

# On Throughput Maximization in a Multi-hop MIMO Ad Hoc Network

Xiaoqi Qin

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Master of Science  
in  
Computer Engineering

Y. Thomas Hou, Chair  
Wenjing Lou  
Yaling Yang

May 8, 2013  
Blacksburg, Virginia

© Copyright 2013, Xiaoqi Qin

# On Throughput Maximization in a Multi-hop MIMO Ad Hoc Network

Xiaoqi Qin

## ABSTRACT

In recent years, there has been a growing research interest in throughput optimization problems in a multi-hop wireless network. MIMO (multiple-input multiple-output), as an advanced physical layer technology, has been employed in multi-hop wireless networks to increase throughput with a given bandwidth or transmit power. It exploits the use of multiple antennas at the transmitter and receiver to increase spectral efficiency by leveraging its spatial multiplexing (SM) and interference cancellation (IC) capabilities. Instead of carrying complex manipulations on matrices, degree-of-freedom (DoF) based MIMO models, which require only simple computations, are widely used in networking research to exploit MIMO's SM and IC capabilities.

In this thesis, we employ a new DoF model, which can ensure feasible solution and achieve a higher DoF region than previous DoF-based models. Based on this model, we study the DoF scheduling for a multi-hop MIMO network. Specifically, we aim to maximize the minimum rate among all sessions in the network. Some researches have been done based on this model to solve throughput optimization problems with the assumption that the route of each session is given priori. Although the fixed routing decreases the size of the problem, it also limits the performance of the network to a great extent.

The goal of this thesis is to employ this new model to solve the throughput maximization problem by jointly considering flow routing, scheduling, and DoF allocation for SM and IC. We formulate it as a mixed integer linear program (MILP), which cannot be solved efficiently by commercial softwares even for moderate sized networks. Thus, we develop an efficient polynomial time algorithm by customizing the *sequential fixing* framework. Through simulation results, we show that this algorithm can efficiently provide near-optimal solutions for networks with different sizes.

# Acknowledgments

“No one can whistle a symphony. It takes a whole orchestra to play it.” This thesis would not have been possible without the guidance and the help from several individuals. I would like to take this opportunity to express my sincere gratitude to these people who contributed their valuable assistance in the preparation and completion of this thesis.

First and foremost, I must offer my gratitude to my advisor, Prof. Y. Thomas Hou. He has been a supportive adviser to me throughout my graduate school studies. His ability to offer precise research vision and strategy is incredible. I cannot thank him enough for offering me the opportunity to learn from him. Throughout the whole process of preparing this thesis, his extensive knowledge, encouragement and advices always inspire me and bring me to a higher level of thinking. I am extremely fortunate to find an advisor who has always done everything in his power to help me. He is an incredible mentor and a research role model to me.

I am also extremely grateful to Dr. Yi Shi for his unselfish and unfailing support, helpful suggestions, and patience during the process of revising this paper. I am impressed by his sharp and insightful questions that contributed to improving this thesis. I have learned so much from him, not only the knowledge but also his rigorous research attitude which ensures absolute quality work.

I feel tremendously lucky to have had the opportunity to work with Xu Yuan. My greatest appreciation goes to him for being an invaluable resource for me whenever I have had questions about my research. His insightful ideas and depth of knowledge helped me think deeply about the

algorithm design in this thesis. His help is indispensable to the completion of this thesis.

Special thanks to my thesis committee, Prof. Yaling Yang and Prof. Wenjing Lou, for their valuable suggestions on my work.

My fellow students in the lab, including Liguang Xie, Huacheng Zeng and Borhan Jalaieian (Brian), also deserve my sincerest thanks. Their friendship and assistance have meant more to me than I could ever express. I am lucky to be surrounded by these brilliant people who are smarter than me. The environment in lab really pushed me and helped me to set a higher expectation for my work.

Last but not the least, I want to thank my parents for their encouragement, understanding and endless love. Although I am thousands miles away from home, I have never felt alone since they deliver their sincere love and encourage through words and phone calls regardless of their busy schedule. I worked hard because I didn't want to let them down. Words are not enough for me to express my appreciation of their faith in me.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Problem Considered in This Thesis . . . . .	2
1.3	Summary of Contributions . . . . .	3
1.4	Thesis Outline . . . . .	3
<b>2</b>	<b>Related Work</b>	<b>4</b>
<b>3</b>	<b>Modeling and Formulation</b>	<b>6</b>
3.1	Overview . . . . .	6
3.2	Mathematical Modeling . . . . .	6
3.3	Problem Formulation . . . . .	11
<b>4</b>	<b>An Efficient Solution Procedure</b>	<b>13</b>
4.1	Overview . . . . .	13
4.2	Some Details . . . . .	14
4.2.1	Stage One . . . . .	14

4.2.2	Stage Two . . . . .	18
4.3	Complexity Analysis . . . . .	22
<b>5</b>	<b>Simulation Results</b>	<b>24</b>
5.1	Simulation Setting . . . . .	24
5.2	Results . . . . .	24
5.2.1	20-node network with 2 sessions . . . . .	25
5.2.2	50-node network with 5 sessions . . . . .	29
5.3	Complete Results . . . . .	29
<b>6</b>	<b>Conclusion and Future Work</b>	<b>34</b>
6.1	Conclusion . . . . .	34
6.2	Future Work . . . . .	35
	<b>Bibliography</b>	<b>35</b>

# List of Figures

4.1	Flow chart for Stage One. . . . .	15
4.2	Flow chart for Stage Two. . . . .	19
4.3	Ordering adjustment. . . . .	21
5.1	The flow routing topology and time slot allocation for the 20-node instance under proposed algorithm. . . . .	27
5.2	The flow routing topology and time slot allocation for the 20-node instance under optimal solution. . . . .	28
5.3	The DoF allocation for IC at active nodes in time slot 1 for the 20-node instance under proposed algorithm. . . . .	28
5.4	The flow routing topology and time slot allocation for the 50-node instance obtained by proposed algorithm. . . . .	32

# List of Tables

- 3.1 Notation . . . . . 7
  
- 5.1 Location of each node for the 20-node network. . . . . 29
  
- 5.2 Details of time slot assignment and number of data streams on each active link for  
the 20-node network. . . . . 30
  
- 5.3 DoF usage at each active node for each time slot in the 20-node network. . . . . 31
  
- 5.4 Results for 50 instances of 20-node network. . . . . 33



# Chapter 1

## Introduction

### 1.1 Background

Recently, throughput maximization problems in multi-hop MIMO networks have been studied by using simplified degree-of-freedom (DoF) based MIMO models [2, 4, 6, 12]. Under a DoF-based model, the total number of DoFs at a node (or the number of antenna elements) represents the available resource at this node. Each DoF can be used for either spatial multiplexing (SM) or interference cancellation (IC) [20]. SM refers to the use of one or multiple DoFs to transport data streams at both transmit node and receive node. IC refers to the use of one or multiple DoFs to cancel interference so that several links can be active simultaneously.

However, these DoF-based models only focus on identifying sufficient conditions for feasible data streams under SM and IC. A new DoF-based link layer model was developed in [17] for multi-hop MIMO networks, in which a node-level ordering scheme is proposed to provide systematic rules to identify which node should perform IC. The rules are given as follows.

- *Transmit Node.* A transmit node should cancel its interference to all non-intended receivers before itself in the node ordering list. The number of DoFs required at the transmitter is equal to the sum of data streams received by those non-intended receivers.

- *Receive Node.* A receive node should cancel interference from all non-intended transmitters before itself in the node order. The number of DoFs required at the receiver is equal to the sum of data streams transmitted by those non-intended transmitters.

It was shown in [17] that for a given ordered list of nodes, the duplication in IC can be completely eliminated if the DoF allocation at each node is performed in the above manner. Thus, this model can ensure a feasible solution and achieve a higher DoF region than previous models.

## 1.2 Problem Considered in This Thesis

In this thesis, we employ the model in [17] to study a throughput maximization problem in a multi-hop MIMO network. We jointly consider flow routing, scheduling, and DoF allocation for SM and IC. This problem is more challenging than the case study in [17], where routing is pre-determined. We aim to maximize the minimum session rate in a multi-hop MIMO network and formulate this problem as a *mixed-integer linear program* (MILP), which is NP-hard in general and is not solvable via commercial solvers even for moderate sized networks. Therefore, we design an efficient and highly competitive algorithm to solve this throughput maximization problem.

