

On Throughput Maximization for a Multi-hop MIMO Network

Xiaoqi Qin Xu Yuan Yi Shi Y. Thomas Hou Wenjing Lou Scott F. Midkiff
Virginia Polytechnic Institute and State University, USA

Abstract—There has been a growing interest to employ the so-called degree-of-freedom (DoF) based models to study multi-hop MIMO networks. Existing DoF-based models differ in their interference cancelation (IC) behavior and suffer from either loss of solution space or possible infeasible solutions. Recently, a DoF model based on a novel node-ordering concept was proposed to overcome the limitations of the exiting DoF models. In this paper, we apply this new DoF model to study a throughput maximization problem in a multi-hop network. The problem formulation jointly considers half duplex, node ordering, DoF consumption constraints and flow routing and is in the form of a mixed integer linear program (MILP). Our main contribution is the development of an efficient polynomial time algorithm that offers a competitive solution to the MILP through a series of linear programs (LPs). The key idea in the algorithm is to explore (i) the impact of node ordering on DoF consumption for IC at a node, and (ii) route diversity in the network while ensuring DoF constraints are satisfied at each node throughout the iterations. Simulation results show that our solutions by the proposed algorithm are competitive and feasible.

I. INTRODUCTION

MIMO is widely considered as one of the major breakthroughs in modern wireless communications [3]. Although significant advances have been made on MIMO at the physical layer or for single-hop cellular communications, advances of MIMO for multi-hop networks remain primitive. The main problem in this research stagnation is the lack of tractable and accurate MIMO models that are suitable for analysis by the networking research community.

In recent years, the so-called degree-of-freedom (DoF) based models have become popular to study MIMO multi-hop networks [2], [4], [5], [11], [15]. The concept of DoF was originally defined to represent the multiplexing gain of MIMO channel in the information theory research community [7], [17], [18]. This concept was then extended by the networking research community to characterize a node’s spatial resources provided by its multiple antennas. The main idea of DoF-based models is as follows: (i) The number of available DoFs at a node is equal to the number of its antennas. (ii) A node consumes DoFs for SM. Specifically, a transmit node consumes DoFs to transmit its data streams while a receive node consumes DoFs to receive its desired data streams. (iii) A node consumes DoFs for IC. Specifically, a transmit node may cancel its interference to its neighboring receive nodes by consuming its DoFs; likewise, a receive node may cancel the interference from its unintended transmit nodes by consuming

its DoFs. (iv) A node can use some or all of its DoFs for SM and IC, as long as the total number of DoFs consumed for SM and IC does not exceed its available DoFs. A key benefit of DoF-based models is its simplicity. Here, complex matrix-based representation for SM and IC is avoided by performing simple addition/subtraction on the number of DoFs.

Since its inception, DoF-based modeling for MIMO has evolved into two branches: *conservative models* and *optimistic models*. Although both branches of DoF models share the same accounting of DoF consumption for SM, they differ significantly in DoF consumption for IC. The conservative models tend to lose feasible solutions due to duplication in IC, restriction on receiver-side IC, or some other predefined IC rules. Examples of conservative models include [2], [11], [15]. On the other hand, optimistic models tend to incorrectly enlarge the feasible solution space due to a lack of systematic scheme for IC among the nodes. As a result, an optimistic model may offer solutions that are in fact infeasible. Examples of optimistic DoF-based models include [4], [5].

Very recently, a new DoF-based MIMO model was developed by Shi *et al.* [14] based on a novel “node ordering” concept. The essence of this model is to place an order on each node in the network through an ordered node list so that IC can be performed on a “disciplined” manner at each node. Under this model, a transmit node only needs to cancel its interference to those receive nodes (within its interference range) that are before itself in the ordered node list. It does not need to consume DoFs to cancel its interference to those receive nodes that are after itself in the ordered list. Likewise, a receive node only needs to cancel the interference from those transmit nodes that are before itself in the ordered node list. It does not need to cancel the interference from those transmit nodes that are after itself in the ordered node list. By referencing the position of a node in the ordered node list, duplication in IC (as in conservative model) can be eliminated. Further, it was shown in [14] that the solution by the new DoF model is always feasible in contrast to optimistic models.

Inspired by the new DoF model in [14], we explore how this model can be applied to study a throughput maximization problem in a multi-hop network. Specifically, we consider a set of unicast sessions in a MIMO network. The objective is to maximize the minimum rate among the sessions. Through joint formulation of half-duplex, node ordering, DoF allocation (for SM and IC), and flow routing, we obtain a mixed-integer linear program (MILP), which is NP-hard in general. Although a commercial solver (CPLEX) may solve our problem for

small-sized networks, a more efficient (and polynomial time) algorithm is needed to handle networks of larger sizes.

The main contribution of this paper is an efficient algorithm for the MILP. In essence, the proposed solution is an iterative greedy algorithm that offers a competitive solution through a series (but limited number) of LPs. The algorithm requires an initial feasible solution, which can be found based on the *sequential fixing* (SF) technique [6]. The SF technique is itself based on a series of LPs. For the initial feasible solution, our algorithm first identifies a "bottleneck" link. Subsequently, our algorithm attempts to increase DoF allocation (for SM) on the bottleneck link by altering the ordering of the node list. This idea exploits the unique property of the new DoF model in which a node's DoF consumption for IC depends on its position in the ordered list. Should the bottleneck link cannot be "widened", our algorithm try to find a parallel route between the transmit and receive nodes of the bottleneck link. Throughout the iterations, our algorithm ensures that DoF constraints for SM and IC at each node are satisfied. Simulation results show that our proposed algorithm can offer a competitive solution to the MILP. Further, the solution by our proposed algorithm is feasible.

The remainder of this paper is organized as follows. In Section II, we develop a problem formulation for the throughput optimization problem based on the new DoF model. In Section III, we present an algorithm to solve the throughput optimization problem. Section IV presents simulation results. Section V concludes this paper.

II. MODELING AND FORMULATION

In this section, we employ the DoF model in [14] to study a throughput maximization problem for multi-hop MIMO networks. For a set of sessions in the network, the objective is to maximize the minimum rate among the sessions (transport layer) by optimizing variables in the DoF model (link layer) and flow routing (network layer).

