

Distributed Stochastic Algorithms for Communication Networks

SHAO, Ziyu

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong

September 2010

Abstract

Designing distributed algorithms for optimizing system-wide performances of large scale communication networks is a challenging task. The key part of this design involves a lot of combinatorial network optimization problems, which are computationally intractable in general and hard to approximate even in a centralized manner. Inspired by the seminal work of Jiang-Walrand, Markov approximation framework was proposed for synthesizing distributed algorithms for general combinatorial network optimization problems. To provide performance guarantees, convergence properties of these distributed algorithms are of significance.

In this thesis, we first review Markov approximation framework and further develop this framework by studying convergence properties of distributed algorithms. These system-wide algorithms consist of the designed Markov chain and resource allocation algorithms. We concentrate on two general scenarios: the designed Markov chain over resource allocation algorithms and resource allocation algorithms over the designed Markov chain. With imprecise measurements of network parameters and without the time-scale separation assumption, we prove convergence to near-optimal solutions for both scenarios under mild conditions. Then we apply Markov approximation framework and associated convergence results to various combinatorial network optimization problems.

First, we consider instances of the designed Markov chain over resource allocation algorithms. We focus on the convergence issues. We find several

examples such that the related convergence results can be applied directly. These examples include optimal path (or tree) selection for wireline networks, optimal neighboring selection for peer-to-peer networks, and optimal channel (or power) assignment for wireless local area networks.

Second, we consider instances of resource allocation algorithms over the designed Markov chain. We focus on the system-wide performances. Two instances are investigated: cross-layer optimization for wireless networks with deterministic channel model and wireless networks with network coding. For both instances, guided by Markov approximation framework, we design distributed schemes to achieve maximum utilities. These schemes include primal-dual flow control algorithms, Markov chain based scheduling algorithms, and routing (or network coding) algorithms. Under time-dependent step sizes and update intervals, we show that these distributed schemes converge to the optimal solutions with probability one. Further, under constant step sizes and constant update intervals, we prove that these distributed schemes also converge to a bounded neighborhood of optimal solutions with probability one. These analytical results are validated by numerical results as well.

Acknowledgments

First, I would like to express my deep gratitude to my supervisor, Professor Shuo-Yen Robert Li, for his invaluable advice, his consistent support and encouragement, and his generous spirit which allows me to explore a spectrum of research problems. He shows me the beauty and power of mathematical tools in solving engineering problems. His broad knowledge and deep insight have a great impact on my research. I think, and I hope, what I have learned from him is not just the math knowledge, but his insights, inspiration, his way of conducting research and living.

I am also sincerely grateful to my co-supervisor, Professor Minghua Chen, for his guidance and support for my research and thesis topic. His enthusiastic attitude towards research quality always drives me to work the best of mine. His willingness and accessibility to discuss technical and non-technical issues with me made my student life much easier and more enjoyable.

I would like to thank Professor Jean Walrand, Professor Sidharth Jaggi, Professor Anthony Man-Cho So, and Professor Angela Yingjun Zhang for the many enlightening discussions we have.

Many thanks to my friends at CUHK who make my life and study cheerful in the past years: Qiang Zhu, Min Tan, Tyler Sun, Vincent Wang, Henry Xu, Jennifer Wu, Shenghao Yang, Tong Liang, and Shaoquan Zhang.

Finally, I would like to thank for my parents, my brother and my girlfriend Helen, for their endless love and support. I dedicate this thesis to them.

Contents

1	Introduction	1
1.1	Convex Network Optimization Problems	2
1.2	Combinatorial Network Optimization Problems	4
1.3	Thesis Outline	7
2	Markov Approximation Framework	11
2.1	Settings	11
2.2	Basic Framework	13
2.2.1	Step 1: Log-sum-exp Approximation	13
2.2.2	Step 2: Distributed Markov Chain Monte Carlo	15
2.3	Networks without Time-scale Separation Assumption	19
2.3.1	Resource Allocation Algorithms Over Markov Chain	21
2.3.2	Markov Chain Over Resource Allocation Algorithms	26
2.4	Connection to Statistical Physics	33
2.5	Appendix of Chapter 2	34
2.5.1	Proof of Lemma 2.3	34
2.5.2	Proof of Theorem 2.4	37
2.5.3	Proof of Lemma 2.12	41
2.5.4	Proof of Theorem 2.6	42
2.5.5	Proof of Theorem 2.7	45
2.5.6	Proof of Theorem 2.9	51

2.5.7	Proof of Theorem 2.10	55
3	Wireless Networks With Deterministic Channel Models	57
3.1	Introduction	58
3.2	System Model and Problem Formulation	61
3.2.1	Deterministic Channel Model	61
3.2.2	Conflict Graph Based Model	63
3.2.3	Extended Conflict Graph Model vs. Existing Conflict Graph Model.	65
3.2.4	Feasible Rate Region	67
3.3	NUM Over General Multi-Hop Network: Link-Centric Formu- lation	69
3.3.1	Markov Approximation	71
3.3.2	Design and Implementation of Markov Chain	72
3.3.3	Solving the Approximated Problem by the Primal-Dual Algorithm	74
3.3.4	Convergence of Stochastic Primal-Dual Algorithm	77
3.4	NUM Over General Multi-Hop Network: Node-Centric Formu- lation	79
3.4.1	Design and Implementation of Markov Chain	81
3.4.2	Solving the Approximated Problem by the Primal-Dual Algorithm	81
3.4.3	Convergence Properties	83
3.5	Numerical Examples	84
3.5.1	Link-centric Formulation	84
3.5.2	Node-centric Formulation	89
3.6	Conclusions	92
3.7	Appendix of Chapter 3	93
3.7.1	Proof of Proposition 3.4	93

3.7.2	Proof of Proposition 3.7	95
3.7.3	Proof of Theorem 3.8	96
4	Wireless Networks With Network Coding	98
4.1	Introduction	98
4.2	Related Work	101
4.3	Preliminaries	103
4.3.1	Network Model	103
4.3.2	Network Coding	104
4.4	Intra-session Network Coding	105
4.4.1	Wireless NUM	105
4.4.2	Markov Approximation	109
4.4.3	Design and Implementation of Markov Chain	110
4.4.4	Solving the Approximated Problem by the Primal-Dual Algorithm	111
4.4.5	Convergence Properties of Primal-dual Algorithms	114
4.5	Inter-session Network coding	116
4.5.1	Wireless NUM	117
4.5.2	Markov Approximation	120
4.5.3	Convergence Properties	122
4.6	Numerical Examples	124
4.6.1	Multicast Sessions With Intra-session Network Coding	124
4.6.2	Unicast Sessions With Inter-session Network Coding	128
4.7	Conclusions	129
4.8	Appendix of Chapter 4	132
4.8.1	Proof of Proposition 4.1	132
4.8.2	Proof of Theorem 4.4	133
5	Conclusions	138

List of Figures

2.1	The Markov chains in (b), (c), (d), by adding/removing transition edge-pair between two states in the time-reversible Markov chain in (a), are also time-reversible. All Markov chains have the same stationary distribution.	17
2.2	An example of the original three-state topology hopping Markov chain and the extended Markov chain. M is the original configuration hopping Markov chain with accurate system performance. M' is the corresponding extended Markov chain with inaccurate system observations. For each configuration $f \in \{1, 2, 3\}$, the observed system performance takes values $\phi(f) - \Delta_f$, $\phi(f)$, $\phi(f) + \Delta_f$ with probability η_{f-1} , η_{f_0} and η_{f_1} respectively. The transition rates are assigned according to (2.31) and (2.32).	29
3.1	One example of deterministic channel based wireless network with channel gains $\rho_{AR} = 2$ and $\rho_{BR} = 1$. Each rectangle represents a transmitter or receiver node, and each knob attaching to a rectangle represents a signal level and what is sent on it is a bit.	62
3.2	Corresponding conflict graph for the network shown in Fig. 1. All independent sets are: \emptyset , $\{a_1\}$, $\{a_2\}$, $\{b_1\}$, $\{a_1, a_2\}$, and $\{\{a_1\}, \{b_1\}\}$	65

- 3.3 A deterministic wireless network with four links AB , BC , CD and DE . Each link has three sublinks. There is a flow trans-
verses from A to E . By the extended conflict graph model,
each link maintains six link configurations and there is only one
four-hop path. In contrast, by the existing conflict graph model
on sublinks, each link maintains three sublinks and there are at
least eighty-one four-hop paths. 66
- 3.4 Example of wireless networks with broadcast advantages. For
transmission initiated by a_1 , its receiver is c_3 , for transmission
initiated by a_2 , its receiver is c_4 , and for transmission initiated
by b_1 , its receivers are c_4 and d_4 . Possible transmissions: $(a_1,$
 $\{c_3\})$, $(a_2, \{c_4\})$, $[(a_1, \{c_3\}), (a_2, \{c_4\})]$, $(b_1, \{c_4\})$, $(b_1, \{d_4\})$
and $(b_1, \{c_4, d_4\})$ 67
- 3.5 Corresponding conflict graph for the network shown in Fig. 3.
All independent sets are: \emptyset , $\{(a_1, \{c_3\})\}$, $\{(a_2, \{c_4\})\}$, $\{(a_1, \{c_3\}), (a_2, \{c_4\})\}$,
 $\{(b_1, \{c_4\})\}$, $\{(b_1, \{d_4\})\}$, $\{(b_1, \{c_4, d_4\})\}$, $\{(a_1, \{c_3\}), (b_1, \{c_4\})\}$,
 $\{(a_1, \{c_3\}), (b_1, \{d_4\})\}$, and $\{(a_1, \{c_3\}), (b_1, \{c_4, d_4\})\}$ 68
- 3.6 [6]. Subfig 1 shows a 2x2 deterministic interference channel
with channel gains $\rho_{T_1R_1} = \rho_{T_2R_2} = 4$ and $\rho_{T_1R_2} = \rho_{T_2R_1} = 3$.
Subfig 2 shows a coding scheme achieving the rate tuple $(3,2)$,
where T_1 sends a, b, b, c to R_1 and T_2 sends $d, \emptyset, \emptyset, e$ to R_2 . R_1
can decode and obtain a, b, c , while R_2 can decode and obtain
 d, e 69
- 3.7 A wireless deterministic network with channel gains $\rho_{AB} =$
 $\rho_{BC} = \rho_{DE} = \rho_{EF} = 3$, $\rho_{AE} = \rho_{DB} = \rho_{BF} = \rho_{EC} = 2$,
and $\rho_{CG} = \rho_{FG} = 1$. Four users are associated with paths
 $A \rightarrow B \rightarrow C$, $B \rightarrow C \rightarrow G$, $D \rightarrow E \rightarrow F$, and $E \rightarrow F \rightarrow G$
respectively. 86