Our algorithm is based on the *sequential fixing* (SF) framework [8], which is an efficient way to solve the MILP problem. We customize this framework and design an algorithm with two main stages. In the first stage, we have four phases to fix all integer variables sequentially according to the roles of different types of variables. In the second stage, we try to increase the number of data streams on the bottleneck link of the session with the smallest data rate by three approaches: (1) increasing the number of data streams on the bottleneck link without ordering adjustment. (2) adjusting the node ordering to accommodate one additional data stream on the bottleneck link. (3) finding a relay for the bottleneck link so that it can forward one data stream for the bottleneck link. This algorithm terminates when the number of data streams on the bottleneck link cannot be further increased.

### 1.3 Summary of Contributions

The main contributions of this paper can be summarized as follows.

- The throughput maximization problem is studied for a multi-hop MIMO network by employing the DoF-based MIMO model in [17]. Unlike the case study in [17], where routing is pre-determined, we consider flow routing as a part of our throughput maximization problem. Thus, our problem has a much larger optimization space, which yields a better solution. An optimization problem is formulated by jointly consideration of flow routing, scheduling, and DoF allocation to maximize the throughput.
- We design a polynomial time algorithm for the formulated MILP problem. This algorithm employs a customized SF framework. In addition, we further improve the SF solution in stage two. The designed algorithm can achieve near-optimal solutions with low complexity.
- Simulation results show that our algorithm is efficient and highly competitive. This algorithm can solve the throughput maximization problem for networks with different sizes, including those are not solvable by commercial softwares.

### 1.4 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we present the related work for using DoF-based models to solve the throughput maximization problem in multi-hop MIMO networks. In Chapter 3, we develop a throughput optimization problem formulation with joint consideration of flow routing, scheduling, and DoF allocation. In Chapter 4, we propose a polynomial time algorithm based on the SF framework and customize it to solve our throughput maximization problem. Chapter 5 presents simulation results and demonstrates the efficiency of our algorithm. In Chapter 6, we summarize our results in this thesis and present our future research directions.

# Chapter 2

## Related Work

Throughput maximization problems in multi-hop MIMO networks have been studied by using DoF-based MIMO models to exploit SM and IC capabilities. In [2], Bhatia and Li studied a throughput maximization problem for wireless multi-hop MIMO networks, by assuming that IC consumes DoFs at both transmitters and receivers. However, IC between an interfering transmitter and an interfered receiver only requires one of them to consume its DoFs [6, 17]. In [13], Mumey, Tang, and Hahn proposed an approximation algorithm to maximize throughput by joint stream control and scheduling in multi-hop MIMO networks. Since IC is performed only by receivers in this work, MIMO's IC capability at transmitters is not exploited.

Hamdaoui and Shin [6] studied the multi-hop MIMO network throughput and found that IC can be performed by either a transmitter or a receiver. Based on the model in [6], Blough *et al.* [4] formulated a MILP problem to maximize network throughput by allocating DoFs for SM and IC in a multi-hop MIMO network. However, it is not clear whether one can arbitrarily let a node to perform IC and obtain a feasible solution.

In [17], Shi *et al.* developed a DoF-based link layer model for multi-hop MIMO networks, which addressed the problem of which node should be responsible for IC to ensure a feasible solution. A throughput maximization problem is solved as a case study. Zeng *et al.* [22] employed

this model to design a distributed scheduling algorithm for throughput maximization in multi-hop MIMO networks. In [21], Yuan *et al.* employed this model to maximize throughput for multi-hop MIMO and cognitive radio networks. However, all these efforts assume that routing are given a prior, which makes the problem easier. In this paper, we consider routing as part of our problem.

# Chapter 3

## Modeling and Formulation

### 3.1 Overview

In this section, we apply the DoF model in [17] to formulate the throughput maximization problem for multi-hop MIMO networks. Under this approach, the total number of DoFs at a node is equal to the number of antenna elements at this node. A node can use some or all of its DoFs for either SM or IC, as long as the number of consumed DoFs does not exceed its total available DoFs.

### 3.2 Mathematical Modeling

We consider a set of MIMO nodes,  $\mathcal{N}$ , in a multi-hop network, where  $N = |\mathcal{N}|$  is the number of nodes. Each MIMO node  $i$  has  $A_i$  antennas. We consider a time slot based scheduling with  $T$  equal-length time slots in a frame. Denote  $\mathcal{F}$  as the set of active user communication (unicast) sessions in the network. Table 3.1 lists notations used in this paper.

**Half-Duplex Constraint.** We assume that wireless transceivers are half-duplex. That is, a node cannot transmit and receive at the same time. To model this constraint in each time slot, we define two binary variables  $x_i[t]$  and  $y_i[t]$  to indicate whether node  $i$  is a transmitter or a receiver in time

Table 3.1: Notation

Symbol	Definition
$\mathcal{N}$	Set of nodes in the network
$\mathcal{F}$	Set of sessions in the network
$T$	Total number of time slots for scheduling in a frame
$\mathcal{T}_i$	Set of nodes within the transmission range of node $i \in \mathcal{N}$
$\mathcal{I}_i$	Set of nodes within the interference range of node $i \in \mathcal{N}$
$A_i$	Number of antennas at node $i \in \mathcal{N}$
$x_i[t]$	=1 if node $i$ is a transmitter in time slot $t$ , and is 0 otherwise
$y_i[t]$	=1 if node $i$ is a receiver in time slot $t$ , and is 0 otherwise
$z_{ij}[t]$	Number of data streams transmitted from node $i$ to node $j$ in time slot $t$
$c_{ij}$	Achievable data rate on link $(i, j)$ over $T$ time slots
$\theta_{ji}[t]$	A binary variable to indicate whether node $i$ is placed after node $j$ in $\pi[t]$
$\pi_i[t]$	The position of node $i$ in a node-level ordering $\pi[t]$
$\mathcal{F}$	Set of multi-hop sessions
$s(f)$	The source node of session $f \in \mathcal{F}$
$d(f)$	The destination node of session $f \in \mathcal{F}$
$r(f)$	The achieved data rate for each session $f \in \mathcal{F}$
$r_{ij}(f)$	The data rate that attributed to session $f$ on link $(i, j)$ in time slot $t$

slot  $t$ , respectively. That is,

$$x_i[t] = \begin{cases} 1 & \text{if node } i \text{ is a transmitter in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.1)$$

$$y_i[t] = \begin{cases} 1 & \text{if node } i \text{ is a receiver in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.2)$$

Then the half-duplex constraint can be modeled as

$$x_i[t] + y_i[t] \leq 1 \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.3)$$

**Constraints for Node and Link Activity.** Denote  $z_{ij}(t)$  as the number of data streams on link  $(i, j)$  in time slot  $t$ . If a node  $i$  is not an active transmitter in time slot  $t$ , then no data stream is transmitted at 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

$$x_i[t] \leq \sum_{j \in \mathcal{T}_i} z_{ij}[t] \leq A_i \cdot x_i[t] \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.4)$$

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

$$y_i[t] \leq \sum_{j \in \mathcal{T}_i} z_{ji}[t] \leq A_i \cdot y_i[t] \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.5)$$

**Ordering Constraints.** The ‘‘ordering’’ concept is proposed in [17] to avoid unnecessary duplication in IC (and thus leads to a waste of DoF resources) and guarantee a feasible solution. Denote  $\pi[t]$  as the ordering list of nodes in time slot  $t$ , and denote integer variable  $\pi_i[t]$  as the position of node  $i$  in time slot  $t$ . Therefore, we have

$$1 \leq \pi_i[t] \leq N \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.6)$$



We use a binary variable  $\theta_{ji}[t]$  to indicate the ordering relationship between two nodes  $i$  and  $j$ . We define  $\theta_{ji}[t]$  as

$$\theta_{ji}[t] = \begin{cases} 1 & \text{if node } i \text{ is after node } j \text{ in } \pi[t]; \\ 0 & \text{otherwise.} \end{cases} \quad (3.7)$$

It was shown in [17] that the following relationships hold among  $\pi_i[t]$ ,  $\pi_j[t]$  and  $\theta_{ji}[t]$ :

$$\pi_i[t] - N \cdot \theta_{ji}[t] + 1 \leq \pi_j[t] \leq \pi_i[t] - N \cdot \theta_{ji}[t] + N - 1 \quad (i, j \in \mathcal{N}, i \neq j, 1 \leq t \leq T). \quad (3.8)$$

**DoF Consumption Constraints.** Under a particular node ordering, we can identify which node (transmitter or receiver) has the responsibility to perform IC as follows [17].