A. Mathematical Modeling

We consider a multi-hop MIMO network with a set of \mathcal{N} nodes, with $N = |\mathcal{N}|$ being the number of nodes. Each MIMO node i has A_i antennas. Denote \mathcal{F} as a set of sessions in the network. For each session $f \in \mathcal{F}$, denote $s(f)$ and $d(f)$ as its source and destination nodes, respectively. Denote $r(f)$ as the achieved throughput of session $f \in \mathcal{F}$.

We assume that time slot based scheduling is used at the link layer. Denote T as the number of time slots in a frame.

Half-duplex Constraint. We assume that wireless transceivers are half-duplex. That is, a node cannot transmit and receive at the same time. In a time slot based system, half-duplex can be implemented by having the transceiver operate in only one mode in a time slot (transmit, receive, idle). To model half-duplex in a time slot with binary variables, we define two binary variables $x_i[t]$ and $y_i[t]$ to indicate whether node i is a transmit node or a receive node in time slot t ,

respectively. That is,

$$x_i[t] = \begin{cases} 1 & \text{if node } i \text{ is a transmit node in time slot } t; \\ 0 & \text{otherwise (either idle or receive).} \end{cases}$$

$$y_i[t] = \begin{cases} 1 & \text{if node } i \text{ is a receive node in time slot } t; \\ 0 & \text{otherwise (either idle or transmit).} \end{cases}$$

Then the half-duplex constraint can be modeled as

$$x_i[t] + y_i[t] \leq 1 \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (1)$$

Constraints on Data Streams. We assume the total number of DoFs at a node i is equal to its number of antenna elements A_i . Denote $z_{ij}(t)$ as the number of data streams from node i to node j in time slot t . If a node i is not an active transmitter in time slot t , then no data stream is transmitted from this node, i.e., $\sum_{j \in \mathcal{T}_i} z_{ij}[t] = 0$ if $x_i[t] = 0$, where \mathcal{T}_i is the set of nodes within the transmission range of node i . Otherwise, the total number of DoFs used for transmission cannot exceed the total number of antennas A_i at this node, i.e., $1 \leq \sum_{j \in \mathcal{T}_i} z_{ij}[t] \leq A_i$ if $x_i[t] = 1$. These two cases can be formulated as

$$x_i[t] \leq \sum_{j \in \mathcal{T}_i} z_{ij}[t] \leq A_i \cdot x_i[t] \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (2)$$

Similarly, considering whether or not node i is a receive node in time slot t , we have

$$y_i[t] \leq \sum_{j \in \mathcal{T}_i} z_{ji}[t] \leq A_i \cdot y_i[t] \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (3)$$

where we assume node j and node i have the same transmission range.

Ordering Constraints. The "ordering" concept was proposed in [14] to avoid duplication in IC among the MIMO nodes in the network. An optimal ordering can be found by incorporating ordering variables in the problem formulation.

Denote $\pi[t]$ as a list with length N , with each element containing a node in the network. The position of the node in the list defines the "order" of that node. Denote $\pi_i[t]$ as the order (position) of node i in $\pi[t]$. For example, if $\pi_i[t] = 5$, then it means that node i is in the fifth element in the ordered list. Therefore, we have

$$1 \leq \pi_i[t] \leq N \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (4)$$

We use a binary variable $\theta_{ji}[t]$ to indicate the relative position between two nodes i and j in $\pi[t]$. $\theta_{ji}[t]$ is defined as:

$$\theta_{ji}[t] = \begin{cases} 1 & \text{if node } j \text{ is before node } i \text{ in } \pi[t]; \\ 0 & \text{otherwise.} \end{cases}$$

Based on the definition of $\theta_{ji}[t]$, it can be easily verified that the following relationships hold between $\pi_i[t]$ and $\pi_j[t]$:

$$\pi_i[t] - N \cdot \theta_{ji}[t] + 1 \leq \pi_j[t] \leq \pi_i[t] - N \cdot \theta_{ji}[t] + N - 1 \quad (i, j \in \mathcal{N}, i \neq j, 1 \leq t \leq T). \quad (5)$$

DoF Consumption Constraints. A node can use its DoFs for either SM or IC, as long as the number of consumed DoFs does not exceed its total available DoFs. Depending on

whether the node is a transmit or receive node in time slot t , it has different IC behavior as follows:

- If the node is a transmit node, in addition to SM, the node should use its DoFs to cancel its interference to all the unintended receive nodes (within its interference range) that are before itself in the ordered node list $\pi[t]$. Further, the number of DoFs consumed by this transmit node for IC is equal to the sum of intended data streams received by those unintended receive nodes.
- If the node is a receive node, in addition to SM, the node should use its DoFs to cancel interference from all unintended transmit nodes (whose interference range covers this receive node) that are before itself in the ordered node list $\pi[t]$. Further, the number of DoFs consumed by this node for IC is equal to the sum of data streams transmitted by those unintended transmit nodes.

Note that in either case (transmitter or receiver), the node only needs to consider the nodes before itself in $\pi[t]$ for IC. Interference to/from nodes that are after the current node in the ordered node list $\pi[t]$ will be taken care of by those nodes later. This is the key in the new DoF model for IC.

We now model both cases mathematically as follows:

$$\text{If } x_i[t] = 1, \text{ then } \sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right) \leq A_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (6)$$

where on the left side of the inequality, the first and second terms represent the number of DoFs consumed by node i for SM and IC, respectively.

$$\text{If } y_i[t] = 1, \text{ then } \sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right) \leq A_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (7)$$

where on the left side of the inequality, the first and second terms represent the number of DoFs consumed by node i for SM and IC, respectively.