3.8	Performance of the primal-dual algorithm on user rates with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Initial values of user rates are all 0 . Because of symmetricity, not only rate of user 1 and rate of user 3 evolves nearly in the same way, but also rate of user 2 and rate of user 4 evolves nearly in the same way.	86
3.9	Performance of the primal-dual algorithm on link prices with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Initial values of link prices are all 0 . Because of symmetricity, price pairs such as price of Link AB and price of Link DE, price of Link BC and price of Link EF, price of Link CG and price of Link FG, evolve nearly in the same way respectively.	87
3.10	A wireless butterfly network with channel gains $\rho_{s_1s_3} = \rho_{s_1t_1} = 1, \rho_{s_2s_3} = 2, \rho_{s_2t_2} = 1, \rho_{s_3t_3} = 1$, and $\rho_{t_3t_2} = \rho_{t_3t_1} = 1$. There are three users (source-destination pairs): user 1 (s_1, t_1), user 2 (s_2, t_2) and user 3 (s_3, t_3).	89
3.11	Performance of the primal-dual algorithm on user rates with $\beta = 100, T_0 = 100, \epsilon = 0.05$. Initial values of all user rates are 0 . All user rates converge to the neighborhood of 1 , the optimal value.	90
3.12	Performance of the primal-dual algorithm on node prices with $\beta = 100, T_0 = 100, \epsilon = 0.05$. Initial values of all node prices for user 1, 2, 3 are 0 . For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are 0 all the time.	91
4.1	Examples of network coding for both wireline and wireless networks.	104

4.2	A wireless butterfly network with three hyperlinks, $(A, \{C, D\})$, $(B, \{C, E\})$, and $(C, \{D, E\})$. All hyperlinks have one unit capacity. There are two multicast sessions, session 1 includes source A multicasting to destinations D and E, and session 2 includes source B multicasting to destinations D and E.	126
4.3	Performance of the primal-dual algorithm on multicast session rates with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all session rates are θ	126
4.4	Performance of the primal-dual algorithm on node prices with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all node prices are θ . For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are θ all the time. Note that all node prices are stable despite some oscillations. . .	127
4.5	A wireless butterfly network with three hyperlinks, $(A, \{C, D\})$, $(B, \{C, E\})$, and $(C, \{D, E\})$. All hyperlinks have one unit capacity. There are two unicast sessions, session 1 includes source A unicasting to destination E, and session 2 includes source B unicasting to destination D.	129
4.6	Performance of the primal-dual algorithm on multicast session rates with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all session rates are θ	130
4.7	Performance of the primal-dual algorithm on node prices with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all node prices are θ . For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are θ all the time. Note that all node prices are stable despite some oscillations. . .	131

List of Tables

3.1	Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Here 'TIAN' denotes 'Treating Interference As Noise'	87
3.2	Performance comparison with $T_0 = 100, \epsilon = 0.01$	88
3.3	Performance comparison with $\beta = 100, \epsilon = 0.01$	88
3.4	Performance comparison with $\beta = 100, T_0 = 100$	88
3.5	Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$	90
3.6	Performance comparison with $T_0 = 100, \epsilon = 0.05$	91
3.7	Performance comparison with $\beta = 100, \epsilon = 0.05$	91
3.8	Performance comparison with $\beta = 100, T_0 = 100$	92
4.1	Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$	127
4.2	Performance comparison with $T_0 = 100, \epsilon = 0.05$	128
4.3	Performance comparison with $\beta = 100, \epsilon = 0.05$	128
4.4	Performance comparison with $\beta = 100, T_0 = 100$	128
4.5	Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$	129
4.6	Performance comparison with $T_0 = 100, \epsilon = 0.05$	129
4.7	Performance comparison with $\beta = 100, \epsilon = 0.05$	130
4.8	Performance comparison with $\beta = 100, T_0 = 100$	130

Chapter 1

Introduction

There is nothing more practical
than a good theory.

Kurt Lewin

Communication networks are constantly evolving and have greatly changed the way we communicate, work, and live. The Internet is a globally shared information superhighway over which people can publish and share information. Peer-to-peer networks enable people to distribute large files of data, music and video over Internet in a faster and robust manner. Wireless communication networks enable us to communicate with anyone and access the global Internet anytime and anywhere. The emerging sensor networks are expected to revolutionize many segments of our society and lives, from manufacturing to transportation, from environmental monitoring to home automation.

As communication networks continue to grow in size, complexity, and diversity, there is a pressing need for theory and algorithms which offer the fundamental knowledge and tools necessary to analyze and design large, complex networks in such a way that their performance can be predicted, controlled, and optimized.

As opposed to traditional algorithm designs focusing on centralized solutions with polynomial-time computational complexity, algorithm design for

modern communication networks prefers distributed solutions, where each network element (computer, sensor, cell phone, ...) in networks runs the algorithm individually and exchanges information with other network elements locally. Any centralized solution requires a central network element to collect and maintain global information from all network elements, which is impossible in practice, especially for large scale networks.

Existing research work on distributed network optimization can be divided into two categories: convex network optimization and combinatorial network optimization.

1.1 Convex Network Optimization Problems

Convex network optimization problems are network designs that can be formulated as problem instances of convex optimization.

For convex network optimization problems, we can design effective and distributed solutions by exploiting the structure information of network and making decomposition [12, 35]. What is more, by Kelly's seminal work [35] and its follow-ups [12, 51], a large class of existing network protocols can be reverse engineered to be viewed as distributed solutions for some hidden convex network optimization problems. In this way, we can better understand the protocols and further improve these protocols. Take the congestion control for TCP (Transmission Control Protocol) as an example.

Congestion control in the Internet was introduced in the late 1980s by Van Jacobson [26]. Jacobson's algorithm which is implemented in the transport layer protocol called TCP (Transmission Control Protocol) has served the Internet well during a time of unprecedented growth. However, the algorithm was designed during a time when the Internet was a relatively small network compared to its size today. Therefore, there has been much interest in reexamining the role of congestion control in the Internet with the goal of enhancing

TCP to make it scalable to large networks.

The traditional method for enhancing TCP is largely based on intuition, heuristics, simulations, and experiments. However, design based on intuition can easily underestimate the importance of certain system features and lead to a suboptimal solution, or even disastrous implementation. On the other hand, formulating theories for such a complex heuristic system afterwards seems infeasible at the first glance, which is partly the reason why theory for TCP is lagging far behind its applications. Therefore, developing theory for the Internet is very important and challenging, as the design and analysis of protocols needs a rigorous framework. Recently, a unified framework to study Internet congestion control has been proposed [12, 35, 51], where the TCP source rates are viewed as primal variables, congestion measures are viewed as the dual variables, and any TCP congestion control algorithm can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to solve the utility maximization problem (maximize aggregate user's utility). The network utility maximization problem is given as follows:

$$\begin{aligned} \text{NUM: } \max_{\mathbf{x} \geq 0} \quad & \sum_{s \in S} U_s(x_s) \\ \text{s.t.} \quad & R\mathbf{x} \leq C. \end{aligned}$$

where S is the set of users, \mathbf{x} is the vector of user's rate, $U_s(x_s)$ is the utility of user $s \in S$ with rate x_s , R is a routing matrix and C is the vector of link capacities. The constraint says that, for each link, the aggregate source rate over this link does not exceed the link capacity.

The utility function measures the efficiency and fairness of algorithms. We assume the utility function to be twice differentiable, increasing and strictly concave. For example, the α -fairness utility function [53] is defined as follows:

$$U^\alpha(x) = \begin{cases} \frac{x^{1-\alpha}}{1-\alpha} & \text{if } \alpha \neq 1 \\ \log x & \text{Otherwise,} \end{cases}$$

where $\alpha = 1$ corresponds to proportional fairness and $\alpha \rightarrow \infty$ corresponds to max-min fairness.

A user's utility function is (often implicitly) defined by its TCP algorithm. For example, the existing TCP protocols such as TCP-Reno and TCP-Vegas have been reverse-engineered to determine the underlying utility functions [51]. Further, the equilibrium properties of a TCP system, such as throughput and fairness, can be readily understood by studying the corresponding optimization problems with these utility functions.

Based on this network utility maximization(NUM) framework, there are a large body of research in congestion control and effective implementations for TCP in practice. More applications of the NUM framework for the design and analysis of network protocols can be found in [12].

In summary, for convex network optimization problems, we can design distributed solutions guided by the NUM framework.