- A transmitter should cancel its interference to all the unintended receivers that are before itself in the ordered node list.
- A receiver should cancel interference from all unintended transmitters that are before itself in the ordered node list.

Then if a node  $i$  is a transmitter, the number of DoFs required at node  $i$  for IC is equal to the total number of data streams received by unintended receivers that are within its interference range and are before itself in the ordered node list, which is  $\sum_{j \in \mathcal{I}_i} (\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t])$ , where  $\mathcal{I}_i$  is the set of nodes within the interference range of node  $i$ . On the other hand, if node  $i$  is not a transmitter, no DoF is consumed. We can model a uniform constraint for the above two cases as follows.

$$\sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \left( \theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \right) \leq A_i x_i[t] + (1 - x_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (3.9)$$

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

The above constraint has a nonlinear term  $\sum_{j \in \mathcal{I}_i} (\theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t])$ . We can employ the *Reformulated-Linearization Technique* (RLT) [16] to reformulate a nonlinear term into a set of linear constraints. We introduce a new variable  $\lambda_{ji}[t] = \theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t]$ . Then constraint (3.9) can be replaced by the following linear constraint.

$$\sum_{j \in \mathcal{T}_i} z_{ij}[t] + \sum_{j \in \mathcal{I}_i} \lambda_{ji}[t] \leq A_i x_i[t] + (1 - x_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.10)$$

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 the relationships between  $\theta_{ji}[t]$ ,  $\sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t]$  and  $\lambda_{ji}[t]$ .

$$\lambda_{ji}[t] \leq \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T) \quad (3.11)$$

$$\lambda_{ji}[t] \leq A_j \cdot \theta_{ji}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T) \quad (3.12)$$

$$\lambda_{ji}[t] \geq A_j \cdot \theta_{ji}[t] + \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{kj}[t] - A_j \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T). \quad (3.13)$$

Similarly, the DoFs consumption at a potential receiver  $i$  is either  $\sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} (\theta_{ji}[t] \cdot$

$\sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t])$  or zero, and can be modeled as follows.

$$\sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} \left( \theta_{ji}[t] \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] \right) \leq A_i y_i[t] + (1 - y_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T). \quad (3.14)$$

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 we can linearize the above constraint by the following set of constraints:

$$\sum_{j \in \mathcal{T}_i} z_{ji}[t] + \sum_{j \in \mathcal{I}_i} u_{ji}[t] \leq A_i y_i[t] + (1 - y_i[t]) B_i \quad (i \in \mathcal{N}, 1 \leq t \leq T), \quad (3.15)$$

$$u_{ji}[t] \leq \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T), \quad (3.16)$$

$$u_{ji}[t] \leq A_j \cdot \theta_{ji}[t] \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T), \quad (3.17)$$

$$u_{ji}[t] \geq A_j \cdot \theta_{ji}[t] + \sum_{k \in \mathcal{T}_j}^{k \neq i} z_{jk}[t] - A_j \quad (i \in \mathcal{N}, j \in \mathcal{I}_i, 1 \leq t \leq T). \quad (3.18)$$

**Flow Balance Constraints.** Denote  $s(f)$  and  $d(f)$  as the source and destination nodes of each session  $f \in \mathcal{F}$ , and  $r(f)$  as the achieved throughput of session  $f$ . For flexibility and 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 transmitted 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

$$\sum_{j \in \mathcal{T}_i} r_{ij}(f) = r(f) \quad (f \in \mathcal{F}). \quad (3.19)$$

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

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(f)} r_{ij}(f) = \sum_{k \in \mathcal{T}_i}^{k \neq d(f)} r_{ki}(f) \quad (f \in \mathcal{F}, i \in \mathcal{N}). \quad (3.20)$$

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

$$\sum_{j \in \mathcal{T}_i} r_{ji}(f) = r(f) \quad (f \in \mathcal{F}). \quad (3.21)$$

It can be easily verified that once (3.19) and (3.20) are satisfied, (3.21) must also be satisfied. As a result, it is sufficient to just include (3.19) and (3.20) in the formulation.

**Link Capacity Constraints.** For each link  $(i, j)$ , the sum of the data rate that is attributed to each session cannot exceed the average of its data rate over  $T$  time slots. For simplicity, we assume that one data stream corresponds to one unit of data rate. Then we have the following constraint:

$$\sum_{f \in \mathcal{F}} r_{ij}(f) \leq C_{ij} = \frac{1}{T} \sum_{t=1}^T z_{ij}[t] \quad (i \in \mathcal{N}, j \in \mathcal{T}_i). \quad (3.22)$$

### 3.3 Problem Formulation

In this paper, we study a throughput optimization problem with the objective of maximizing the minimum data rate among all sessions. Denote  $r_{\min}$  as the minimum throughput among all sessions. The problem can be formulated as follows:

OPT

max  $r_{\min}$

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

Half Duplex Constraints: (3.3);

Node and Link Activity Constraints: (3.4), (3.5);

Node Ordering Constraints: (3.6), (3.8);

DoF Consumption Constraints for Transmitters: (3.10), (3.11), (3.12), (3.13);

DoF Consumption Constraints for Receivers: (3.15), (3.16), (3.17), (3.18);

Flow Balance Constraints: (3.19), (3.20);

Link Capacity Constraints: (3.22).

In this formulation,  $x_i[t]$ ,  $y_i[t]$ ,  $z_{ij}[t]$ ,  $\pi_i[t]$  and  $\theta_{ji}[t]$  are integer variables,  $r_{\min}$ ,  $r(f)$ ,  $\lambda_{ji}[t]$  and  $\mu_{ji}[t]$  are continuous variables,  $A_i$  and  $B_i$  are constants. 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 want to develop a highly competitive and efficient algorithm for this problem.

# Chapter 4

## An Efficient Solution Procedure

### 4.1 Overview

In this section, we design an efficient algorithm based on the so-called *sequential fixing* (SF) technique [8]. SF offers a general framework to handle integer variables in a MILP problem. The basic idea behind SF is as follows. For a MILP like ours, if we were able to set the optimal values for all the integer variables and thereby reduce the original problem to a linear program (LP), then we can solve the reduced problem optimally in polynomial time. Thus, the key challenge in such contexts is how to determine the values of all the integer variables. This can be done by examining the relaxed version of the original problem, which is obtained by relaxing all the integer variables to continuous variables. Although the solution to this relaxation may not have an integer value for each integer variable, we can set (i.e., fix) the values of integer variables based on the *closeness* to certain integer values in the relaxed solution. Instead of determining all integer variable values via a single relaxation, we can fix only one or a few integer variables in each iteration. For the remaining (unfixed) integer variables, we can solve a new relaxation (with some integer variables' values being already fixed) and then fix more integer variables. This SF procedure terminates after we fix all the integer variables. Then the value of other non-integer variables in the original problem can be obtained by solving the resulting LP.

There are two main stages in our algorithm. At the first stage, the algorithm employs SF framework to obtain a feasible solution. Although the idea of SF is straightforward, this stage can be enhanced by problem specific customization. At the second stage, the algorithm attempts to increase the data rate on the current bottleneck link in each iteration until the bottleneck rate cannot be further increased.

## 4.2 Some Details

### 4.2.1 Stage One

In this stage, we obtain a feasible solution by the SF framework. Since different types of integer variables play different roles in the optimization problem, we decide to first fix ordering variables  $(\pi, \theta)$ , then fix node status variables  $(x, y)$ , and finally fix data stream variable  $z$ . The ordering variables  $(\pi, \theta)$  play a key role in the feasible solution since they determine the IC responsibility of each node. Therefore, we determine the values of  $(\pi, \theta)$  in Phase I. Then in Phase II, we determine the values of  $(x, y)$  by identifying the active (corresponding  $z > 0$ ) or inactive (corresponding  $z = 0$ ) status of each link. Note that the route of each session is also identified by active links. After all active links are determined, to avoid wasting of DoF resources among nodes in the network, we will de-active some links which do not contribute to session rates in Phase III. In Phase IV, we will fix the exact integer value of the number of data streams ( $z$  value) on each remaining active link iteratively. Note that these phases are within the SF framework. That is, we determine all integer variables in iterations. Once we fix some integer variables' values in an iteration, we build a new MILP for the remaining variables in the next iteration and solve its relaxed LP. Then we can determine more integer variables based on the solution to the relaxed LP. A flow chart of stage one is shown in Fig. 4.1.

