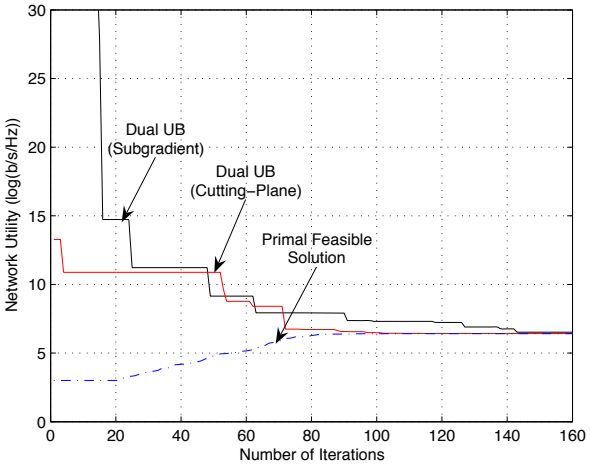

Fig. 6. Convergence behavior of cutting-plane (centralized) and subgradient (distributed) algorithms for the 15-node ad hoc network example.

diversity gain) of MIMO links that can be leveraged by routing layer protocols in rich multi-path environments to improve performance. In [35], Lee *et al*. proposed a distributed algorithm for MIMO-based multi-hop ad hoc networks, where diversity and multiplexing gains of each link are controlled to achieve the optimal rate-reliability trade-off. They assumed fixed SINRs and fixed routes between source and destination nodes. In these efforts, node power control, per-antenna power

allocation and their impact on upper layers are not considered.

It is also worth pointing out that the Lagrangian decomposition framework has also been employed in other networking problems in the literature. For example, in [36], Xiao *et al.* used a similar decomposition technique to solve simultaneous routing and resource allocation problems. However, their routing setting is very different from ours and their link layer is not MIMO-based. Due to the spatial dimension resulted from MIMO, the link layer subproblem in this paper is different and much more challenging.

IX. Conclusion

In this paper, we investigated the problem of cross-layer optimization of routing, power allocation, and bandwidth allocation for MIMO-based ad hoc networks. We developed a mathematical solution procedure, which combines Lagrangian decomposition, gradient projection, cutting-plane, and subgradient methods. We showed the decomposable structure of the Lagrangian dual problem and the details of our proposed algorithms. We also presented a distributed implementation. Our numerical results showed that the performance gain by the optimal cross-layer design is significant.

Acknowledgements

The authors thank the anonymous reviewers for their constructive comments. The work of Y.T. Hou, J. Liu, and Y. Shi has been supported in part by the National Science Foundation (NSF) under Grant CNS-0721421 and Office of Naval Research (ONR) under Grant N00014-08-1-0084. The work of H.D. Sherali has been supported in part by NSF Grant CMMI-0552676.

References

[1] J. Winters, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 5, pp. 871–878, Jun. 1987.

[2] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecomm.*, vol. 10, no. 6, pp. 585–596, Nov. 1999.

[3] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, pp. 311–355, Mar. 1998.

[4] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 281–302, Apr. 2003.

[5] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 1, pp. 684–702, Jun. 2003.

[6] J. H. Winters, "Smart antenna techniques and their application to wireless ad hoc networks," *IEEE Wireless Commun. Mag.*, vol. 13, no. 4, pp. 77–83, Aug. 2006.

[7] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 3rd ed. New York: John Wiley & Sons Inc., 2005.

[8] F. P. Kelly, A. K. Malullo, and D. K. H. Tan, "Rate control in communications networks: Shadow prices, proporotional fairness and stablity," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[9] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. New York, NY: John Wiley & Sons Inc., 2006.

[10] J. B. Rosen, "The gradient projection method for nonlinear programming, Part I, Linear constraints," *SIAM J. Applied Mathematics*, vol. 8, pp. 181–217, 1960.

[11] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2839–2848, Nov. 2003.

[12] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Economics*. New York: Wiley, 1999.

[13] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[14] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Berlin: Springer-Verlag, 2001.

[15] S. Cartreux, L. J. Greenstein, and P. F. Dressen, "Simulation results for an interference-limited multiple-input multiple-output cellular system," *IEEE Commun. Lett.*, vol. 4, no. 11, pp. 334–336, Nov. 2000.