1.2 Combinatorial Network Optimization Problems

Combinatorial optimization is concerned with the arrangement, grouping, ordering, or selection of discrete objects from a finite set. Many combinatorial optimization problems involve seeking a best configuration of a set of parameters to achieve desired objectives. The configurations can be sequences, permutations, graphs and partitions.

Combinatorial optimization problems arise in all subfields of communication networks, for instance, wired local area networks, Internet, peer-to-peer

(P2P) networks, wireless ad-hoc networks, wireless mesh networks, sensor networks, mobile communication systems, and satellite networks. In communication networks, combinatorial optimization is used in network design and management to meet various operational needs. Classical applications of combinatorial optimization in communication networks trace back to the use of shortest paths in the Internet routing and spanning trees in the bridged LAN(local area network) configuration. More example network problems covered include infrastructure deployment, media access control, routing optimization, topology control, resource allocation, light-path establishment in optical networks, scheduling and QoS(quality of service) provisioning in various wireless networks.

Combinatorial network optimization addresses problems at the intersection of the two areas: combinatorial optimization and communication networks. Combinatorial network optimization usually involves mathematical modeling of the problem, with the objective of reducing network deployment cost or operation cost, to provide better quality of service, or to improve the network performance. As opposed to convex network optimization problems, there are a large class of combinatorial network optimization problems, for which effective solutions do not exist. These combinatorial network optimization problems often suffer from two shortcomings: (i) the optimization problem could be intractable when the network size is large (i.e., it is NP-hard); (ii) the optimization problem could be amenable to centralized implementation only. Examples include throughput-optimal scheduling for wireless ad-hoc networks, channel assignment for wireless mesh networks, neighbor selection for P2P networks, and path selection for both wired and wireless networks.

Even if until now no proof is known that NP-hard problems cannot be solved by means of algorithms that run in polynomial time, there is a strong evidence that such algorithms may not exist [56]. Considerable effort has been devoted to constructing and investigating methods for solving to optimality

or proximity combinatorial optimization problems [40, 43, 56, 64]. Well-known examples are enumeration methods using cutting plane, branch-and-bound, branch-and-cut, or dynamic programming [64]; general heuristic algorithms using tabu search, simulated annealing, artificial neural networks or genetic algorithms [25, 42], and other tailored heuristics using problem-specific information [64].

However, each of these algorithms suffers from at least one of the following shortcomings:

- Dependence on initial settings.
- No performance guarantees .
- Bounded above or below from the optimal solutions. Cannot achieve performance that is arbitrarily close-to-optimal.
- Not amenable to distributed solutions.
- Distributed solutions with high control overhead, e.g, heavy message exchanges.
- Apply to special cases only.

Therefore, we need new insights into mathematical structures of combinatorial network optimization problems, which lead to novel approaches and efficient solution techniques.

Recently, there is a breakthrough for throughput-optimal scheduling for wireless ad-hoc networks [31]. The proposed adaptive CSMA (Carrier Sensing Multiple Access) algorithms have been shown to be an effective solution. However, for general combinatorial network optimization problems, there is still a lack of theoretical frameworks for effective solutions.

Motivated by the seminal result [31] and its follow ups, the Markov approximation framework [10] is proposed as one design option of such framework.

By this framework, we can design distributed stochastic algorithms for a large class of combinatorial network optimization problems arising in communication networks. In these algorithms, each network element only uses its local information. No or very few explicit communication or control messages are required among the nodes. Moreover, systems running distributed algorithms, compared with those running centralized algorithms, are more adaptable to users joining and leaving the systems (e.g., peer churn in peer-to-peer systems) and are more robust to system dynamics (e.g., channel fading in wireless networks). In theory, these distributed algorithms can achieve close-to-optimal values, though we do not have guarantees of the polynomial-time running time.

In practice, these system-wide distributed algorithms consist of two parts: resource allocation algorithms and Markov chain based scheduling algorithms. Therefore, there are two sequences of events that interact with each other: trajectories of resource allocation algorithms, and network dynamics induced by Markov chain designed within the Markov approximation framework. Further, in practice, obtained network parameters are always noisy (imprecise), and there is no time-scale separation assumption. Then an important question arises naturally: do system-wide algorithms converge or not? The theoretical understanding of this question can help to predict algorithm performance and ultimately design new and improved algorithms.

1.3 Thesis Outline

In summary, the material presented in this thesis can be subdivided into two main categories: on one hand, the design of algorithms by following Markov approximation framework; and on the other hand, further developing the Markov approximation framework by studying convergence properties of combinations of Markov chain and Resource allocation algorithms.

The rest of this thesis and main results are provided as follows:

In chapter 2, we first review the basics of Markov approximation framework, i.e., log-sum-exp approximation and distributed Markov chain Monte Carlo (MCMC). We are concerned with convergence property of system-wide algorithms. We focus on the scenario of trajectories of resource allocation algorithms and network dynamics induced by Markov chain interacting with each other. With imprecise measurement of network parameters and without time-scale separation assumption, we prove convergence properties of system-wide algorithms under mild conditions for two general scenarios: the designed Markov chain over resource allocation algorithms and resource allocation algorithms over the designed Markov chain. The proof technique for the former scenario is inspired by the one used in Jiang-Walrand's paper [30].

As a consequence, we consider application of Markov approximation framework and associated convergence results. First, we consider instances of the designed Markov chain over resource allocation algorithms. We focus on the theoretical issues of convergence. We find some examples for which the related convergence result can be applied, including optimal path (or tree) selection for wireline networks, optimal neighboring selection for peer-to-peer networks, and optimal channel (or power) assignment for wireless local area networks.

Then in the following two chapters, we consider instances of resource allocation algorithms over the designed Markov chain. We provide detailed analysis of these instances.

In Chapter 3, we present results on cross-layer design of wireless networks with deterministic channel models. first, we extend the well-studied conflict graph model to capture the flow interactions over the deterministic channels and characterize the feasible rate region. Then we consider general multi-hop wireless networks with both a link-centric formulation and a node-centric formulation. We approximately solve the problem in a distributed manner by Markov approximation framework, including the scheduling algorithm and primal-dual flow-control algorithms. The whole system is shown to converge to

the optimal solutions without the time-scale separation assumption. Further, we show the convergence to a bounded neighborhood of optimal solutions with probability one under constant step sizes and constant update intervals. Numerical results illustrate not only the advantage of deterministic channel model over simple physical model such as treating interference as noise, but also the convergence and optimality of our joint scheduling and primal-dual flow control algorithms.

In Chapter 4, we present results on cross-layer design of wireless networks with network coding. We consider both multicast sessions with intra-session network coding and unicast sessions with suboptimal inter-session network coding. For intra-session network coding scenario, we adopt the random linear network coding algorithm; while for inter-session network coding algorithm, we adopt local butterfly structure based session decomposition scheme, i.e., a sub-optimal coding scheme. Then we exploit the wireless broadcast advantage and develop a unified distributed scheme guided by Markov approximation framework, including the scheduling algorithm, back-pressure algorithm, network coding algorithm and primal-dual flow-control algorithm. The convergence result of the whole system is similar to the scenario of wireless networks with deterministic channel. Further, numerical results validate the analytical results. Actually the proposed distributed scheme and related convergence results can be applied to other inter-session network coding scenarios by replacing session decomposition scheme with other sub-optimal coding schemes.

In Chapter 5, we conclude this thesis and point out several future research directions. First, we consider the distributed design for optimal selection of paths, trees, neighbors, channels, and power levels for wireless or wireline (P2P) networks without message passing among network nodes or links. Second, we consider the mixing time problem of the designed Markov chain, i.e., how to design Markov chain so that it converges to its stationary distribution quickly? Third, we want to find more instances of combinatorial network

optimization problems, exploit the structural information of these instances, and design distributed stochastic algorithms guided by Markov approximation framework.

Chapter 2

Markov Approximation

Framework

A good theory should shrink rather than grow the knowledge tree.

Robert G. Gallager

In this chapter, we will introduce the basic principle of Markov approximation framework. Then we discuss the convergence properties of distributed schemes designed within the Markov approximation framework. We also discuss the connection to statistical physics.

2.1 Settings

Consider a network with a set of users S , and a set of configurations \mathcal{F} . A network configuration $f \in \mathcal{F}$ consists of individual users using one of its local configurations. When the network system operates under f , each user obtains certain performance, denoted by $\phi_s(f)$ ($s \in S$). Note that $\phi_s(f)$ can be some direct system measurement, e.g., throughput, under configuration f , or a function of the measurement. Let $\phi(f)$ denote the the system performance under configuration f . For example, $\phi(f)$ can be aggregate user's performance under configuration f , i.e., $\phi(f) = \sum_{s \in S} \phi_s(f)$.

The problem of maximizing the system performance by choosing the best configuration can then be cast as following combinatorial optimization problem called maximum-weight configuration problem (MWC):

$$\mathbf{MWC} : \max_{f \in \mathcal{F}} \phi(f) \quad (2.1)$$

An equivalent formulation is

$$\begin{aligned} \mathbf{MWC} - \mathbf{EQ} : \max_{p \geq 0} & \sum_{f \in \mathcal{F}} p_f \phi(f) \\ \text{s.t.} & \sum_{f \in \mathcal{F}} p_f = 1, \end{aligned} \quad (2.2)$$

where p_f is the time fraction (or probability) of the configuration h , i.e., the percentage of time the configuration f is in use.

Many practical and important problems, or their subproblems, can be formulated into the form of (2.1). We list some examples as follows [10]:

- CSMA network design problem: the configuration f is a independent set, i.e., collection of non-interfered links.
- Path selection problem: the configuration f is one combination of selected paths.
- Channel assignment problem: the configuration f is one combination of assigned wireless channels.