**Phase I: Fixing  $\pi$  and  $\theta$  variables.** In this phase, we fix  $\pi$  and  $\theta$  variables in  $N - 1$  iterations. Specifically, we run the following process in the  $k$ -th iteration for each time slot  $t$ . We identify

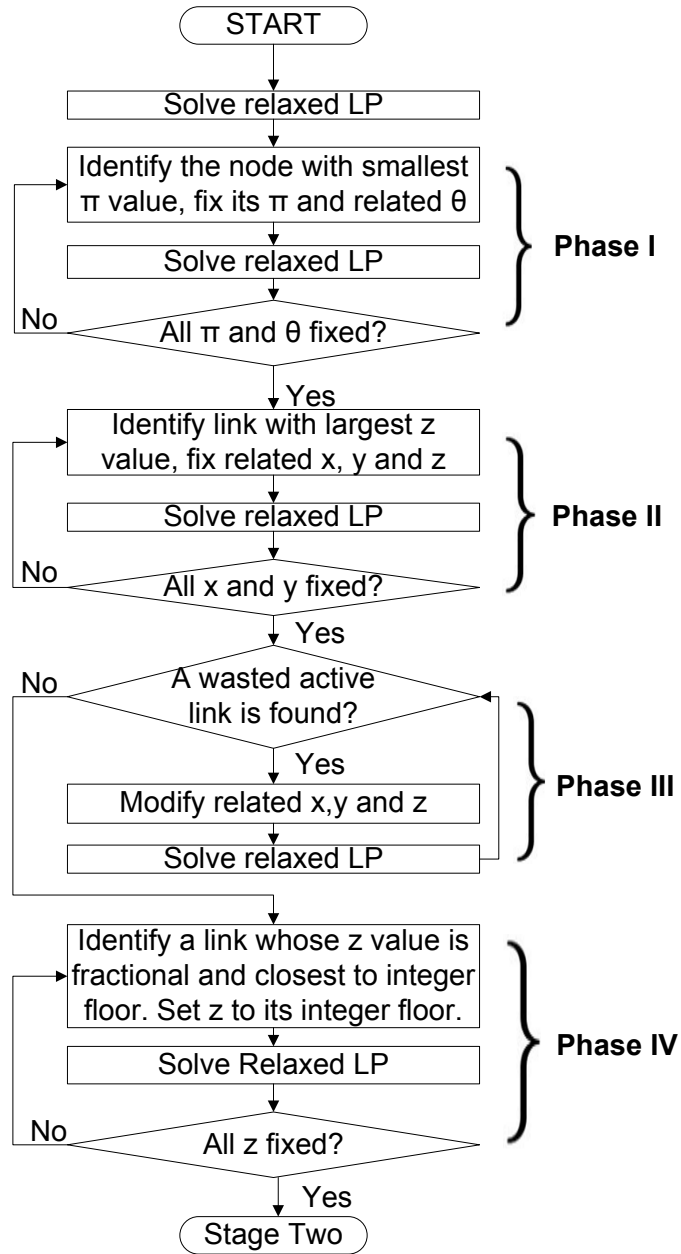


Figure 4.1: Flow chart for Stage One.

the node  $i$  with the smallest  $\pi_i[t]$  among all nodes with unfixed  $\pi$  and set it as the  $k$ -th node in the ordering list (i.e.,  $\pi_i[t] = k$ ). Then we can fix  $\theta_{ji}[t] = 0$  and  $\theta_{ij}[t] = 1$  (according to Eq. (3.7)) for each node  $j$  with unfixed  $\pi_j[t]$ . Note that after the  $(N - 1)$ -th iteration, there is only one node with unfixed  $\pi$  variable and we can fix this variable as  $N$ . Thus, all  $\pi$  and  $\theta$  variables are fixed in  $N - 1$  iterations.

**Phase II: Fixing  $x$  and  $y$  variables.** In this phase, we fix  $x_i[t]$  and  $y_i[t]$  values by identifying the active or inactive status of the corresponding link in time slot  $t$ .

Specifically, we run the following processes in an iteration for each time slot  $t$ . We set link  $(i, j)$  with the largest  $z_{ij}[t]$  among links whose statuses are unknown to an active link. Note that this is feasible because when we set a link as active in previous iteration, we already identified all links that cannot be active simultaneously with this link and set them as inactive. As a result, in current iteration, any link with unknown status can be set as active.

Once we set link  $(i, j)$  as active in time slot  $t$ , we can fix  $x_i[t] = 1$  and  $y_j[t] = 1$  based on Eqs. (3.1) and (3.2). We can further fix more  $x, y$ , and  $z$  variables by the following rules.

- According to (3.3), we can also fix  $y_i[t] = 0$  for the transmitter and  $x_j[t] = 0$  for the receiver of link  $(i, j)$ . Then we can fix all the outgoing links from node  $j$  as inactive (according to Eq. (3.4)) and all the incoming links to node  $i$  as inactive (according to Eq. (3.5)). That is, we set  $z_{ki}[t] = 0$  for each node  $k \in \mathcal{T}_i$  and  $z_{jk}[t] = 0$  for each node  $k \in \mathcal{T}_j$ .
- We can further determine more links' inactive status as follows. Since link  $(i, j)$  becomes active, this link has at least one data stream (previously at least zero). By assuming one more data stream received at node  $j$ , we can calculate the minimum DoF consumption for IC at a potential transmitter  $k$ , where  $k \in \mathcal{I}_j$  and  $\pi_k[t] > \pi_j[t]$ . If the DoF consumption at node  $k$  equals to or is larger than the number of antennas at this node, then it cannot be an active transmitter (set  $x_k[t] = 0$ ) based on Eq. (3.9). Moreover, all outgoing links from node  $k$  cannot be active (set corresponding  $z$  to be zero) based on Eq. (3.4). Similarly, by considering node  $i$ , some nodes cannot be active receivers based on Eq. (3.14) and incoming



links to these nodes cannot be active based on Eq. (3.5). Thus, we can set the corresponding  $y$  and  $z$  to be zero.

- We can also calculate the minimum DoF consumption at nodes  $i$  and  $j$ . For node  $j$ , we consider each active transmitter  $k$ , where  $k \in \mathcal{I}_j$ ,  $\pi_k[t] < \pi_j[t]$ , and  $x_k[t] = 1$ . By assuming one data stream on the corresponding links, we can calculate the minimum DoF consumption for IC at node  $j$ . If the sum of this DoF consumption for IC and the DoF consumption for SM at node  $j$  equals to its number of antennas, then each node  $h$ , where  $h \in \mathcal{I}_j$ ,  $\pi_h[t] < \pi_j[t]$ , and  $x_h[t]$  unfixed, cannot be an active transmitter (set  $x_h[t] = 0$ ) based on Eq. (3.9). Moreover, all outgoing links from node  $h$  cannot be active (set corresponding  $z$  to be zero) based on Eq. (3.4). Similarly, by considering node  $i$ , some nodes cannot be active receivers based on Eq. (3.14) and incoming links to these nodes cannot be active by Eq. (3.5). Thus, we can set the corresponding  $y$  and  $z$  to be zero.

We continue the process until the largest  $z_{ij}[t]$  is 0, which means that all possible active links have been fixed. Then we fix all the remaining links as inactive links (i.e., set  $z$  to zero). For these links' transmitters (or receivers), we set their  $x$  (or  $y$ ) values to zero if these variables are not determined yet.

**Phase III: De-active wasted active links.** In this phase, we identify all active links that do not contribute to any session's throughput, and change these wasted links' status to be inactive. Specifically, if we find an active link  $(i, j)$  with  $r_{ij}(f) = 0$  for all sessions, then we set  $z_{ij}[t] = 0$  for each time slot  $t$ . After these changes, if all outgoing links from node  $i$  are inactive, we set the status of node  $i$  as an inactive transmitter (i.e.,  $x_i[t] = 0$  for each time slot  $t$ ), so that constraint (3.4) holds. Similarly, if all incoming links to node  $j$  are inactive, we set the status of node  $j$  as an inactive receiver (i.e.,  $y_j[t] = 0$  for each time slot  $t$ ), so that constraint (3.5) holds.