Flow Balance Constraints. For flexibility and better load balancing, we allow flow splitting in the network. That is, the flow of a session may split and merge inside the network in whatever manner as long as it can help to achieve a high data rate. Denote $r_{ij}(f)$ as the data rate on link (i, j) that is attributed to session $f \in \mathcal{F}$, where $i \in \mathcal{N}$ and $j \in \mathcal{T}_i$. Then we have the following flow balance constraints:

- If node i is the source node of session f (i.e., $i = s(f)$), then

$$\sum_{j \in \mathcal{T}_i} r_{ij}(f) = r(f) \quad (f \in \mathcal{F}). \quad (8)$$

- If node i is an intermediate relay node for session f (i.e., $i \neq s(f), i \neq d(f)$), then

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(f)} r_{ij}(f) = \sum_{k \in \mathcal{T}_i}^{k \neq d(f)} r_{ki}(f) \quad (f \in \mathcal{F}, i \in \mathcal{N}). \quad (9)$$

- If node i is the destination node for session f (i.e., $i = d(f)$), then

$$\sum_{j \in \mathcal{T}_i} r_{ji}(f) = r(f) \quad (f \in \mathcal{F}). \quad (10)$$

It can be easily verified that once (8) and (9) are satisfied, (10) must also be satisfied. As a result, it is sufficient to just include (8) and (9) in the formulation.

Link Capacity Constraints. For each link (i, j) , the sum of the data rates over all sessions that traverse this link cannot exceed the average data rate on this link. For simplicity, we assume that fixed modulation and coding scheme (MCS) is used for each data stream and that each data stream corresponds to one unit data rate. Then on each link (i, j) , we have:

$$\sum_{f \in \mathcal{F}} r_{ij}(f) \leq \frac{1}{T} \sum_{t=1}^T z_{ij}[t] \quad (i \in \mathcal{N}, j \in \mathcal{T}_i), \quad (11)$$

where T is the number of time slots in a frame.

B. Formulation

Based on the mathematical models for the link and network layers, we study a throughput maximization problem (at transport layer) in multi-hop MIMO networks. There are various objectives that can be considered. In this paper, we employ the objective of maximizing the minimum data rate among the sessions. Denote r_{\min} as the minimum throughput among all sessions. The problem can be formulated as follows:

OPT-raw

$$\begin{aligned} \max \quad & r_{\min} \\ \text{s.t} \quad & r_{\min} \leq r(f), \quad f \in \mathcal{F}; \\ & \text{half duplex constraints: (1);} \\ & \text{constraints on data streams: (2), (3);} \\ & \text{node ordering constraints: (4), (5);} \\ & \text{DoF consumption for transmitters: (6);} \\ & \text{DoF consumption for receivers: (7);} \\ & \text{flow balance constraints: (8), (9);} \\ & \text{link capacity constraints: (11).} \end{aligned}$$

In this formulation, note that the two sets of constraints in (6) and (7) are stated in the form of sufficient conditions rather than in the form of mathematical programming. Therefore, a reformulation of (6) and (7) is needed.

For constraint (6), if $x_i[t] = 1$, then we have $\sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right) \leq A_i$. On the other hand, if $x_i[t] = 0$, then no DoF is consumed. Constraint (6) can be reformulated by incorporating binary variable $x_i[t]$ into the expression as follows:

$$\sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right) \leq A_i x_i[t] + (1 - x_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (12)$$

where $B_i = \sum_{j \in \mathcal{I}_i} A_j$ is an upper bound of $\sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right)$.

Similarly, constraint (7) can be reformulated as follows:

$$\sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} \left(\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] \right) \leq A_i y_i[t] + (1 - y_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (13)$$

Note that constraints (12) and (13) have nonlinear terms $\sum_{j \in \mathcal{I}_i} (\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t])$ and $\sum_{j \in \mathcal{I}_i} (\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t])$, respectively. To remove the nonlinear terms in the formulation, we employ the *Reformulated-Linearization Technique (RLT)* [13]. For constraint (6), we introduce a new variable $\lambda_{ji}[t] = \theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t]$. Then constraint (6) can be replaced by the following linear constraint:

$$\sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \lambda_{ji}[t] \leq A_i x_i[t] + (1 - x_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (14)$$

Now we need to add constraints for $\lambda_{ji}[t]$. Since $\theta_{ji}[t]$ is a binary variable and $0 \leq \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \leq A_j$, we can use the following three constraints to specify $\lambda_{ji}[t]$:

$$\lambda_{ji}[t] \leq \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T) \quad (15)$$

$$\lambda_{ji}[t] \leq A_j \cdot \theta_{ji}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T) \quad (16)$$

$$\lambda_{ji}[t] \geq A_j \cdot \theta_{ji}[t] + \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] - A_j \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T). \quad (17)$$

Similarly, for constraint (7), we introduce a new variable $u_{ji}[t] = \theta_{ji}[t] \cdot \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t]$. Then, constraint (7) can be replaced by the following set of linear constraints:

$$\sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} u_{ji}[t] \leq A_i y_i[t] + (1 - y_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (18)$$

$$u_{ji}[t] \leq \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T), \quad (19)$$

$$u_{ji}[t] \leq A_j \cdot \theta_{ji}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T), \quad (20)$$

$$u_{ji}[t] \geq A_j \cdot \theta_{ji}[t] + \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] - A_j \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T). \quad (21)$$

The problem is now reformulated as follows:

OPT

max r_{\min}

s.t $r_{\min} \leq r(f) \quad f \in \mathcal{F};$

half duplex constraints: (1);

constraints on data streams: (2), (3);

node ordering constraints: (4), (5);

DoF consumption for transmitters: (14)-(17);

DoF consumption for receivers: (18)-(21);

flow balance constraints: (8), (9);

link capacity constraints: (11).

In this formulation, A_i and B_i are constants, $x_i[t]$, $y_i[t]$, $z_{ij}[t]$, $\pi_i[t]$ and $\theta_{ji}[t]$ are integer variables, and r_{\min} , $r(f)$, $\lambda_{ji}[t]$ and $\mu_{ji}[t]$ are continuous variables. This optimization problem is in the form of a mixed-integer linear program (MILP), which is NP-hard in general. It is not solvable via commercial solvers even for moderate sized networks. Thus, we need to develop a competitive and efficient algorithm to find a solution.

III. AN EFFICIENT SOLUTION PROCEDURE

A. Overview

The proposed solution is an iterative greedy algorithm. To start the algorithm, an initial feasible solution to OPT is needed. There are many ways to obtain an initial feasible solution. But we recommend to use the so-called *sequential fixing* (SF) technique in [6], which can offer an excellent initial feasible solution.