More examples will be given in the following chapters.

For many problems, formulation in (2.1) could be very challenging to solve, even in a centralized manner. For example, when \mathcal{F} is the set of all independent sets in a general conflict graph, and $\phi(f)$ is the weight of the independent set f , we obtain the MWIS (Maximum Weight Independent Set) problem. MWIS is NP-hard and hard to approximate even in a centralized way [64]. In practice, it is often acceptable to solve the problem approximately, but in a distributed

manner. Systems running distributed algorithms are more robust to user and system dynamics than those running centralized algorithms.

In the following, we describe the Markov approximation framework, to approach problem in (2.1). It often leads to distributed stochastic algorithms that can be implemented in practice with limited or no message passing among users.

2.2 Basic Framework

The essence of Markov approximation framework [10] can be summarized as two steps: *log-sum-exp* function approximation for the *max* function and distributed *Markov Chain Monte Carlo* methods to construct Markov Chains with the desired stationary distributions.

2.2.1 Step 1: Log-sum-exp Approximation

We use the *log-sum-exp* function to approximate the *max* function smoothly, shown as follows:

$$\max_{f \in \mathcal{F}} \phi(f) \approx \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right) \quad (2.3)$$

where β is a positive constant. Let $|\mathcal{F}|$ denote the size of the set \mathcal{F} , then the approximation accuracy is known as follows:

Proposition 2.1.

$$\max_{f \in \mathcal{F}} \phi(f) \leq \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right) \leq \max_{f \in \mathcal{F}} \phi(f) + \frac{1}{\beta} \log |\mathcal{F}| \quad (2.4)$$

Hence,

$$\max_{f \in \mathcal{F}} \phi(f) = \lim_{\beta \rightarrow \infty} \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right) \quad (2.5)$$

Proof. Given $\beta > 0$, we have

$$\exp(\max_{f \in \mathcal{F}} \beta \phi(f)) \leq \sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \leq |\mathcal{F}| \exp(\max_{f \in \mathcal{F}} \beta \phi(f)).$$

So

$$\max_{f \in \mathcal{F}} \beta \phi(f) \leq \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right) \leq \max_{f \in \mathcal{F}} \beta \phi(f) + \log(|\mathcal{F}|)$$

By dividing β in both sides, we obtain the inequality (2.4). When $\beta \rightarrow \infty$, we get the desired equality (2.5). \square

We have some important observations in the following proposition.

Proposition 2.2. $\frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right)$ is the optimal value of the following optimization problem

$$\begin{aligned} \max_{\mathbf{p} \geq 0} \quad & \sum_{f \in \mathcal{F}} p_f \phi(f) - \frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f \\ \text{s.t.} \quad & \sum_{f \in \mathcal{F}} p_f = 1. \end{aligned} \tag{2.6}$$

and the corresponding optimal solution is

$$p_f^*(\phi) = \frac{\exp[\beta \phi(f)]}{\sum_{f' \in \mathcal{F}} \exp[\beta \phi(f')]}, \quad \forall f \in \mathcal{F}, \tag{2.7}$$

where $\phi \triangleq [\phi(f), f \in \mathcal{F}]$.

Proof. Let

$$g_\beta(\phi) \triangleq \frac{1}{\beta} \log \left(\sum_{f \in \mathcal{F}} \exp[\beta \phi(f)] \right). \tag{2.8}$$

Then the conjugate function of $g_\beta(\phi)$ is given by

$$g_\beta^*(\mathbf{p}) = \begin{cases} \frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f & \text{if } \mathbf{p} \geq 0 \text{ and } \mathbf{1}^T \mathbf{p} = 1 \\ \infty & \text{otherwise.} \end{cases} \tag{2.9}$$

Further, the conjugate function of $g_\beta^*(\mathbf{p})$ is given by

$$g_\beta^{**}(\boldsymbol{\phi}) \triangleq \max_{\mathbf{p} \geq 0} \sum_{f \in \mathcal{F}} p_f \phi(f) - \frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f \quad (2.10)$$

$$\text{s.t. } \sum_{f \in \mathcal{F}} p_f = 1.$$

$g_\beta(\boldsymbol{\phi})$ is a convex function because the log-sum-exp function is a convex function [5]. Further, $g_\beta(\boldsymbol{\phi})$ is continuous, and its domain is a closed set, thus $g_\beta(\boldsymbol{\phi})$ is a closed function. Hence by [5, Section 3.3.2], the conjugate of its conjugate $g_\beta^*(\mathbf{p})$ is itself, i.e., $g_\beta(\boldsymbol{\phi}) = g_\beta^{**}(\boldsymbol{\phi})$.

On the other hand, by the KKT(Karush-Kuhn-Tucker) conditions [5] for problem (2.6), we obtain the desired probability distribution (2.7). \square

By time-sharing among different configurations h according to their portions $p_f^*(\boldsymbol{\phi})$, we can solve the problem **MWC – EQ**, and hence the problem **MWC**, approximately. We remark that the optimality gap is bounded by $\frac{1}{\beta} \log |\mathcal{F}|$, which can be made small by choosing large β . Note that the approximation becomes exact as β approaches infinity. However, usually there are practical constraints or overhead concerns of using large β [10]. Therefore, by the *log-sum-exp* approximation in (3.6), we are implicitly solving an approximated version of the problem **MWC – EQ**, off by an *entropy* term $-\frac{1}{\beta} \sum_{f \in \mathcal{F}} p_f \log p_f$. Under the context of CSMA scheduling, [31] arrives at similar observations using a different approach.

2.2.2 Step 2: Distributed Markov Chain Monte Carlo

We need to to construct Markov Chains with the stationary distributions $p_f^*(\boldsymbol{\phi})$, $f \in \mathcal{F}$ in a distributed way, i.e., distributed *Markov Chain Monte Carlo* methods. The basic idea is to define an irreducible time-reversible Markov chain whose set of states is the configuration space \mathcal{F} and whose stationary distribution is $p_f^*(\boldsymbol{\phi})$, $f \in \mathcal{F}$. As the Markov chain converges to its stationary distribution, we approach $p_f^*(\boldsymbol{\phi})$, $f \in \mathcal{F}$ in a distributed manner. Time-reversible

Markov chains usually have structures that allows distributed implementation. This opens a new design space for exploration. Actually we have the following result:

Lemma 2.3. *For any probability distribution of the product form $p_f^*(\phi)$, $f \in \mathcal{F}$ in (2.7), there exists at least one continuous-time time-reversible ergodic Markov chain whose stationary distribution is $p_f^*(\phi)$, $f \in \mathcal{F}$. Further, for any continuous-time time-reversible ergodic Markov chain, its stationary distribution can be expressed by the product form $p_f^*(\phi)$, $f \in \mathcal{F}$ in (2.7).*

The proof is relegated to Appendix 2.5.1.

To construct a time-reversible Markov chain with its stationary distribution $p_f^*(\phi)$, $f \in \mathcal{F}$, we let $f \in \mathcal{F}$ be the state of the Markov chain, and denote $q_{f,f'}$ as the transition rate between two states f and f' . It suffices to design $q_{f,f'}$ so that

- the resulting Markov chain is irreducible, i.e., any two states are reachable from each other,
- and the detailed balance equation is satisfied: for all f and f' in \mathcal{F} and $f \neq f'$, $p_f^*(\phi)q_{f,f'} = p_{f'}^*(\phi)q_{f',f}$.

We remark that the above two sufficient requirements allow a large degree of freedom in design.

First, it allows us to cut off direct transition between any two states, given that they are still reachable from other states. The modified Markov chain is still time-reversible and its stationary distribution is still $p_f^*(\phi)$, $f \in \mathcal{F}$. For example, assume the 4-states Markov chain in Fig. 2.1.(a) is time-reversible. The “sparse” Markov chains in Fig. 2.1.(b)-(d), modified from the one in Fig. 2.1.(a), are also time-reversible. Furthermore, all Markov chains share the same stationary distribution.

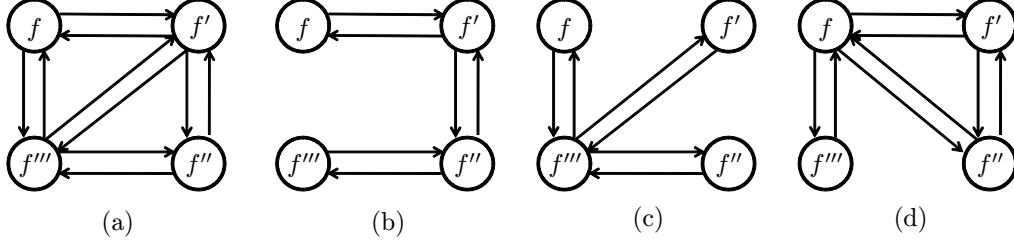


Figure 2.1: The Markov chains in (b), (c), (d), by adding/removing transition edge-pair between two states in the time-reversible Markov chain in (a), are also time-reversible. All Markov chains have the same stationary distribution.

Second, for two states f and f' that have direct transitions, there are many options in designing $q_{f,f'}$ and $q_{f',f}$. For example, when $\phi(f) = \sum_{s \in S} \phi_s(f)$, we have $\exp(\beta \sum_{s \in S} \phi_s(f)) q_{f,f'} = \exp(\beta \sum_{s \in S} \phi_s(f')) q_{f',f}, \forall f, f' \in \mathcal{F}$. Corresponding design options include, but are not limited to, the following ones: let α be a positive constant,

OPT1: let $q_{f,f'}$ be negatively correlated to the system performance $\sum_{s \in S} \phi_s(f)$ under configuration f , specifically,

$$q_{f,f'} = \alpha \left[\exp \left(\beta \sum_{s \in S} \phi_s(f) \right) \right]^{-1}. \quad (2.11)$$

$q_{f',f}$ is defined in a symmetric way.