[16] N. Jindal, W. Rhee, S. Vishwanath, S. A. Jafar, and A. Goldsmith, "Sum power iterative water-filling for multi-antenna Gaussian broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1570–1580, Apr. 2005.

[17] T. Lan and W. Yu, "Input optimization for multi-antenna broadcast channels and per-antenna power constraints," in *Proc. IEEE GLOBECOM*, vol. 1, Dallas, TX, U.S.A., Nov. 2004, pp. 420–424.

[18] H. Viswanathan, S. Venkatesan, and H. Huang, "Downlink capacity evaluation of cellular networks with known-interference cancellation," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 802–811, Jun. 2003.

[19] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 461–471, Feb. 2004.

[20] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative water-filling for Gaussian vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 145–152, Jan. 2004.

[21] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, Sep. 2006.

[22] M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[23] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658–2668, Oct. 2003.

[24] P. Viswanath and D. N. C. Tse, "Sum capacity of the vector Gaussian broadcast channel and uplink-downlink duality," *IEEE Trans. Inf. Theory*, vol. 49, no. 8, pp. 1912–1921, Aug. 2003.

[25] W. Yu, "Uplink-downlink duality via minimax duality," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 361–374, Feb. 2006.

[26] W. Yu, "A dual decomposition approach to the sum power Gaussian vector multiple-access channel sum capacity problem," in *Proc. Conf. Information Sciences and Systems (CISS)*, Baltimore, MD, U.S.A., Mar. 2003.

[27] J. Liu, Y. T. Hou, and H. D. Sherali, "Conjugate gradient projection approach for MIMO Gaussian broadcast channels," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, France, Jun.24-29, 2007, pp. 781–785.

[28] E. A. Jorswieck and H. Boche, "Performance analysis of capacity of MIMO systems under multiuser interference based on worst-case noise behavior," *EURASIP Journal on Wireless Communications and Networking*, vol. 2004, no. 2, pp. 273–285, Feb. 2004.

[29] M. F. Demirkol and M. A. Ingram, "Power-controlled capacity for interfering MIMO links," in *Proc. IEEE Veh. Technol. Conf.*, Atlantic City, NJ, U.S.A., Oct. 2001, pp. 187–191.

[30] M. F. Demirkol and M. A. Ingram, "Stream control in network with interfereing MIMO links," in *Proc. IEEE Wireless Commun. and Networking Conf.*, New Orleans, LA, U.S.A., Mar. 2003, pp. 343–348.

[31] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang, "Medium access control in ad hoc networks with MIMO links: Optimization considerations and algorithms," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 350–365, Oct. 2004.

[32] B. Chen and M. J. Gans, "MIMO communications in ad hoc networks," *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2773–2783, Jul. 2006.

[33] M. Hu and J. Zhang, "MIMO ad hoc networks: Medium access control, saturation throughput, and optimal hop distance," *Special Issue on Mobile Ad Hoc Networks, Journal of Communications and Networks*, pp. 317–330, Dec. 2004.

[34] K. Sundaresan and R. Sivakumar, "Routing in ad hoc networks with MIMO links," in *Proc. IEEE International Conf. on Network Protocols*, Boston, MA, U.S.A., Nov. 2005, pp. 85–98.

[35] J.-W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithms for rate-reliability tradeoff in network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 962–976, May 2006.

[36] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136–1144, Jul. 2004.

Jia Liu (S'03) received his B.S. degrees from South China University of Technology in 1996, in Electrical Engineering and Computer Science, respectively. He received his M.S. degree in Electrical Engineering from South China University of Technology in 1999. He is currently working toward his Ph.D. degree in the Bradley Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA. His research focuses on MIMO technologies, multicarrier communications systems, cognitive radio, information theory, and mathematical optimization theory for communications networks.

Prior to joining Virginia Tech, Liu was with Bell Labs, Lucent Technologies in Beijing China for four years. His main responsibilities include participating in 3GPP2 and making contributions to various standards of cdma2000-1x/1xEV-DO/1xEV-DV. He is a co-recipient of Bell Labs President Gold Award in 2001. During summer 2005, he worked as a software engineer intern in Mercury Computer Systems Inc.

Liu is a student member of IEEE. He is a regular reviewer for major IEEE conferences and journals. He is a member of Tau Beta Pi, the national engineering honor society. He is a recipient of IEEE ICC 2008 Best Paper Award

Y. Thomas Hou (S'91–M'98–SM'04) received the B.E. degree from the City College of New York in 1991, the M.S. degree from Columbia University in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1998, all in Electrical Engineering.