Figure 1 shows the flow chart of the proposed algorithm. The initial feasible solution includes a set of feasible values for $\mathbf{x}[t]$, $\mathbf{y}[t]$, $\mathbf{z}[t]$, and $\boldsymbol{\theta}[t]$, where $\mathbf{x}[t]$ and $\mathbf{y}[t]$ represent the vectors $[x_1[t], x_2[t], \dots, x_N[t]]$ and $[y_1[t], y_2[t], \dots, y_N[t]]$. $\mathbf{z}[t]$ and $\boldsymbol{\theta}[t]$ represent matrices $[z_{ij}[t]]_{N \times N}$ and $[\theta_{ji}[t]]_{N \times N}$. Based on these values, in Step 1 of the algorithm, we find the current r_{\min} and identify the session with this bottleneck rate. For the session with this minimum rate,¹ we identify the link associated with this bottleneck rate among all links traversed by this session.²

In Step 2, we try to enlarge the ‘‘pipe’’ of the bottleneck link by increasing the number of data streams on this link. This is done by examining: (i) at the transmit and receive nodes of this link, if there is any remaining DoF (in any time slot); and (ii) for nodes that are behind these two nodes in the ordered node list, if there are enough remaining DoFs for IC should a new data stream is added to the bottleneck link.

If Step 2 is not successful, then in Step 3, we try to see if any change in node ordering can change the DoF consumption at a node. This step is motivated by the fact that the relative position of a node in the ordered node list affects its IC behavior and DoF consumption.

If Step 3 is not successful, we try to modify the current routes for the bottleneck session. The design space here is large and will have different complexity and performance trade-off.

¹In case of a tie, we choose the session with the smallest session number (index). This will eliminate any randomness in choosing a session.

²In case of a tie, we may randomly choose one from the bottleneck links.

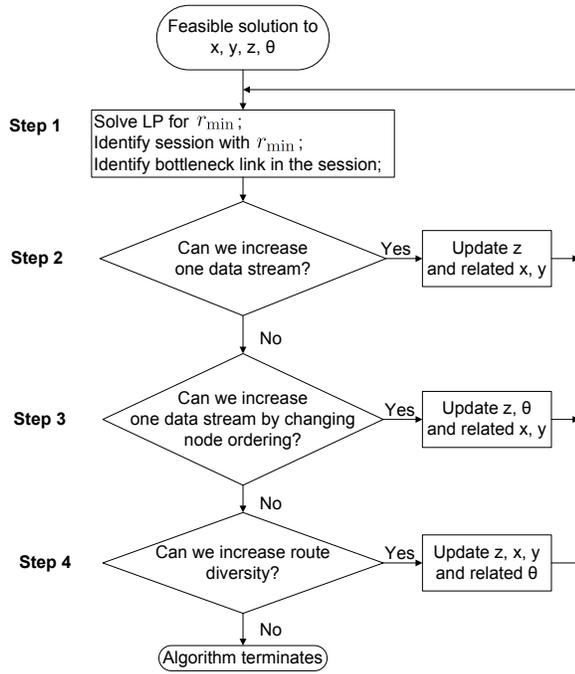


Fig. 1. Flow chart for the proposed algorithm.

As an illustration, we propose a simple algorithm in this step. Specifically, for the underlying bottleneck link (which already failed Steps 2 and 3), we check if we can find a nearby relay node so that we can add a new route in parallel to this link.

The algorithm will continue its iteration as long as any of Steps 2 to 4 is successful, in which case we update $\mathbf{z}[t]$ and corresponding $\mathbf{x}[t]$, $\mathbf{y}[t]$ and $\boldsymbol{\theta}[t]$ variables as needed. The algorithm terminates when none of Steps 2 to 4 is successful. In the rest of this section, we give details of these four steps.

B. Details

In this section, we give detailed descriptions of each step in the algorithm.

Step 1: Finding bottleneck link. In this step, we are given a feasible solution which gives values for $\mathbf{x}[t]$, $\mathbf{y}[t]$, $\mathbf{z}[t]$ and $\boldsymbol{\theta}[t]$. With these fixed values for $\mathbf{x}[t]$, $\mathbf{y}[t]$, $\mathbf{z}[t]$, problem OPT degenerates into a LP. By solving this LP, we can find the current r_{\min} . Subsequently, we identify a session with rate r_{\min} , which we denote f . A tie is broken by choosing the session with the smallest session number (index). Such deterministic tie-breaking mechanism ensures that we keep working on the same session before moving on to the next one. For the chosen session, we find a bottleneck link, i.e., constraint (11) is binding. A tie among multiple bottleneck links may be broken arbitrarily.

Step 2: Adding a data stream. Denote (i, j) as the bottleneck link that we have identified for session f in Step 1, where i is the transmit node and j is the receive node. In Step 2, we try to increase one data stream on this link in some time slot over a frame. This increment is a successful one if the following conditions are satisfied:

- **(C-1):** Both transmit and receive nodes i and j have at least one remaining DoF.
- **(C-2):** For receive nodes after i (and within i 's interference range) in the ordered node list, there is at least one DoF available for IC. Likewise, for transmit nodes after j (and have j in their interference range) in the ordered node list, there is at least one DoF available for IC.

In the case when the above increment is successful, then $z_{ij}[t]$ is incremented by 1, and $x_i[t]$ and $y_j[t]$ are updated to 1 if otherwise. Since we have T time slots in a frame, we will check each time slot in Step 2.

Step 3: Adjusting node ordering. Step 2 will fail if condition (C-1) or (C-2) cannot be satisfied in the same time slot. Note that the ordering of nodes has a profound impact on each node's DoF consumption for IC. Therefore, in Step 3, we will try to adjust the node ordering $\pi[t]$ in each time slot to see if both conditions (C-1) and (C-2) can be satisfied. We propose a two-phase ordering change, denoted (A-1) and (A-2), to address the requirements in (C-1) and (C-2), respectively. **(A-1):** Since condition (C-1) is not satisfied, we have that either transmit node i or receive node j does not have any remaining DoF. We consider the case for transmit node i first. The case for receive node j is similar.