OPT2: let $q_{f,f'}$ be positively correlated to the system performance under the targeting configuration f' :

$$q_{f,f'} = \alpha \exp \left(\beta \sum_{s \in S} \phi_s(f') \right). \quad (2.12)$$

$q_{f',f}$ is defined in a symmetric way.

OPT3: let $q_{f,f'}$ be positively correlated to the difference of system performance under configurations f and f' :

$$q_{f,f'} = \alpha \exp \left(\frac{1}{2} \beta \sum_{s \in S} (\phi_s(f') - \phi_s(f)) \right). \quad (2.13)$$

$q_{f',f}$ is defined in a symmetric way.

OPT4: let $q_{f',f} = \alpha$, and $q_{f,f'}$ be positively correlated to the difference of system performance under configurations f and f' , i.e.,

$$q_{f,f'} = \alpha \exp \left(\beta \sum_{s \in S} (\phi_s(f') - \phi_s(f)) \right). \quad (2.14)$$

Design option OPT1 implies the transition rate from f to f' , i.e., $q_{f,f'}$, is *independent* of the performance under the targeted configuration f' . In contrast, $q_{f,f'}$ in OPT2 depends only on the performance of targeting configuration f' . Design of $q_{f,f'}$ in OPT3 combines flavors from previous two options, where the system is more likely to switch to a configuration with better performance. In practice, both OPT2 and OPT3 require the system to know the performance under targeting configuration f' a priori, or through a probing phase. Option OPT4 is similar to OPT3, but $q_{f,f'}$ and $q_{f',f}$ are no longer symmetric. Actually CSMA protocol in fact implements a Markov chain with transition rate fitting into option OPT4 [10].

Recall that in our setting, a configuration f consists of each individual user using one of its local configurations. Transitions between f and f' are done via users switching their local configurations accordingly. By users running individual continuous-time clock and wait for a random amount of time before switching local configurations, we can design transition to happen only between two configurations f and f' that differ by one user's local configuration. If individual users can collect system performance that its configuration-switching affects in a distributed manner, then the Markov chain can be implemented distributedly.

2.3 Networks without Time-scale Separation

Assumption

From above, we know that in order to apply the Markov approximation framework for combinatorial network optimization problems, a number of items have to be defined: a set of configurations, a neighboring structure of configuration space, a generation mechanism of Markov chain for configuration transitions, and a weight function (or equivalently a cost function).

In this process, we can see that two sequences of events coupled together. One is the network dynamics evolve over time modeled and induced by the designed Markov chain, e.g, service rate, channel state, buffer size, and network topology. The other is certain proposed algorithms updating resource allocation variables, e.g., source rates, link prices, channel contention probabilities, channel holding times, transmit powers, and routes. Note that the generator (the collection of transition rates) of the Markov chain evolves in time as it depends on the parameters updated by the resource allocation algorithm. Therefore, algorithmic updates change the network dynamics, which in turn changes the trajectory of the algorithm, thus forming a loop that couples the two sequences of events. There are numerous examples of such systems, which can be further classified by two categories:

- (a) *The resource allocation algorithms are running over the underlying Markov chain.* Take the cross-layer optimization for wireless networks as an example. Wireless links are scheduled by the underlying Markov chain and average link service rates are observed between two updates of resource allocation algorithms. Here network dynamics are link service rates. Then the average link service rates from MAC layer are provided to upper-layer resource allocation algorithms, where source rates and link prices will be updated.

- (b) *The resource allocation algorithms are running under the overlying Markov chain.* Take the optimal streaming rates for peer-to-peer network with bounded neighbors as an example. When each peer chooses its neighbor set, the resulted peer topology is fixed, then the resource allocation algorithms compute the corresponding streaming capacity. Then the overlying Markov chain updates network peer topology and tries to find optimal peer topology under which the streaming capacity is maximized. Here network dynamics are network peer topologies.

In between the updates of resource allocation algorithms, the network dynamics and the designed Markov chain continue to evolve randomly. This turns out to introduce substantial difficulty in the analytic work built on these systems. The standard way used to avoid this issue is to assume the separation of timescales, i.e., that the network dynamics are either much faster or much slower than the update frequency of resource allocation algorithm.

In the former case, the resource allocation algorithm is assumed to see an averaged network behavior, i.e., between consecutive updates of algorithm, the network dynamics have time to converge to some equilibrium, and the designed Markov chain converges to its stationary distribution. However, in practice, most resource allocation algorithms do not satisfy this assumption. There are mainly two reasons: (1) it may take a long time for the designed Markov chain to converge, especially in large networks (2) Many resource allocation algorithms need to keep track of variations of some fast-changing network parameters.

In the latter case, there are two possibilities. One is that the network dynamics is slowly varying. However, many important network scenarios do not satisfy this property, e.g, peer-to-peer networks with high rates of peer churn (peers join and leave the network with high frequency). The other is that the resource allocation algorithm updates very frequently. This will incur the cost of high communication complexity and high computation complexity.

Throughout this thesis, we do not assume timescale separation in either of the above two ways. Instead we take the natural and general way where the network dynamics and the designed Markov chain evolve continuously while the resource allocation algorithm updates on discrete periods. The resource allocation algorithm does not need to achieve optimality in each update, nor do the network dynamics or the designed Markov chain have to converge between two algorithm updates.

As shown in the following, under mild conditions, we provide some convergence results for scenarios that the resource allocation algorithms are running over the underlying Markov chain, and the *vice versa*.

2.3.1 Resource Allocation Algorithms Over Markov Chain

In this scenario, resource allocation algorithms updates without waiting for the convergence of the underlying Markov chain in each iteration. We provide some convergence results by combination of stochastic approximation method [41] and mixing time bound of Markov chain [17]. For convenience, we focus on some special cases, however, our technique can be applied for more general cases.

In the following, we assume that network design problem has been transformed into the following problem after applying Markov approximation framework and Lagrange decomposition:

$$\mathbf{RA} : \min_{\mathbf{r} \geq 0} \max_{\mathbf{x} \geq 0} L_{\beta}(\mathbf{x}, \mathbf{r}) \quad (2.15)$$

where $L_{\beta}(\mathbf{x}, \mathbf{r})$ is some Lagrange function, β represents the accuracy of approximation, $\mathbf{x} = [x_s, s \in S]$ is the vector of primal resource allocation variables, and $\mathbf{r} = [r_l, l \in L]$ is the vector of dual resource allocation variables. For example, \mathbf{x} can represent a vector of source rates and \mathbf{r} represent a vector of link prices. Without loss of generality, we assume that only \mathbf{r} depends on the ensemble average of network dynamics induced by Markov chain

The vector \mathbf{x} and the vector \mathbf{r} are updated at time t_m , $m = 1, 2, \dots$ with $t_0 = 0$. Define $T_m = t_{m+1} - t_m$, and the “period m (update interval m)” as time interval $[t_m, t_{m+1})$, $m = 0, 1, \dots$. $\mathbf{x}(t), \mathbf{r}(t)$ remain the same in period m . Let $\mathbf{x}(m), \mathbf{r}(m)$ be the value of $\mathbf{x}(t), \mathbf{r}(t)$ for all $t \in [t_m, t_{m+1})$. To begin with, we assume $\mathbf{x}(0) = \mathbf{0}$ and $\mathbf{r}(0) = \mathbf{0}$ for simplicity.

Given $\mathbf{x}(m), \mathbf{r}(m)$ at the beginning of period m , denote the vector $\mathbf{f}(m) = [f_s(m), s \in S]$ as the subgradient vector of $L_\beta(\mathbf{x}(m), \mathbf{r}(m))$ with respect to $\mathbf{x}(m)$, and the vector $\mathbf{g}(m) = [g_l(m), l \in L]$ be the subgradient vector of $L_\beta(\mathbf{x}(m), \mathbf{r}(m))$ with respect to $\mathbf{r}(m)$. Further, we denote $\bar{\mathbf{g}}(m) = [\bar{g}_l(m), l \in L]$ as the estimation of $\mathbf{g}(m)$ by replacing the ensemble average of network dynamics induced by Markov chain with time average of network dynamics induced by Markov chain within the period m .

Now denote \mathcal{F}_m as the σ -field containing accumulated history information until time t_m , and denote $\mathbf{B}(m)$ as the biased estimation error of $\mathbf{g}(m)$, i.e.,

$$\mathbf{B}(m) \triangleq E[\bar{\mathbf{g}}(m)|\mathcal{F}_m] - \mathbf{g}(m) \quad (2.16)$$

The resource allocation algorithm tries to find the saddle points of problem **RA** (2.15). The corresponding stochastic primal-dual subgradient algorithm is shown as follows:

$$x_s(m+1) = [x_s(m) + \epsilon(m) \cdot f_s(m)]_+, \quad \forall s \in S \quad (2.17)$$

$$r_l(m+1) = [r_l(m) - \epsilon(m) \cdot \bar{g}_l(m)]_+, \quad \forall l \in L \quad (2.18)$$

where where $[\cdot]_+ \triangleq \max(\cdot, 0)$, and $\epsilon(m)$ is the update step size for period m .