**Phase IV: Fixing  $z$  variables.** In Phases II and III, we have fixed  $z_{ij}[t]$  to 0 for those inactive links and wasted links. In this phase, we fix the exact integer value of  $z_{ij}[t]$  for the remaining active links. Specifically, we run the following process in an iteration for each time slot  $t$ . We select the link  $(i, j)$  whose  $z_{ij}[t]$  value is closest to its integer floor  $\lfloor z_{ij}[t] \rfloor$  among all links with unfixed  $z$

variables in time slot  $t$  and set this  $z$  variable's value as  $\lfloor z_{ij}[t] \rfloor$ .

### 4.2.2 Stage Two

In this stage, we increase throughput for all sessions in iterations. There are four phases in each iteration. We first identify the bottleneck link in Phase V. We want to increase the number of data streams from the transmitter of this bottleneck link to its receiver by approaches in Phases VI, VII, and VIII. Specifically, in Phase VI, we try to increase the number of data streams on the bottleneck link without changing node ordering. In Phase VII, we try to alter the node ordering to relieve some DoFs from some nodes to support one more data stream on the bottleneck link. We hope that some DoFs can be relieved from some nodes in the network such that one more data stream can be transmitted on the bottleneck link. In Phase VIII, we try to find a relay node to forward one data stream from the bottleneck link's transmitter to its receiver. If the number of data streams is increased, we will update related  $x, y, z, \pi$ , and  $\theta$  values if necessary and solve an LP to improve routing solution (i.e., update  $r$  values). Then the current iteration is completed and we will identify a new bottleneck link in the next iteration. On the other hand, if all the above three phases fail to increase the number of data streams, our algorithm terminates. A flow chart of stage two is shown in Fig. 4.2.

**Phase V: Finding bottleneck link.** In this phase, we identify a link for data rate increment. We first identify a session  $f$  with the minimum throughput among all sessions in the network. Then among those links that transmit data for session  $f$ , we find the bottleneck link, i.e., a link with its average data rate over  $T$  time slots equal to its total data rate for all sessions. A tie among sessions or links can be broken arbitrarily.

**Phase VI: Increasing number of data streams without ordering adjustment.** Suppose that in Phase V, we identified the bottleneck link  $(i, j)$  for session  $f$ . In this phase, we try to increase at least one data stream on this link in a time slot.

The link  $(i, j)$  can transport one more data stream in a currently used time slot  $t$  if the following

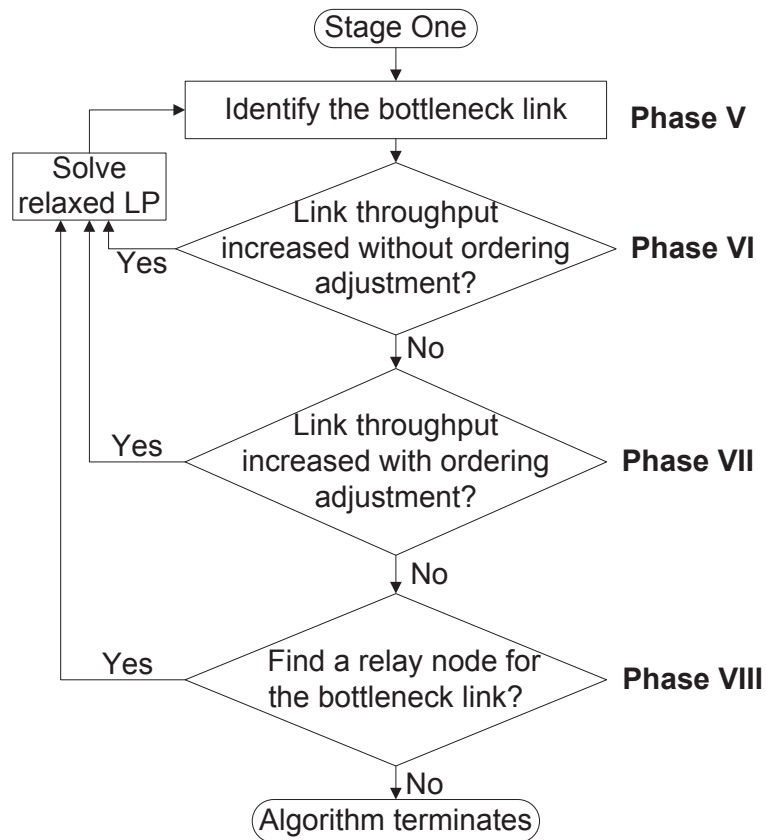


Figure 4.2: Flow chart for Stage Two.

conditions are satisfied:

- Both nodes  $i$  and  $j$  have at least one remaining DoF.
- Each receiver  $h, h \in \mathcal{I}_i$  and  $\pi_h[t] > \pi_i[t]$ , has at least one remaining DoF to cancel additional interference from node  $i$ . Each transmitter  $k, k \in \mathcal{I}_j$  and  $\pi_k[t] > \pi_j[t]$ , has at least one remaining DoF to cancel interference to the additional data stream at node  $j$ .

Same conditions should be satisfied if we want to activate link  $(i, j)$  in a time slot  $t$  that is currently not used by link  $(i, j)$ . But we need to update more variables to activate link  $(i, j)$  in time slot  $t$ . That is, we need to set  $z_{ij}[t] = 1$ . If current  $x_i[t]$  and  $y_j[t]$  are not one, we need to update them to one.

**Phase VII: Ordering adjustment.** When Phase VI fails to increase one data stream on the bottleneck link, it can be concluded that there is a lack of DoF resources at a subset of nodes among the bottleneck link's transmitter, receiver and their neighboring nodes in every time slot. Recall that the amount of consumed DoFs for IC is determined by the node ordering. In this phase, we try to modify node ordering in a time slot so that some DoFs can be relieved to allow one more data stream on the bottleneck link.

For the bottleneck link  $(i, j)$  and a time slot  $t$ , the altering process has the following two steps.

*Step 1:* If in time slot  $t$ , either node  $i$  or node  $j$  does not have any remaining DoF, then we try to relieve some DoFs consumed by IC as follows. Otherwise, we can skip this step.

If transmit node  $i$  does not have any remaining DoF, we can try to decrease its DoF consumption for IC as follows. Denote  $\varepsilon_i$  as the set of nodes that includes any receiver  $h$  with  $h \in \mathcal{I}_i$  and  $\pi_h[t] < \pi_i[t]$ . Suppose node  $k$  has the largest  $\pi_k[t]$  among nodes in  $\varepsilon_i$ . For other nodes in  $\varepsilon_i$ , we compute the remaining DoFs with the assumption that we move it to the position  $\pi_k[t]$  and keep other nodes' relative position unchanged. Suppose that node  $m$  has the largest remaining DoFs (after this move). A tie can be broken by selecting the node with a smaller node ID. If the remaining DoFs at node  $m$  is greater than the DoFs consumed by SM at node  $i$ , we make the

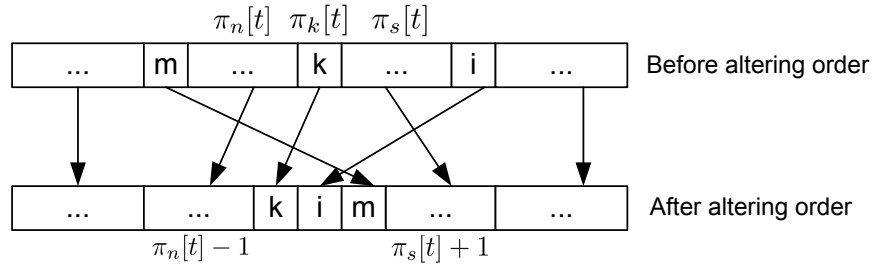


Figure 4.3: Ordering adjustment.

following changes (See Fig. 4.3): (i) move node  $m$  to position  $\pi_k[t] + 1$ ; (ii) for any node  $n$  between nodes  $m$  and  $k$  in the original list, move it to position  $\pi_n[t] - 1$ ; (iii) for any node  $s$  between nodes  $k$  and  $i$  in the original list, move it to position  $\pi_s[t] + 1$ ; (iv) move node  $i$  to position  $\pi_k[t]$ ; and (v) move node  $k$  to position  $\pi_k[t] - 1$ . It can be verified that the new solution is feasible.

If receiver node  $j$  does not have any remaining DoF, we can try to decrease its DoF consumption by a similar approach.