The ordered list L_1 in Fig. 2 shows the current ordering of nodes in the network, in which we have shown the position of transmit node i in this order as well as those nodes (i.e., p , m and k in this example) that are receive nodes before node i in L_1 and are within node i 's interference range. We do not identify other receive nodes (except p , m and k) before node i in L_1 because they are outside the interference range of i . Among receive nodes p , m and k , k is closest to node i in L_1 . Our idea of adjusting node ordering for node i is as follows. Since transmit node i has run out of DoFs, it is likely that it is using some of its DoFs for IC to nodes p , m , and k . If we could move node i before one of these nodes, then the IC burden on node i will be reduced, allowing some DoFs to be freed up for SM of one more data stream. The outcome of such reordering (successful or not) depends on the DoF consumption on each node (after reordering) and whether the DoF constraints (6) and (7) can be met. There are many ways to move up node i (before p , m or k) in the ordered list. In the following, we present an algorithm that we have designed for this purpose.

To reduce the IC burden on node i , we want to choose a receive node (i.e., p , m and k in this example) and put it after node i . This move will add IC burden on the chosen node (p , m or k) as it will be responsible for IC for more transmit nodes (transmit nodes that are among the nodes between itself and node i in L_1). To reduce the number of these new transmit nodes for IC, we first move node i to position $\pi_k[t] + 1$. This will cause the set of nodes between node k and node i in L_1 to be shifted to the right by one position, as shown in L_2 in Fig. 2. Note that this operation will not change DoF consumption at any node in the network and DoF constraints at all nodes remain satisfied.

Now we need to choose a receive node among p , m and k

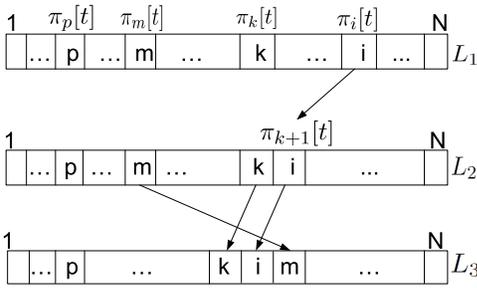


Fig. 2. A schematic illustrating adjusting node ordering.

in L_2 and move it after node i . A receive node is eligible for selection if it has enough DoFs available to cancel interference from all the interfering transmit nodes before itself in the node list after it is moved behind node i . We check nodes p , m , and k individually for its eligibility. Among the eligible nodes,³ we choose the one (say m) that has the largest remaining DoFs after this move.⁴ L_3 in Fig. 2 shows the ordered node list after m is moved after node i , where all the nodes between node m and i in L_2 have been shifted by one position to the left. In L_3 , transmit node i is before receive node m and is no longer responsible for canceling its interference to node m . As a result, node i now has at least one DoF available and can use it for SM of one more data stream on the bottleneck link (i, j) .

For receiver node j , it is not hard to see that a similar approach can be applied to increase its available DoFs. To conserve space, we omit its discussion.

(A-2): Since condition (C-2) is not satisfied, we know that either (i) for a receive node after transmit node i (and within i 's interference range) in the ordered node list, there is no DoF left for IC; or (ii) for a transmit node after receive node j (and has j in its interference range) in the ordered node list, there is no DoF left for IC.

For (i), denote h as such a receive node. Then we will try to change the order for node h so that it will have at least one DoF available. But this is precisely the same reordering problem that we would have done for receive node j in (A-1). Therefore, the same node reordering procedure can be applied to node h . For (ii), again the reordering problem is precisely the same as that for transmit node i in (A-1) and therefore the same node reordering procedure for i can be applied.

Both (A-1) and (A-2) are performed in each time slot until a data stream can be added or they fail in all time slots.

Step 4: Improving route diversity. In Step 2 and 3, we try to increase one data stream on the bottleneck link (i, j) . When both steps fail, it suggests that it may be futile to add one more data stream on this bottleneck link (i, j) . A plausible approach is to open up some other routes (i.e., multiple parallel paths) between nodes i and j so that the extra data stream can be diverted over the new path. There has been extensive research on finding multiple paths between two nodes [9], [10] and the design space is large. For the purpose of this paper, we show

³When there is no eligible node, we move on to the next time slot.

⁴In case of a tie, we choose the node with the smallest node index.

one simple algorithm that only employs one extra relay node to create a second path between nodes i and j .

A node k can be considered as a relay node only if k can serve as node i 's receive node in one time slot and transmit node of node j in a different time slot. For k , we need to check whether both links (i, k) and (k, j) can support one more data stream. For either link (i, k) , or link (k, j) , we are addressing the same problem for link (i, j) in Steps 2 and 3. Therefore, our algorithms in Steps 2 and 3 can be applied here. If both links (i, k) and (k, j) can support one more data stream, then we update $\mathbf{z}[t]$, $\mathbf{x}[t]$, $\mathbf{y}[t]$ and $\boldsymbol{\theta}[t]$, and return to Step 1. Otherwise, the algorithm terminates.

Lemma 1: A solution following the successful outcome of Step 2, 3, or 4 is feasible.

Proof: The feasibility of the solution following the successful exit of Step 2, 3, or 4 can be verified by checking whether the DoF constraints (6) and (7) are satisfied at each node. Specifically, during Step 2, one more data stream can be added to the bottleneck link only if (C-1) and (C-2) are satisfied. If (C-1) and (C-2) are satisfied, then the DoF constraints (6) and (7) must remain satisfied at each node after the extra data stream is added to the bottleneck link. Therefore, if Step 2 is successful, then the DoF constraints (6) and (7) must be satisfied at each node.

If Step 2 fails, it indicates that either (C-1) or (C-2) cannot be satisfied under current node ordering. Then in Step 3, we try to alter the node ordering by using (A-1) and (A-2). (A-1) and (A-2) address the requirements in (C-1) and (C-2), respectively. If operations in Step 3 are successful, (C-1) and (C-2) are satisfied and therefore the DoF constraints (6) and (7) must remain satisfied at each node.

If Step 3 fails, it indicates that either (C-1) or (C-2) cannot be satisfied. Then in Step 4, we try to employ a relay node to create a second path between the transmit node and receive node of the bottleneck link. A node can be chosen as a relay node only if constraint (1) is satisfied at this node. Then the same algorithms in Step 2 and Step 3 are applied to these two new links. Thus, Step 4 is successful only if (C-1) and (C-2) are satisfied and therefore the DoF constraints (6) and (7) must remain satisfied at each node.