Under suitable choices of step sizes and update intervals, we can establish the convergence of the stochastic primal-dual algorithm (2.17)-(2.18) with probability one in the following theorem:

Theorem 2.4. *Assume that $\max_{s,m} |f_s(m)| < \infty$, $\max_{l,m} |\bar{g}_l(m)| < \infty$, $\max_{s,m} x_s(m) < \infty$ and $\max_{l,m} r_l(m) < \infty$. The stochastic primal-dual algorithm in (2.17)-(2.18) converges to the optimal solutions of problem **RA** asymptotically with*

probability one under the following conditions on step sizes and update intervals:

$$\epsilon(m) > 0 \forall m, \sum_{m=0}^{\infty} \epsilon(m) = \infty, \sum_{m=0}^{\infty} \epsilon^2(m) < \infty \quad (2.19)$$

$$\sum_{m=0}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \quad (2.20)$$

$$\sum_{m=0}^{\infty} \epsilon(m) \cdot |[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| < \infty \quad (2.21)$$

The proof is relegated to Appendix 2.5.2. Condition (2.19) means the step size is decreasing with time, but it can not decrease too fast. Condition (2.20) means the update interval should increase with time. Condition (2.21) is a technique condition on the underlying Markov chain. Intuitively, these three conditions say that as the update interval approaches infinity, the time average is approaching the ensemble average, and the accumulated estimation errors are canceled by diminishing step sizes. Note that this theorem applies to any markov chains.

For the Markov chains we consider in this thesis, they are all irreducible, and finite. For this type of Markov chain, we have the following lemma [46]:

Lemma 2.5 (cf. [17, 46]). *Given an irreducible and finite Markov chain, its stationary distribution is denoted by π . For any initial distribution δ , the distribution at time t is denoted by $\pi_\delta(t)$. Then for any $t \geq 0$, there exist constants $\alpha > 0$ and $C > 0$ such that*

$$\max_{\delta} \|\pi_\delta(t) - \pi\|_{TV} \leq C \cdot \exp(-\alpha t) \quad (2.22)$$

where we use $\|\cdot\|_{TV}$ to denote the total variation distance between two distributions.

It turns out the condition in (2.21) is automatically satisfied if lemma 2.5 is true. Thus we obtain a new theorem with less number of conditions:

Theorem 2.6. *Assume that the underlying Markov chain is irreducible and finite, $\max_{s,m} |f_s(m)| < \infty$, $\max_{l,m} |\bar{g}_l(m)| < \infty$, $\max_{s,m} x_s(m) < \infty$ and*

$\max_{l,m} r_l(m) < \infty$. The stochastic primal-dual algorithm in (2.17)-(2.18) converges to the optimal solutions of problem **RA** asymptotically with probability one under the following conditions on step sizes and update intervals:

$$\epsilon(m) > 0 \forall m, \sum_{m=0}^{\infty} \epsilon(m) = \infty, \sum_{m=0}^{\infty} \epsilon^2(m) < \infty \quad (2.23)$$

$$\sum_{m=0}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \quad (2.24)$$

Further, the setting $\epsilon(0) = T(0) = 1, \epsilon(m) = \frac{1}{m}, T_m = m, m \geq 1$ is one specific choice of step sizes and update intervals satisfying conditions (2.23)-(2.24). This setting depends only on the time index m , and thus can be generally applied to any network.

The proof is relegated to Appendix 2.5.4. We have the following remarks:

- The results in theorem 2.4 and 2.6 are more general than existing results [30].
- It is not clear of the extension to ergodic Markov chain with infinite number of states.

In the following, we provide convergence results under the setting of constant step sizes and constant update intervals.

Theorem 2.7. Assume that $\max_{s,m} |f_s(m)| < \infty, \max_{l,m} |\bar{g}_l(m)| < \infty, \max_{s,m} x_s(m) < \infty$ and $\max_{l,m} r_l(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:

$$T_m = T_0 > 0 \quad \forall m \quad (2.25)$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \quad (2.26)$$

$$|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \leq \frac{C_3}{T_0}, \quad \forall m \quad (2.27)$$

where C_3 is a positive constant, then the stochastic primal-dual algorithm (2.17)-(2.18) converges with probability 1 to a bounded neighborhood of $\hat{\mathbf{x}}$ and

$\hat{\mathbf{r}}$, i.e., optimal solutions of **RA** (2.15) as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\},$$

where C_1, C_2 are upper bounds on $\max_m \|\mathbf{f}(m)\|^2$ and $\max_m \|\bar{\mathbf{g}}(m)\|^2$ respectively, $\mathbf{f}(m) = [f_s(m), s \in S]$ and $\bar{\mathbf{g}}(m) = [\bar{g}_l(m)], l \in L$.

Similarly, we have the following convergence result if the underlying Markov chain is irreducible and finite:

Theorem 2.8. *Assume that the underlying Markov chain is irreducible, $\max_{s,m} |f_s(m)| < \infty$, $\max_{l,m} |\bar{g}_l(m)| < \infty$, $\max_{s,m} x_s(m) < \infty$ and $\max_{l,m} r_l(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:*

$$T_m = T_0 > 0 \quad \forall m \tag{2.28}$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \tag{2.29}$$

then the stochastic primal-dual algorithm (2.17)-(2.18) converges with probability 1 to a bounded neighborhood of $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$, i.e., optimal solutions of **RA** (2.15) as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\},$$

where C_3 is a positive constant, C_1, C_2 are upper bounds on $\max_m \|\mathbf{f}(m)\|^2$ and $\max_m \|\bar{\mathbf{g}}(m)\|^2$ respectively, $\mathbf{f}(m) = [f_s(m), s \in S]$ and $\bar{\mathbf{g}}(m) = [\bar{g}_l(m)], l \in L$.

The proof is omitted since it is similar to the proof of Theorem 2.6.

Our emphasis on the constant step size and constant update interval has two reasons. First, a diminishing step size usually leads to slow convergence near the optimal solutions. Second, it is convenient to implement the constant step size and constant update interval in practice.

The proof technique here is inspired by the one used in Jiang-Walrand's paper [30]. The difference between our proof and [29, 30] lies in the following aspects: our proof studies the saddle points of Lagrangian function, while [29,30] studies the optimal dual solutions directly. Our results are more general without assuming that the Markov chain is time-reversible.

2.3.2 Markov Chain Over Resource Allocation Algorithms

In this scenario, given any configuration $f \in \mathcal{F}$, system performance $\phi(f)$ is obtained by running some resource allocation algorithms. For example, let configuration f represents some topology of peer-to-peer networks, streaming capacity $\phi(f)$ is obtained by running distributed broadcast algorithms.

Since we want to choose the best configuration in which system performance is maximized, by Markov approximation framework, we design a Markov chain with stationary distribution

$$p_f^* = \frac{\exp[\beta\phi(f)]}{\sum_{f' \in \mathcal{F}} \exp[\beta\phi(f')]}, \quad \forall f \in \mathcal{F}, \quad (2.30)$$

We know that transition rates of the designed Markov chain depend on the system performance $\phi(f)$, $f \in \mathcal{F}$. If we can obtain exact value of $\phi(f)$ for any configuration $f \in \mathcal{F}$, then we know that the designed Markov chain will converge to the desired stationary distribution (2.30) given any initial distribution [34].

In practice, however, we usually obtain inaccurate values of $\phi(f)$ for any configuration $f \in \mathcal{F}$, result in the inaccurate transition rates. These inaccuracies are caused by two factors. One is the noisy or imprecise measurements of the system performance. The other is the fast state transitions of Markov chain, i.e., for some configuration $f \in \mathcal{F}$, the Markov chain will jump to other configurations $f' \in \mathcal{F} - f$ before the underlying resource allocation algorithms

converge.

Therefore, the designed Markov chain with inaccurate transition rates may *not* converge to the desired stationary distribution $p_f^*, f \in \mathcal{F}$ (2.30). We need to study the convergence properties of the designed Markov chain with inaccurate transition rates.

We adopt the following setting:

- 1 Given topology of state space, $\forall f, f' \in \mathcal{F}$ and direct transition rates $q_{f,f'} \neq 0$ and $q_{f',f} \neq 0$, we set $q_{f,f'} = \alpha \frac{1}{\exp[\beta\phi(f)]}$ and $q_{f',f} = \alpha \frac{1}{\exp[\beta\phi(f')]}$, where $\alpha > 0$ is a constant.
- 2 For each configuration $f \in \mathcal{F}$ with exact system performance $\phi(f)$, we assume its corresponding inaccurate observed system performance belongs to the bounded region $[-\Delta_f, \Delta_f]$. Δ_f is the inaccuracy bound and can be different for different f . The observed system performance for configuration f only takes one of the following $2n_f + 1$ discrete values:

$$\left[\phi(f) - \Delta_f, \dots, \phi(f) - \frac{1}{n_f} \Delta_f, \phi(f), \phi(f) + \frac{1}{n_f} \Delta_f, \dots, \phi(f) + \Delta_f \right],$$

where n_f is a positive constant. Further, with probability η_{f_j} , the observed system performance takes value $\phi(f) + \frac{j}{n_f} \Delta_f, \forall j \in \{-n_f, \dots, n_f\}$ and $\sum_{j=-n_f}^{n_f} \eta_{f_j} = 1$.

With the inaccurate observed system performance, the original Markov chain behaves as follows. Suppose the current configuration is f and the observed system performance is $\phi(f) + \frac{j}{n_f} \Delta_f$, where $j \in \{-n_f, \dots, n_f\}$. After some count-down process, with a rate $\alpha \frac{1}{\exp[\beta(\phi(f) + \frac{j}{n_f} \Delta_f)]}$, the system transits to a new configuration f' . The measured system performance is $\phi(f') + \frac{k}{n_{f'}} \Delta_{f'}$, where $k \in \{-n_{f'}, \dots, n_{f'}\}$. This event happens with probability $\eta_{f'_k}$.