*Step 2:* In time slot  $t$ , both nodes  $i$  and  $j$  have some remaining DoFs but some neighboring nodes have no remaining DoFs.

Denote  $\varphi_i$  as the set of nodes that includes any receiver  $h$  with zero remaining DoF,  $h \in \mathcal{I}_i$ , and  $\pi_h[t] > \pi_i[t]$ . For each receiver  $h$  in  $\varphi_i$ , we follow a similar approach for receiver node mentioned in step I to alter its ordering. If success, node  $h$  can relieve some of its DoFs consumed by IC, and use them to perform IC for node  $i$ . Similarly, we can relieve some DoFs for node  $j$ 's neighbors.

Recall that there are  $T$  time slots in a time frame. We check each time slot until a rate increment is done or all time slots fail.

**Phase VIII: Finding a relay.** When both Phases VI and VII fail to increase one data stream on bottleneck link  $(i, j)$  in all time slots, we enter this phase. In this phase, we select a relay node  $k$  to construct an additional route between nodes  $i$  and  $j$  by adding one data stream on both links  $(i, k)$  and  $(k, j)$ . A node  $k$  can be selected as a relay if it satisfies the following conditions: (i) the set of time slots  $\tau_i$ , where node  $i$  has remaining DoFs (if we set it as an active transmitter) and node  $k$  is

or can be a receiver and has remaining DoFs, is not empty. (ii) the set of time slots  $\tau_j$ , where node  $j$  has remaining DoFs (if we set it as an active receiver) and node  $k$  is or can be a transmitter and has remaining DoFs, is not empty. (iii) the union of  $\tau_i$  and  $\tau_j$  has at least two time slots.

For link  $(i, k)$  in time slot  $t \in \tau_i$ , link  $(i, k)$  can carry one more data stream with the current node ordering if the following conditions are satisfied: (i) node  $i$  has at least one remaining DoF (if we set it as an active transmitter) in time slot  $t$ ; (ii) each receiver  $h \in \mathcal{I}_i$  and  $\pi_h[t] > \pi_i[t]$  has at least one remaining DoF to cancel additional interference from node  $i$ ; (iii) each transmitter  $h \in \mathcal{I}_k$  and  $\pi_h[t] > \pi_k[t]$  has at least one remaining DoF to cancel interference to node  $k$ . If these conditions cannot be satisfied, we use a similar process in Phase VII to alter the node ordering by assuming link  $(i, k)$  is the bottleneck link. If the rate increment fails in the current time slot, we try another time slot in  $\tau_i$ , until the data rate is increased or all slots in  $\tau_i$  fail.

Following a similar approach, we can try to add one more data stream on link  $(k, j)$ . If both links  $(i, k)$  and  $(k, j)$  can carry one more data stream, then the data rate between nodes  $i$  and  $j$  can be increased.

### 4.3 Complexity Analysis

We now show that our algorithm has a polynomial complexity. For each stage of the algorithm, we analyze the number of iterations and the complexity of each iteration.

At stage one, the complexity of each iteration in each phase involves solving the relaxed LP, searching for the unfixed integer variable with the largest value in the relaxed LP solution, and fixing the integer values for selected variables. The complexity of searching for the unfixed integer variable with the largest value and fixing integer values for selected variables are much less than solving an LP. The complexity of solving an LP is  $O(V^3)$  [1], where  $V$  is the number of variables. In our algorithm, three sets of variables,  $\theta_{ji}[t]$ ,  $z_{ij}[t]$  and  $r_{ij}(f)$ , dominate the total amount of variables. Both  $\theta_{ji}[t]$  and  $z_{ij}[t]$  have  $O(N^2 \cdot T)$  variables while  $r_{ij}(f)$  has  $O(N^2 \cdot |\mathcal{F}|)$  variables. Therefore,  $V = O(N^2 \cdot \max\{T, |\mathcal{F}|\})$ .

To analyze the total number of iterations at stage one, we now analyze the total number of iterations needed for each phase. In Phase I, we have a total of  $N - 1$  iterations as we discussed. In Phase II, we determine at least one link's active or inactive status for each time slot in an iteration. Since there are  $O(N^2)$  links, the maximum number iterations in this phase is  $O(N^2)$ . In Phase III, we determine a wasted link for each time slot in an iteration. Then the maximum number iterations in this phase is also  $O(N^2)$ . In Phase IV, we fix one  $z$  value (for a link) for each time slot in an iteration. Then the total number of iterations is  $O(N^2)$ . Therefore, we have no more than  $O(N + 3N^2) = O(N^2)$  iterations for stage one. As a result, the complexity of stage one is  $O(N^2V^3)$ .

At stage two, the complexity of each iteration involves identifying the bottleneck link (Phase V), increasing data streams on this link (Phases VI, VII, and VIII), and solving an LP to improve routing solution. Variables  $r_{ij}(f)$  dominate the total amount of variables. Thus, we have  $O(N^2 \cdot |\mathcal{F}|) < V$  variables. Then complexity of solving an LP is  $O(V^3)$  and is much higher than the complexity in Phases V – VIII. Thus, the complexity of an iteration is  $O(V^3)$ .

We now analyze the total number of iterations at stage two. Since the maximum number of links is  $O(N^2)$ , and for each of these links, we can increase its data streams at most  $A_i$  times, the total number of iterations is  $O(N^2AT)$ , where  $A$  is the maximum antenna number among all nodes. As a result, the complexity of stage two is  $O(N^2AT \cdot V^3)$ .

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

# Chapter 5

## Simulation Results

### 5.1 Simulation Setting

We consider a multi-hop ad hoc network, where all nodes are randomly generated in a  $100 \times 100$  area. For generality, we normalize all units for distance, time, bandwidth, and data rate with appropriate dimensions. We assume that all nodes' transmission range and interference range are 30 and 50, respectively. We present our results for two different network settings: (1) 20 nodes with 2 sessions and (2) 50 nodes with 5 sessions. For each network instance, the source and destination nodes of each session are randomly selected. We assume that all nodes in the network are equipped with four antennas, and there are four time slots in a time frame. We will first show one example for either network size, and then show the complete simulation results of 50 randomly generated network instances for 20-node networks.

### 5.2 Results

Before we present the complete simulation results, we first show one example for each network size.



### 5.2.1 20-node network with 2 sessions

We first show a case study of a 20-node network instance. The locations of the 20 nodes are shown in Table 5.1. The source and destination nodes for each session are randomly selected and are shown in Figure 5.1. The objective value found by our algorithm is 0.75. The optimal objective value found by CPLEX is also 0.75, which shows that our solution is optimal in this case study.

Now, we show the details of solution obtained by our algorithm. The node ordering lists in the four time slots are the following:

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

Figure 5.1 shows the routing topologies and scheduling for each session obtained by our proposed algorithm. Fig. 5.2 shows the routing topologies and scheduling for each session under the optimal solution obtained by CPLEX. The number in the box on each link indicates the time slots where the link is active. Details of the number of data streams transmitted on each link are listed in Table 5.2.

Given that a time frame is divided into 4 time slots, we should have DoF allocation in each of the 4 time slots. We now give a detailed explanation of the DoF allocation at each active node in a particular time slot, say time slot 1. To show how nodes perform SM and IC, we consider time

slot 1. Note that time slot 1 is used by links  $(N_2, N_3)$ ,  $(N_9, N_{19})$ ,  $(N_{18}, N_{13})$  and  $(N_{18}, N_{14})$  in Fig. 5.1. The number of data streams on each link can be found in Table 5.2, i.e., in time slot 1, there are three data streams on link  $(N_2, N_3)$ , one data stream on link  $(N_9, N_{19})$  and  $(N_{18}, N_{14})$ , and two data streams on link  $(N_{18}, N_{13})$ . The interference relationships among these transmitters and receivers are shown in Fig. 5.3 by the dashed arrows, i.e., node  $N_9$  interferes node  $N_3$  and  $N_{13}$ , node  $N_{19}$  interferes  $N_{18}$ , and node  $N_2$  interferes  $N_{19}$ . The number on each dashed arrow represents the number of DoFs consumed for IC.