Moreover, after each iteration, the number of data streams on each link either remains the same or is increased by one. Therefore, the current r_{\min} can still be supported. ■

C. Complexity

Referring to Fig. 1, the complexity of obtaining an initial feasible solution depends on the specific algorithm. We recommend to use the SF technique [6]. An algorithm based on this technique to find an initial feasible solution is given in [12], which has a computational complexity of $O(N^2V^3)$, where V is the number of variables. In [12], we showed that $V = O(N^2 \cdot \max\{T, |\mathcal{F}|\})$.

We now analyze the complexity of the main algorithm in Fig. 1, which includes the number of iterations and the complexity of each iteration.

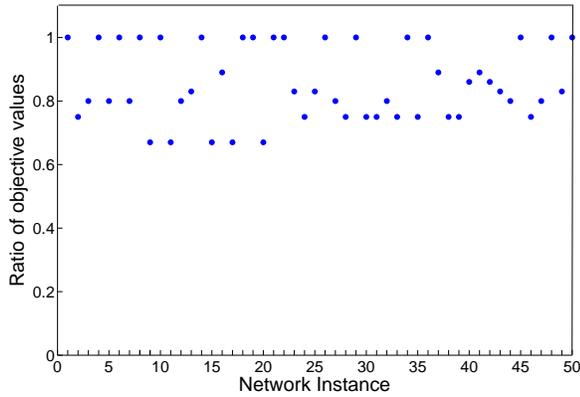


Fig. 3. Ratio between objective values from our algorithm and those from CPLEX for 50 network instances.

In each iteration, the complexity involves solving an LP and identifying a bottleneck link (Step 1), increasing data streams on this link (Step 2 and 3) and adding a parallel path for this link (Step 4). The complexity of solving an LP is $O(V^3)$ [1]. It is not hard to see that solving an LP is the dominant component of complexity among the four steps in an iteration. Therefore, the complexity of an iteration is $O(V^3)$.

We now analyze the total number of iterations. Since the links in the network is upper bounded by $O(N^2)$, and for each link, we can increase its data streams by at most A_i times in each time slot. Therefore, the total number of iterations is $O(N^2 AT)$, where $A = \max_{i \in \mathcal{N}} \{A_i\}$. As a result, the main algorithm has an overall complexity of $O(N^2 AT \cdot V^3)$.

In summary, our algorithm has a complexity of $O(N^2 V^3) + O(N^2 AT \cdot V^3) = O(N^2 AT \cdot V^3)$.

IV. SIMULATION RESULTS

In this section, we use simulation results to demonstrate the performance of our proposed algorithm. We also use a case study to demonstrate the feasibility of a final solution.

Simulation Setting. We consider a multi-hop ad hoc network, with nodes being randomly deployed in a 100×100 area. For scalability, we normalize all units for distance, time, bandwidth, and data rate with appropriate dimensions. We assume that each node in the network is equipped with four antennas and that a node's transmission and interference ranges are 30 and 50, respectively. There are four time slots in a time frame.

We present our results for 50 random network instances, each with 20 nodes and 2 sessions. The source and destination nodes of each session are randomly selected in the network.

Results. For 20-node networks, a commercial solver such as CPLEX is able to find an optimal solution. So we compare the performance of our proposed algorithm to the results from CPLEX. Fig. 3 shows the ratio between the objective values (r_{\min}) from our algorithm and those from CPLEX for 50 network instances. We find that the average ratio is 85.6%, with standard deviation of 0.12.

To see more details, we pick one network instance (the 8-th) from the 50 network instances and show the results. Fig. 4

shows the locations of the 20 nodes, where $s(1)$, $d(1)$, $s(2)$ and $d(2)$ represent the source and destination nodes for sessions 1 and 2, respectively. The objective values (r_{\min}) found by our algorithm and CPLEX are both 0.75, indicating the optimality of our solution for this network instance.

Although the two objective values by our algorithm and CPLEX coincide for this network instance, the flow routing and scheduling behavior under the two solutions are different. Figures 5 and 6 show the routing topologies and scheduling for each session by our algorithm and CPLEX, respectively, where the tuple next to each link represents the time slot index of a frame in which the number of data streams are transmitted. For example, in Fig. 5, (1, 3) next to link (N_2, N_3) denotes that in time slot 1, there are 3 data streams on this link. In the case when there are multiple such tuples next to a link, it means that this link is active over multiple time slots in a frame.

Now let's focus on our solution for this network instance. The node ordering in each time slot is as follows:

- Time slot 1: $\{N_{19}, N_3, N_{11}, N_8, N_{12}, N_1, N_{15}, N_5, N_{20}, N_{16}, N_6, N_2, N_4, N_7, N_9, N_{10}, N_{13}, N_{14}, N_{17}, N_{18}\}$.
- Time slot 2: $\{N_{13}, N_1, N_3, N_8, N_{20}, N_{19}, N_6, N_2, N_4, N_5, N_7, N_9, N_{10}, N_{11}, N_{12}, N_{14}, N_{15}, N_{16}, N_{17}, N_{18}\}$.
- Time slot 3: $\{N_1, N_3, N_{11}, N_4, N_5, N_{20}, N_{18}, N_{12}, N_9, N_6, N_7, N_{10}, N_2, N_8, N_{13}, N_{14}, N_{15}, N_{16}, N_{17}, N_{19}\}$.
- Time slot 4: $\{N_5, N_8, N_{11}, N_8, N_3, N_{20}, N_{17}, N_{16}, N_{15}, N_7, N_4, N_6, N_{13}, N_1, N_2, N_9, N_{10}, N_{12}, N_{14}, N_{19}\}$.

Table I shows the set of active nodes in each time slot and the DoF allocation for SM and IC at these nodes. As an example, consider the set of active nodes in time slot 1 in Table I, which is shown in Fig. 7. The interference relationships among these transmit and receive nodes are shown by the dashed arrows, i.e., node N_2 interferes N_{19} , node N_9 interferes nodes N_3 and N_{13} , and node N_{18} interferes N_{19} . It can be easily verified that by following the relative ordering of these 7 nodes in time slot 1, i.e., $N_{19}, N_3, N_2, N_9, N_{13}, N_{14}, N_{18}$, the DoF constraints in (6) and (7) are satisfied at each of these 7 nodes.

Now we discuss the details of DoF consumption at each active node following the order of their positions in the ordered list in time slot 1.