We construct a Markov chain to capture and study the above configuration hopping behavior. In this Markov chain, a state is associated with a configuration and an observed system performance. Given any configuration

$f \in \mathcal{F}$ and its corresponding $\phi(f)$, there are $2n_f + 1$ states in the extended Markov chain: $(f, \phi(f) + \frac{j}{n_f} \Delta_f)$, $j \in \{-n_f, \dots, n_f\}$. Further, Given direct transitions between configuration f and f' in the original configuration hopping Markov chain, there are direct transitions between states $(f, \phi(f) + \frac{j}{n_f} \Delta_f)$ and $(f', \phi(f') + \frac{k}{n_{f'}} \Delta_{f'})$ ($\forall j \in \{-n_f, \dots, n_f\}, k \in \{-n_{f'}, \dots, n_{f'}\}$) in the corresponding new Markov chain. The corresponding transition rates are shown as follows:

$$\begin{aligned} & q_{(f, \phi(f) + \frac{j}{n_f} \Delta_f), (f', \phi(f') + \frac{k}{n_{f'}} \Delta_{f'})} \\ &= \alpha \frac{\eta_{f'_k}}{\exp[\beta(\phi(f) + \frac{j}{n_f} \Delta_f)]} \end{aligned} \quad (2.31)$$

and

$$\begin{aligned} & q_{(f', \phi(f') + \frac{k}{n_{f'}} \Delta_{f'}), (f, \phi(f) + \frac{j}{n_f} \Delta_f)} \\ &= \alpha \frac{\eta_{f_j}}{\exp[\beta(\phi(f') + \frac{k}{n_{f'}} \Delta_{f'})]}, \end{aligned} \quad (2.32)$$

where $\sum_{j=-n_f}^{n_f} \eta_{f_j} = 1$ and $\sum_{k=-n_{f'}}^{n_{f'}} \eta_{f'_k} = 1$. This new Markov chain can be thought as an extended version of the original configuration hopping Markov chain. As an example, an extended Markov chain is shown and explained in Fig. 2.2.

The extended Markov chain has a unique stationary distribution since it is irreducible and only has a finite number of states. We can study the impact of inaccurate broadcast rates by comparing the stationary configuration distribution of the new Markov chain and that of the original configuration hopping Markov chain.

We denote the stationary distribution of the *states* in the new Markov chain by

$$\tilde{\mathbf{p}} \triangleq [\tilde{p}_{f, \phi(f) + \frac{j}{n_f} \Delta_f}, j \in \{-n_f, \dots, n_f\}, f \in \mathcal{F}] \quad (2.33)$$

We also denote $\bar{\mathbf{p}} : [\bar{p}_f(\mathbf{x}), f \in \mathcal{F}]$ as the stationary distribution of the *configurations* in the extended Markov chain. Given a configuration $f \in \mathcal{F}$, there

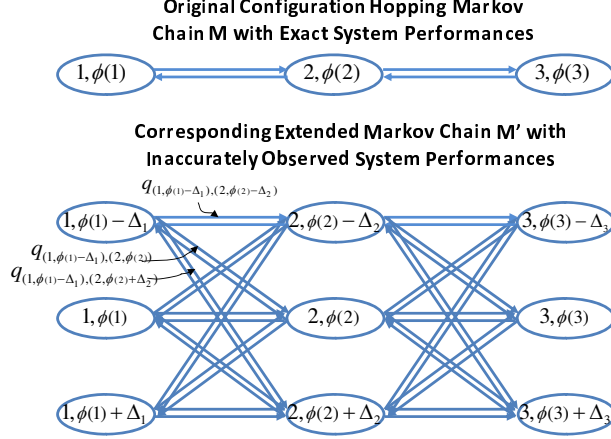


Figure 2.2: An example of the original three-state topology hopping Markov chain and the extended Markov chain. M is the original configuration hopping Markov chain with accurate system performance. M' is the corresponding extended Markov chain with inaccurate system observations. For each configuration $f \in \{1, 2, 3\}$, the observed system performance takes values $\phi(f) - \Delta_f$, $\phi(f)$, $\phi(f) + \Delta_f$ with probability η_{f-1} , η_{f_0} and η_{f_1} respectively. The transition rates are assigned according to (2.31) and (2.32).

are $2n_f + 1$ states associated with f in the extended Markov chain. We have

$$\bar{p}_f(\mathbf{x}) = \sum_{j \in \{-n_f, \dots, n_f\}} \tilde{p}_{f, \phi(f) + \frac{j}{n_f} \Delta_f}, \forall f \in \mathcal{F} \quad (2.34)$$

Recall that the stationary distribution of the configurations for the original configuration hopping Markov chain is $\mathbf{p}^* : [p_f^*(\mathbf{x}), f \in \mathcal{F}]$. We use the total variance distance [17] to quantify the difference between \mathbf{p}^* and $\bar{\mathbf{p}}$, as

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \triangleq \frac{1}{2} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \quad (2.35)$$

We have the following result:

Theorem 2.9. *Let $\Delta_{\max} = \max_{f \in \mathcal{F}} \Delta_f$, and $x_{\max} = \max_{f \in \mathcal{F}} x_f$. The $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ are bounded as follows:*

$$0 \leq d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \leq 1 - \exp(-2\beta\Delta_{\max}). \quad (2.36)$$

Further, the optimality gap in broadcast rates $|\mathbf{p}^* \mathbf{x}^T - \bar{\mathbf{p}} \mathbf{x}^T|$ is bounded as below:

$$|\mathbf{p}^* \mathbf{x}^T - \bar{\mathbf{p}} \mathbf{x}^T| \leq 2x_{\max}(1 - \exp(-2\beta\Delta_{\max})). \quad (2.37)$$

The proof is relegated to Appendix 2.5.6.

The general upper bound of $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ (2.36) can be further tightened if we know more about the error distribution. We have the following result:

Theorem 2.10. *The upper bound of $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ can be tightened as follows:*

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \leq 1 - \exp(-\beta \Delta_{\max}). \quad (2.38)$$

if the error follows a uniformly distribution, which means for any configuration $f \in \mathcal{F}$, the corresponding error can be any one of $2n_f + 1$ values $[-\Delta_f, \dots, -\frac{1}{n_f}\Delta_f, 0, \frac{1}{n_f}\Delta_f, \dots, \Delta_f]$ with equal probability $\frac{1}{2n_f+1}$. In addition, no other error distribution can further tighten this bound (2.38). In this sense, the uniformly distribution is the optimal error distribution.

The proof is relegated to Appendix 2.5.7.

Remark:

- It is not hard to verify that this theorem also holds for other options of transition rates, e.g, $q_{f,f'} = \alpha \exp[\beta\phi(f')], \forall f, f' \in \mathcal{F}, q_{f,f'} \neq 0$, or $q_{f,f'} = \alpha \cdot \frac{\exp(\beta\phi_{f'})}{\exp(\beta\phi_f) + \exp(\beta\phi_{f'})}, \forall f, f' \in \mathcal{F}, q_{f,f'} \neq 0$.
- The upper bound on deviation from the optimal Markov chain distribution $d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}})$ (2.36) decreases exponentially with the maximum error decreasing in the system performances.
- When the maximum error $\Delta_{\max} = 0$, i.e., there are no errors with streaming rates, the deviation is zero and we obtain the desired stationary distribution.
- The upper bound on deviation is independent of the number of configurations $|\mathcal{F}|$, and the number of quantized error levels $2n_f + 1, \forall f \in \mathcal{F}$.

Application of Theorem 2.9

Note that scenario in this subsection is Markov chain over resource allocation algorithms. Since we focus on the convergence issues, we assume that the distributed solutions has been given. In the following, we list several examples of combinatorial optimization problems, to which the results of Theorem 2.9 can be applied once we have the distributed solutions.

- **Example 1: optimal path selection problem for wireline networks.**

Considering a wireline network with a set of users S and a set of path configurations \mathcal{F} . Each user $s \in S$ picks D_s paths from J_s , its set of available paths. A path configuration $f \in \mathcal{F}$ consists of individual users s using D_s paths. When the network system operates under f , the system-wide performance is $\phi(f)$. We need to select the best configuration where the system-wide performance is maximized, i.e., the optimal path selection.

- **Example 2: optimal multicast(broadcast) tree problem for wireline networks.**

Considering a wireline network with a set of users S and a set of tree configurations \mathcal{F} . Each user $s \in S$ picks D_s multicast(broadcast) trees from J_s , its set of available multicast(broadcast) trees to connect R_s , the receiver set of s . A tree configuration $f \in \mathcal{F}$ consists of individual users s using D_s trees. When the network system operates under f , the system-wide performance is $\phi(f)$. We need to select the best configuration where the system-wide performance is maximized, i.e., the optimal multicast(broadcast) tree selection.

- **Example 3: optimal neighboring selection problem for peer-to-peer networks.**

Considering a peer-to-peer network with a set of peers S and a set of neighboring configurations \mathcal{F} . Each peer $s \in S$ picks D_s neighbors from J_s , its set of available neighbors. A neighboring configuration $f \in \mathcal{F}$

consists of individual peers s connecting to D_s neighbors. When the network system operates under f , the system-wide performance is $\phi(f)$. We need to select the best configuration where the system-wide performance is maximized, i.e., the optimal peering topology selection.

- **Example 4: optimal channel assignment problem for wireless local area networks.**

Considering a wireless network with a set of access points (APs) S and a set of channel configurations \mathcal{F} . Each AP $s \in S$ chooses one channel from J_s , its set of available channels. A channel configuration $f \in \mathcal{F}$ consists of individual APs s using one channel. When the network system operates under f , the system-wide performance is $\phi(f)$. We need to select the best configuration where the system-wide performance is maximized, i.e., the optimal channel assignment.