- The position of node  $N_2$  is 12 in the ordered node list. Since the receiver  $N_{19}$  is within its interference range and before it in the ordering, node  $N_2$  needs to use one DoFs to cancel its interference to node  $N_{19}$ . Then, it can use three remaining DoFs to transmit three data streams to node  $N_3$ .
- The position of node  $N_3$  is 2 in the ordered node list. There is no transmit node which is within its interference range and before it in the ordering, then it does not consume DoFs for IC. Therefore, it can use three DoFs to receive data streams from node  $N_2$ .
- The position of node  $N_9$  is 15 in the ordered node list. Since the receiver  $N_3$  is within its interference range and before it in the ordering list, node  $N_9$  need to consume three DoFs to cancel interference to node  $N_3$ . Then it can use the remaining one DoF to transmit one data stream to node  $N_{19}$ .
- The position of node  $N_{19}$  is the first node in the ordered node list. It is a receive node, it uses one DoF to receive one data stream from node  $N_9$ . Since there is no transmit node which is within its interference range before it in the ordering, it does not need to use any DoF to do IC.
- Node  $N_{18}$  is the last node in ordered node list, i.e.,  $\pi_{20}[1] = 20$ . It is a transmit node, it uses one DoF to transmit one data stream to node  $N_{14}$  and two DoFs to transmit two data streams to node  $N_{13}$ . In addition, since receive node  $N_{19}$  is within its interference range and

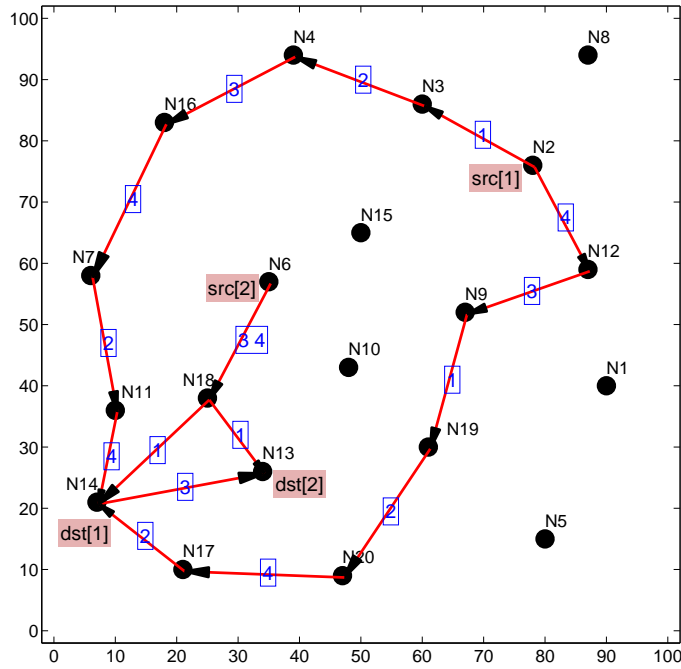


Figure 5.1: The flow routing topology and time slot allocation for the 20-node instance under proposed algorithm.

before it in the ordered list, it needs to ensure that its transmissions do not interfere with  $N_{19}$ 's reception. Therefore, it must use one DoF to cancel the interference to  $N_{19}$ .

- The position of node  $N_{13}$  is 17 in the ordered node list. It is a receive node, it uses two DoFs to receive two data streams from node  $N_{18}$ . In addition, since transmit node  $N_9$  is within its interference range and before it in the ordering, it needs to ensure that its reception is not interfered by  $N_9$ 's transmission. Therefore, it must use one DoFs to cancel the interference from  $N_9$ .
- The next node is  $N_{14}$ , its position is 18 in the ordered node list. It is a receive node, it uses one DoF to receive one data stream from node  $N_{18}$ . Since none of the transmit nodes before it in the ordering is within its interference range, it does not need to use any DoF to do IC.

The details of DoFs allocation for SM and IC at each node in each time slot are shown in Table 5.3.

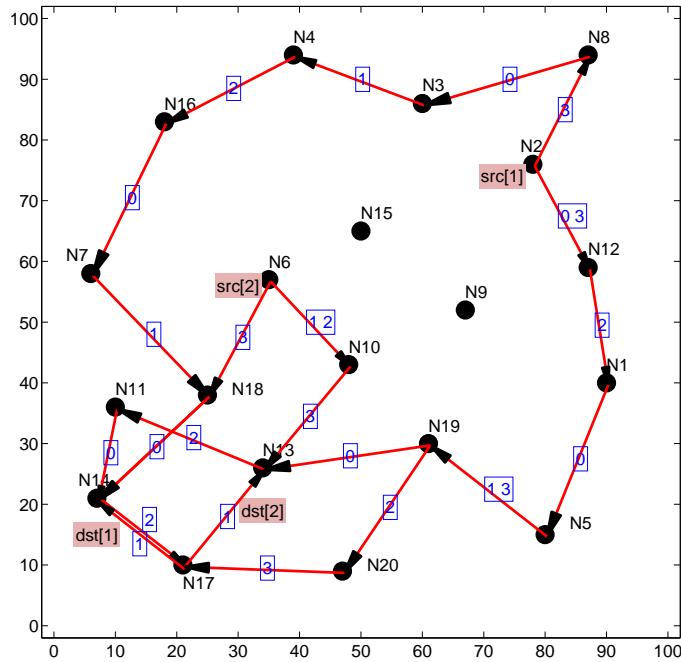


Figure 5.2: The flow routing topology and time slot allocation for the 20-node instance under optimal solution.

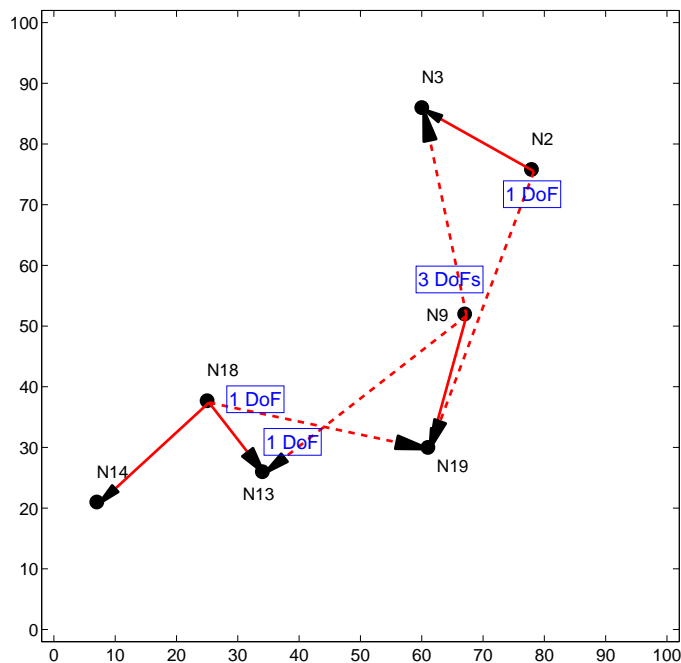


Figure 5.3: The DoF allocation for IC at active nodes in time slot 1 for the 20-node instance under proposed algorithm.

Table 5.1: Location of each node for the 20-node network.

Node	Location	Node	Location
$N_1$	(90.1, 40.2)	$N_{11}$	(10, 36.3)
$N_2$	(77.9, 75.8)	$N_{12}$	(87, 59)
$N_3$	(60, 86)	$N_{13}$	(34.2, 26)
$N_4$	(39, 94)	$N_{14}$	(7, 21)
$N_5$	(80.2, 15)	$N_{15}$	(50.2, 64.8)
$N_6$	(35, 57.3)	$N_{16}$	(18, 83)
$N_7$	(6, 58)	$N_{17}$	(21.2, 10)
$N_8$	(86.9, 93.7)	$N_{18}$	(25, 37.7)
$N_9$	(67, 52)	$N_{19}$	(61, 30)
$N_{10}$	(48, 43.1)	$N_{20}$	(46.8, 9)

### 5.2.2 50-node network with 5 sessions

Now we show the result of a 50-node network instance. The location of each node and the source and destination nodes for each session are shown in Fig. 5.4. Our algorithm can solve it efficiently. The objective value found by our algorithm is 0.25. However, CPLEX cannot even get a feasible solution to this problem in a reasonable amount of time. Figure 5.4 shows the routing topologies and scheduling for each session. Therefore, our algorithm is also efficient under large sized networks.

## 5.3 Complete Results

We now present complete simulation results. We generate 50 network instances for 20-node network with 2 sessions. For each network instance, the location of each node is generated randomly, and the source and destination nodes of each session are randomly selected. We calculate the ratio between the objective value obtained by our algorithm and the optimal objective value found by CPLEX. The average ratio is 85.6%. The detailed results are shown in Table 5.4.