- The first active node in the ordered node list is N_{19} . It is a receive node. For SM, N_{19} consumes one DoF to receive one data stream from N_9 . For IC, since it is the first node in the node list, it does not consume any DoF for IC.
- The second active node in the ordered node list is N_3 . It is a receive node. For SM, N_3 consumes three DoFs to receive three data streams from N_2 . For IC, since there is no active transmit node before N_3 in the node list, it does not consume any DoF for IC.
- The third active node in the ordered node list is N_2 . It is a transmit node. For SM, N_2 consumes three DoFs to transmit three data streams to node N_3 . For IC, only receive node N_{19} is within its interference range and

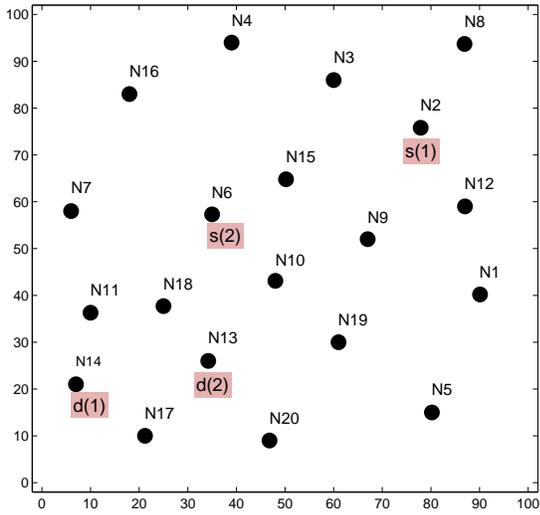


Fig. 4. An instance of 20-node network.

before it in the ordered node list. So N_2 needs to consume one DoF to cancel its interference to node N_{19} .

- The fourth active node in the ordered node list is N_9 . It is a transmit node. For SM, N_9 consumes one DoF to transmit one data stream to node N_{19} . For IC, only transmit node N_3 is within its interference range and before it in the ordered node list. So N_9 needs to consume three DoFs to cancel its interference to node N_3 .
- The fifth active node in the ordered node list is N_{13} . It is a receive node. For SM, N_{13} consumes two DoFs to receive two data streams from node N_{18} . For IC, only transmit node N_9 is within its interference range and before it in the ordered node list. So N_{13} needs to consume one DoF to cancel its interference to node N_9 .
- The sixth active node in the ordered node list is N_{14} . It is a receive node. For SM, N_{14} consumes one DoF to receive one data stream from node N_{18} . For IC, since there is no active transmit node before N_{14} in the ordered node list, it does not consume any DoF for IC.
- The seventh active node in the ordered node list is N_{18} . It is a transmit node. For SM, N_{18} consumes two DoFs to transmit two data streams to node N_{13} and one DoF to transmit one data stream to node N_{14} . For IC, only receive node N_{19} is within its interference range and before it in the ordered node list. So node N_{18} needs to consume one DoF to cancel its interference to node N_{19} .

Note that the DoF constraints for SM and IC at each node are satisfied in time slot 1. Based on Table I, the readers can easily verify that the DoF constraints at each node are also satisfied in time slot 2, 3, and 4 per its respective node ordering in the corresponding time slot.

When the network size becomes large, CPLEX is no longer able to compute a solution in a reasonable amount of time. On the other hand, the proposed algorithm is polynomial and is able to offer a solution very quickly. Readers are referred to

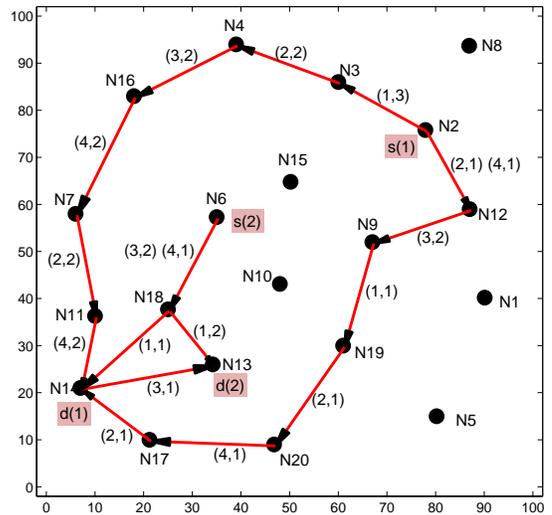


Fig. 5. Solution for flow routing and scheduling by our algorithm.

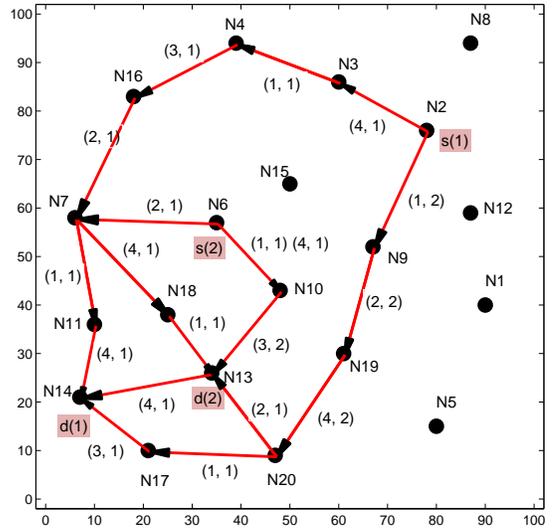


Fig. 6. Solution for flow routing and scheduling by CPLEX.

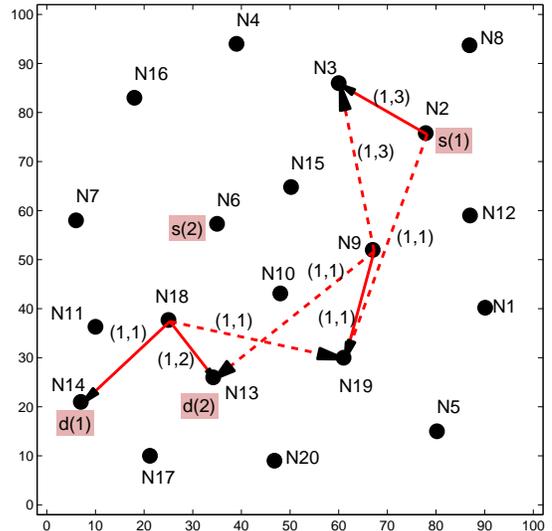


Fig. 7. Scheduling in time slot 1 by our algorithm.