Since August 2002, he has been with Virginia Tech, the Bradley Department of Electrical and Computer Engineering, Blacksburg, VA, where he is now an Associate Professor. His current research interests are radio resource (spectrum) management and networking for cognitive radio wireless networks, optimization and algorithm design for wireless ad hoc and sensor networks, and video communications over dynamic ad hoc networks. From 1997 to 2002, Dr. Hou was a Researcher at Fujitsu Laboratories of America, Sunnyvale, CA, where he worked on scalable architectures, protocols, and implementations for differentiated services Internet; service overlay networking; video streaming; network bandwidth allocation policies and distributed flow control algorithms.

Prof. Hou is a recipient of an Office of Naval Research (ONR) Young Investigator Award (2003) and a National Science Foundation (NSF) CAREER Award (2004) for his research on algorithm design and optimizations for wireless ad hoc and sensor networks. He has published extensively in leading journals and top tier conferences and received several best paper awards from IEEE (including IEEE INFOCOM 2008 Best Paper Award).

Prof. Hou is active in professional services and is currently serving as an Editor of *IEEE Transactions on Wireless Communications*, *ACM/Springer Wireless Networks (WINET)*, and Elsevier *Ad Hoc Networks* Journal. He was Co-Chair of Technical Program Committee (TPC) of the Second International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM 2007), Orlando, FL, August 1-3, 2007. He was the Founding Chair of the First IEEE Workshop on Networking Technologies for Software Defined Radio Networks, September 25, 2006, Reston, VA.

Prof. Hou was Co-Chair (with Tony Ephremides) of NSF Workshop on Bridging the Gap between Wireless Networking Technologies and Advances at the Physical Layer, August 27-28, 2007, Reston, VA. He is a TPC Co-Chair of IEEE INFOCOM 2009, to be held in Rio de Janeiro, Brazil.

Yi Shi (S'02–M'08) received his B.S. degree from University of Science and Technology of China, Hefei, China, in 1998, a M.S. degree from Institute of Software, Chinese Academy of Science, Beijing, China, in 2001, a second M.S. degree from Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, in 2003, all in computer science, and a Ph.D. degree in computer engineering from Virginia Tech, in 2007. He is currently a Senior Research Associate in the Department of Electrical and Computer Engineering at Virginia Tech.

His current research focuses on algorithms and optimization for cognitive radio wireless networks, MIMO and cooperative communication networks, sensor networks, and ad hoc networks. His work has appeared in some highly selective international conferences (ACM MobiCom, ACM MobiHoc, and IEEE INFOCOM) and IEEE journals.

While an undergraduate, he was a recipient of Meritorious Award in International Mathematical Contest in Modeling in 1997 and 1998, respectively. He was a recipient of Chinese Government Award for Outstanding Ph.D. Students Abroad in 2006. He is a recipient of IEEE INFOCOM 2008 Best Paper Award.

Dr. Shi is active in professional services. He was a TPC member of IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks (held in conjunction with IEEE SECON 2006), Reston, VA, Sept. 25, 2006, and ChinaCom, Hangzhou, China, April 25–27, 2008. He is a TPC member of ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (co-located with ACM MobiHoc 2008), Hong Kong, China, May 26–30, 2008; IEEE ICCCN, St. Thomas, U.S. Virgin Islands, August 4–7, 2008; IEEE PIMRC, Cannes, France, September 15–18, 2008; IEEE MASS, Atlanta, GA, Oct. 6–9, 2008; and IEEE ICC, Dresden, Germany, June 14–18, 2009.

Hanif D. Sherali is a University Distinguished Professor and the W. Thomas Rice Chaired Professor of Engineering in the Industrial and Systems Engineering Department at Virginia Polytechnic Institute and State University.

His areas of research interest are in analyzing problems and designing algorithms for specially structured linear, nonlinear, and integer programs arising in various applications, global optimization methods for non-convex programming problems, location and transportation theory and applications, economic and energy mathematical modeling and analysis. He has published over 238 refereed articles in various Operations Research journals, has (co-) authored six books in this area, and serves on the editorial board of eight journals.

He is an elected member of the National Academy of Engineering.