Table 5.2: Details of time slot assignment and number of data streams on each active link for the 20-node network.

Session	Link	Time Slot	DoF for SM
Session 1	$(N_2, N_3)$	1	3
	$(N_2, N_{12})$	2	1
		4	1
	$(N_3, N_4)$	2	2
	$(N_4, N_{16})$	3	2
	$(N_7, N_{11})$	2	2
	$(N_9, N_{19})$	1	1
	$(N_{11}, N_{14})$	4	2
	$(N_{12}, N_9)$	3	2
	$(N_{16}, N_7)$	4	2
	$(N_{17}, N_{14})$	2	1
	$(N_{19}, N_{20})$	2	1
	$(N_{20}, N_{17})$	4	1
Session 2	$(N_6, N_{18})$	3	2
		4	1
	$(N_{14}, N_{13})$	3	1
	$(N_{18}, N_{13})$	1	2
	$(N_{18}, N_{14})$	1	1

Table 5.3: DoF usage at each active node for each time slot in the 20-node network.

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

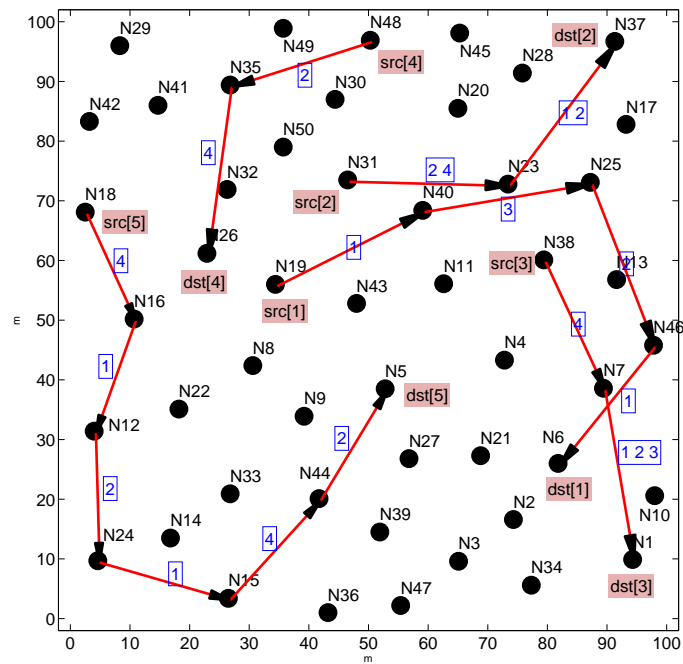


Figure 5.4: The flow routing topology and time slot allocation for the 50-node instance obtained by proposed algorithm.



Table 5.4: Results for 50 instances of 20-node network.

Network Instance	Our Algorithm	CPLEX	Network Instance	Our Algorithm	CPLEX
1	1	1	26	0.5	0.5
2	0.75	1	27	1	1.25
3	1	1.25	28	0.75	1
4	2	2	29	1	1
5	1	1.25	30	0.75	1
6	0.5	0.5	31	0.75	1
7	1	1.25	32	1	1.25
8	0.75	0.75	33	0.75	1
9	0.5	0.75	34	0.75	0.75
10	0.75	0.75	35	0.75	1
11	0.5	0.75	36	2	2
12	1	1.25	37	1	1.125
13	1.25	1.5	38	0.75	1
14	1.25	1.25	39	0.75	1
15	0.5	0.75	40	0.75	0.875
16	1	1.125	41	1	1.125
17	1	1.5	42	0.75	0.875
18	0.75	0.75	43	1.25	1.5
19	0.75	0.75	44	1	1.25
20	0.5	0.75	45	1	1
21	1	1	46	0.75	1
22	1.25	1.25	47	0.5	0.625
23	1.25	1.5	48	2	2
24	0.75	1	49	1.25	1.5
25	1.25	1.5	50	2	2

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

In this thesis, we studied a cross-layer throughput maximization problem for a multi-hop MIMO network. Specifically, we aimed to maximize the minimum session rate among all sessions with joint consideration of flow routing, scheduling and DoF allocation. We employed a new DoF-based model, which can ensure feasible solution and achieve a higher DoF region than previous DoF models. The essence of this model is the systematic rules of DoF allocation for IC based on a node ordering concept.

The throughput maximization problem was formulated as a MILP problem, which cannot be solved efficiently by commercial softwares even for moderate sized networks. Therefore, an efficient polynomial time algorithm was designed by customizing the sequential fixing framework. This algorithm consists of two stages. We obtained a solution by customized SF in stage one and further improved it in stage two. Simulation results show that this algorithm can solve this throughput maximization problem efficiently and provide near-optimal solutions for networks with different sizes.

## 6.2 Future Work

In this thesis, we employed the DoF-based model in [17] and proposed a low-complexity polynomial time algorithm to solve the throughput optimization problem in a multi-hop MIMO network. There are still many directions which we can further explore: (1) We can extend the application of this model to different type of networks, such as cognitive radio network, vehicle network and so on. The proposed algorithm enables this model to solve important problems associated with these networks. (2) We can design a distributed algorithm that is amenable to local implementation to solve this throughput maximization problem.

# Bibliography

- [1] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, *Linear Programming and Network Flows*, Fourth Edition, 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. 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] L.-U. Choi and R.D. Murch, “A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 20–24, Jan. 2004.
- [6] 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.
- [7] *IEEE Transactions on Information Theory—Special Issue on Space-Time Transmission, Reception, Coding and Signal Processing*, Guest Editors: B.M. Hochwald, G. Caire, B. Hassibi, and T.L. Marzetta, vol. 49, no. 10, Oct. 2003.

- [8] Y.T. Hou, Y. Shi, and Hanif.D. Sherali, "Spectrum sharing for multi-hop networking with cognitive radios," *IEEE Journal on Selected Areas in Commun.*, vol. 26, no. 1, pp. 146–155, Jan. 2008.
- [9] *IEEE Journal on Selected Areas in Communications—Special Issue on MIMO Systems and Applications: Field Experience, Practical Aspects, Limitations and Challenges*, Guest Editors: M. Shafi, H. Huang, A. Hottinen, P.J. Smith, and R.A. Valenzuela, vol. 26, no. 6, Aug. 2008.
- [10] S. Jafar and M. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Trans. on Information Theory*, Vol. 53, no. 7, pp. 2637–2642, July 2007.
- [11] S.-J. Kim, X. Wang, and M. Madihian, "Cross-layer design of wireless multihop backhaul networks with multiantenna beamforming," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1259–1269, Nov. 2007.
- [12] J. Liu, Y. Shi, and Y.T. Hou, "A tractable and accurate cross-layer model for multi-hop MIMO networks," in *Proc. IEEE INFOCOM*, 9 pages, San Diego, CA, March 15–19, 2010.
- [13] B. Mumei, J. Tang, and T. Hahn, "Algorithmic Aspects of Communications in Multihop Wireless Networks with MIMO Links," in *Proc. IEEE ICC*, Cape Town, South Africa, July 2010.
- [14] 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.
- [15] *IEEE Journal on Selected Areas in Communications—Special Issue on MIMO Systems and Applications: Part I and Part II*, Guest Editors: M. Shafi, D. Gesbert, D. Shiu, and P.J. Smith, vol. 21, no. 3 & 5, April & June 2003.
- [16] H.D. Sherali and W.P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Chapter 8, Kluwer Academic Publishers, 1999.

- [17] 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*, 9 pages, Shanghai, China, April 10–15, 2011.
- [18] Q.H. Spencer, A.L. Swindlehurst, and M. Haardt, “Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels,” *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 461–471, Feb. 2004.
- [19] 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.
- [20] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, Cambridge, UK, 2005.
- [21] X. Yuan, C. Jiang, Y. Shi, Y.T. Hou, W. Lou, and S. Kompella, “Beyond interference avoidance: On transparent coexistence for multi-hop secondary CR networks,” in *Proc. IEEE SECON*, 9 pages, New Orleans, LA, USA, June 24–27, 2013.
- [22] H. Zeng, Y. Shi, Y.T. Hou, and W. Lou, “An efficient DoF scheduling algorithm for multi-hop MIMO networks,” in *Proc. IEEE INFOCOM*, 9 pages, Turin, Italy, April 14–19, 2013.