TABLE I
DoF ALLOCATION AT EACH ACTIVE NODE IN EACH TIME SLOT FOR A
20-NODE NETWORK INSTANCE.

Time Slot 1				
Node Ordering	Active Node	Node Status	DoF for SM	DoF for IC
1	N_{19}	receive	1	0
2	N_3	receive	3	0
12	N_2	transmit	3	1
15	N_9	transmit	1	3
17	N_{13}	receive	2	1
18	N_{14}	receive	1	0
20	N_{18}	transmit	3	1
Time Slot 2				
Node Ordering	Active Node	Node Status	DoF for SM	DoF for IC
3	N_3	transmit	2	0
5	N_{20}	receive	1	0
6	N_{19}	transmit	1	0
8	N_2	transmit	1	0
9	N_4	receive	2	1
11	N_7	transmit	2	2
14	N_{11}	receive	2	0
15	N_{12}	receive	1	3
16	N_{14}	receive	1	2
19	N_{17}	transmit	1	3
Time Slot 3				
Node Ordering	Node ID	Node Status	DoF for SM	DoF for IC
4	N_4	transmit	2	0
7	N_{18}	receive	2	0
8	N_{12}	transmit	2	0
9	N_9	receive	2	0
10	N_6	transmit	2	2
15	N_{13}	receive	1	2
16	N_{14}	transmit	1	2
18	N_{16}	receive	2	2
Time Slot 4				
Node Ordering	Active Node	Node Status	DoF for SM	DoF for IC
3	N_{11}	transmit	2	0
4	N_{18}	receive	1	0
6	N_{20}	transmit	1	1
7	N_{17}	receive	1	2
8	N_{16}	transmit	2	1
10	N_7	receive	2	2
12	N_6	transmit	1	3
15	N_2	transmit	1	0
18	N_{12}	receive	1	0
19	N_{14}	receive	2	2

see [12] for additional results by our algorithm for large sized networks.

V. CONCLUSIONS

In this paper, we studied a throughput maximization problem in a multi-hop MIMO network. The foundation of this study is a new DoF model based on a novel node ordering concept. We formulated a throughput maximization problem by taking into consideration of half duplex, node ordering, DoF constraints and flow routing. Since the problem formulation is in the form of MILP, we proposed a polynomial time solution. Some highlights of our proposed solution include

(i) it offers a feasible solution to the MILP through a series of LPs, (ii) it exploits the impact of a node's ordering on its DoF consumption for IC, and (iii) it exploits route diversity in the network to circumvent bottleneck between two nodes. Simulation results show that the solutions by our proposed algorithm are competitive and feasible.

ACKNOWLEDGMENTS

This research was supported in part by NSF Grants 0831865, 1064953, 1102013, 1247830, 1156318, 1156311, and ONR Grant N000141310080.

REFERENCES

- [1] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, *Linear Programming and Network Flows*, fourth edition, Chapter 8, John Wiley & Sons, 2010.
- [2] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proc. IEEE INFOCOM*, pp. 2326–2330, Anchorage, AK, May 6–12, 2007.
- [3] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H.V. Poor, *MIMO Wireless Communications*, Cambridge University Press, Jan. 2007.
- [4] D.M. Blough, G. Resta, P. Santi, R. Srinivasan, and L.M. Cortes-Pena, "Optimal one-shot scheduling for MIMO networks," in *Proc. IEEE SECON*, pp. 377–385, Salt Lake City, Utah, June 2011.
- [5] B. Hamdaoui and K.G. Shin, "Characterization and analysis of multi-hop wireless MIMO network throughput," in *Proc. ACM MobiHoc*, pp. 120–129, Montreal, Quebec, Canada, Sep. 9–14, 2007.
- [6] Y.T. Hou, Y. Shi, and H.D. Sherali, *Applied optimization methods for wireless networks*, Chapter 10, Cambridge University Press, 2014.
- [7] S.A. Jafar and M. Fakhreddin, "Degrees of freedom for the MIMO interference channel," *IEEE Trans. on Information Theory*, vol. 53, no. 7, pp. 2637–2642, July 2007.
- [8] B. Mumei, J. Tang, and T. Hahn, "Algorithmic aspects of communications in multihop wireless networks with MIMO Links," in *Proc. IEEE ICC*, 6 pages, Cape Town, South Africa, July 2010.
- [9] S. Murthy and J. J. Garcia-Luna-Aceves, "Congestion-oriented shortest multi-path routing," in *Proc. IEEE INFOCOM*, pp. 1038–1036, San Francisco, CA, May 1996.
- [10] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proc. ACM MobiHoc*, pp. 1–11, Lausanne, Switzerland, Jun. 2002.
- [11] J-S. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: Enabling spatial reuse using MIMO channel-aware MAC," in *Proc. IEEE ICC*, pp. 3642–3646, Seoul, Korea, May 16–20, 2005.
- [12] X. Qin, X. Yuan, Y. Shi, Y.T. Hou, W. Lou, and S.F. Midkiff, "On throughput maximization for a multi-hop MIMO Network," Technical Report, the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, August 2013.
- [13] H.D. Sherali and W.P. Adams, *A Reformulation-Linearization Technique for solving discrete and continuous nonconvex problems*, Chapter 8, Kluwer Academic Publishers, 1999.
- [14] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y.T. Hou, "An optimal MIMO link model for multi-hop wireless networks," in *Proc. IEEE INFOCOM*, pp. 1916–1924, Shanghai, China, April 10–15, 2011.
- [15] K. Sundaresan, R. Sivakumar, M. Ingram, and T-Y. Chang, "Medium access control in ad hoc networks with MIMO links: Optimization considerations and algorithms," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 350–365, Oct. 2004.
- [16] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, Cambridge, UK, 2005.
- [17] L. Zheng and D.N.C. Tse, "Communication on the Grassmann Manifold: A Geometric Approach to the Noncoherent Multiple-Antenna Channel," *IEEE Trans. on Information Theory*, vol. 48, pp. 359–383, Feb. 2002.
- [18] L. Zheng and D.N.C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. on Information Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.