- **Example 5: optimal power assignment problem for wireless local area networks.**

Considering a wireless network with a set of access points (APs) S and a set of power configurations \mathcal{F} . Each AP $s \in S$ chooses one power level from J_s , its set of available power levels. A power configuration $f \in \mathcal{F}$ consists of individual APs s using one power level. When the network system operates under f , the system-wide performance is $\phi(f)$. We need to select the best configuration where the system-wide performance is maximized, i.e., the optimal power assignment.

Actually, under some mild conditions, some special cases of example 1-5 have been solved [10]. However, in general, how to design distributed solutions for these examples without messaging passing is an open problem and deserves future investigation.

2.4 Connection to Statistical Physics

Simulated annealing [42] is a technique to find global optimal solution by utilizing physical annealing principle from statistical physics. Simulated Annealing [42] also uses Markov chain for algorithm design. The difference between Simulated Annealing and our work is that Simulated Annealing in general focuses on solving the problem exactly using centralized algorithms, while we focus on designing distributed algorithm to solve the optimization problem approximately.

Glauber dynamics [21] are single-vertex update mechanism based sampling techniques for any given input graph, and they are also from statistical physics. Glauber dynamics construct reversible Markov chains that have the desired distribution as stationary distribution and where at each step the status of one vertex of the graph is updated. This is a special case of our design framework and corresponds to option 4 (2.14) with assumption that $\phi(f) = \sum_{s \in \mathcal{S}} \phi_s(f), \forall f \in \mathcal{F}$.

Note that when $\phi(f)$ can not be decomposed into separate sub-comments, Glauber dynamics can not be applied. For example, in example 3 in the above session 2.3.2, f represents one feasible peering topology, $\phi(f)$ is the streaming capacity given f , then for any two feasible topology f and f' , the difference between $\phi(f)$ and $\phi(f')$ can not depend only on single peer, i.e, single-vertex update (Glauber dynamics) is not possible. Similarly, Glauber dynamics fails for all other examples in the above session 2.3.2, including example 1, example 2, example 4 and example 5.

Furthermore, to be best of our knowledge, existing work on Glauber dynamics does not consider either the interaction between Glauber dynamics and trajectories of some controlled dynamics, or the impacts of inexact transition rates on the stationary distribution.

In contrast, we relate the product-form distribution to the solution of general combinatorial network optimization problems, discuss various ways to obtain the distribution in a distributed manner, and explore the impact of noisy(in-accurate) measurement on achieving the distribution and thus the close-to-optimal value.

2.5 Appendix of Chapter 2

2.5.1 Proof of Lemma 2.3

First, we will construct a continuous-time time-reversible ergodic Markov chain and show that its stationary distribution is $p_f^*(\phi), f \in \mathcal{F}$ in (2.7). In particular, we construct a continuous-time Markov chain Y with a finite state space \mathcal{F} . We design the Markov chain Y such that any two states f and f' can communicate directly with each other, i.e., the transition rate from f to f' is $q_{f,f'} \neq 0$ for any $f, f' \in \mathcal{F}$. Furthermore, for any $f, f' \in \mathcal{F}$, we set

$$q_{f,f'} = \alpha [\exp(\beta\phi(f))]^{-1}. \quad (2.39)$$

Thus Y is an ergodic Markov chain with unique stationary distribution. By (2.39) and (2.7), we can check that detailed balance equations hold, by Theorem 1.3 and Theorem 1.14 in [34], we know that Y is reversible, and its stationary distribution is indeed $p_f^*(\phi), f \in \mathcal{F}$ in (2.7).

Next, we will establish that for any continuous-time time-reversible ergodic Markov chain X , its stationary distribution π can be expressed by the product form $p_f^*(\phi), f \in \mathcal{F}$ in (2.7). For the state-transition diagram for the Markov chain X , we map it to an undirected graph $G=(V,E)$, where the node set $V = \mathcal{F}$ is the set of states and any edge $e(i,j) \in E, i,j \in V$ represents the state-pair (i,j) with $q_{i,j} \neq 0$.

Let the stationary distribution of state j be denoted by π_j , and transition rate from state j to state j' is denoted by $q_{j,j'}$, then by detailed balance equation

of time-reversible Markov chain, we know that $\pi_j q_{j,j'} = \pi_{j'} q_{j',j}$. Let $\rho_{j,j'} = q_{j,j'}/q_{j',j}$ for any $q_{j',j} \neq 0$, then $\pi_{j'} = \pi_j \rho_{j,j'}$.

Since X is an ergodic Markov chain, any two states can reach each other within finite transitions, and G is a connected graph. We can always find a spanning tree to connect all nodes in G and there exists only one path between any pair of nodes. Suppose we have constructed a spanning tree on G . Then we denote the root state as state 0, and denote nodes in V as state $1, 2, \dots, |V|-1$, according to the result of the breadth-first search on the spanning tree. Let $\text{PATH}(0, i)$ be the path between state 0 and the state i ($1 \leq i \leq |V|-1$), passing m_i+1 number of states (including states 0 and i). We order the nodes on the path $\text{PATH}(0,i)$ according to their distances to state 0, and denoted them as $v_{0,i}^j$ ($0 \leq j \leq m_i$). Then according to detailed balanced equations along the $\text{PATH}(0, i)$, we have the following:

$$\pi_i = \pi_0 \cdot \prod_{j=0}^{m_i-1} \rho_{v_{0,i}^j, v_{0,i}^{j+1}}, \quad 1 \leq i \leq |V|-1 \quad (2.40)$$

$$\pi_0 + \sum_{i=1}^{|V|-1} \pi_i = 1 \quad (2.41)$$

Then we get the distribution

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{|V|-1} \prod_{j=0}^{m_k-1} \rho_{v_{0,k}^j, v_{0,k}^{j+1}}} \quad (2.42)$$

$$\pi_i = \frac{\prod_{j=0}^{m_i-1} \rho_{v_{0,i}^j, v_{0,i}^{j+1}}}{1 + \sum_{k=1}^{|V|-1} \prod_{j=0}^{m_k-1} \rho_{v_{0,k}^j, v_{0,k}^{j+1}}}, \quad 1 \leq i \leq |V|-1 \quad (2.43)$$

We now verify the distribution computed based on the spanning tree, i.e., (2.42)-(2.43), is the correct stationary distribution, by testing the detailed balance equations between any two states $j, j' \in V$.

1. If $q_{j,j'} = 0$, then the detailed balance equation trivially holds.

2. If $q_{j,j'} \neq 0$ and the edge $e(j, j')$ belongs to the spanning tree, then by (2.43), we know that $\pi_{j'} = \pi_j \rho_{j,j'}$, i.e., the detailed balance equation holds.
3. If $q_{j,j'} \neq 0$ and the edge $e(j, j')$ does not belong to the spanning tree, then we focus on the cycle consisting of $PATH(j', j)$ and $e(j, j')$. Starting from node $j' = v_{j',j}^0$, we can visit nodes $v_{j',j}^k, 1 \leq k \leq m_{j',j} - 1$ and node $j = v_{j',j}^{m_{j',j}}$ in sequence along the $PATH(j', j)$. By Kolmogorov's criteria for time-reversible Markov chain [34], we have

$$\rho_{j',j} = \prod_{k=0}^{m_{j',j}-1} \rho_{v_{j',j}^k, v_{j',j}^{k+1}} \quad (2.44)$$

By (2.43) and (2.44) we have

$$\frac{\pi_j}{\pi_{j'}} = \prod_{k=0}^{m_{j',j}-1} \rho_{v_{j',j}^k, v_{j',j}^{k+1}} = \rho_{j',j} \quad (2.45)$$

Therefore, the detailed balance equation between j and j' holds.

Combining above scenarios, we know that the detailed balance equations between any two states $j, j' \in V$ hold. So the distribution shown in (2.42)-(2.43) is indeed the stationary distribution.

Further, the stationary distribution π shown in (2.42)-(2.43) can be expressed in the product form in (2.7) as follows:

$$\pi_0 = \frac{\exp(0)}{\exp(0) + \sum_{k=1}^{|V|-1} \exp\left(\sum_{j=0}^{m_k-1} \log \rho_{v_{0,k}^j, v_{0,k}^{j+1}}\right)} \quad (2.46)$$

$$\pi_i = \frac{\exp\left(\sum_{j=0}^{m_i-1} \log \rho_{v_{0,i}^j, v_{0,i}^{j+1}}\right)}{\exp(0) + \sum_{k=1}^{|V|-1} \exp\left(\sum_{j=0}^{m_k-1} \log \rho_{v_{0,k}^j, v_{0,k}^{j+1}}\right)}, 1 \leq i \leq |V| - 1. \quad (2.47)$$

So we get the desired conclusion.

2.5.2 Proof of Theorem 2.4

Let $\mathbf{y}^0(m)$ be the state of the Markov chain at the beginning of period m . Define the random vector $U(m) = (\bar{\boldsymbol{\theta}}(m-1), \mathbf{x}(m), \mathbf{r}(m), \mathbf{y}^0(m))$ for $m \geq 1$ and $U(0) = (\mathbf{x}(0), \mathbf{r}(0), \mathbf{y}^0(0))$. For $m \geq 1$, let \mathcal{F}_m be the σ -field generated by $U(0), U(1), \dots, U(m)$, denoted by

$$\mathcal{F}_m = \sigma(U(0), U(1), \dots, U(m)). \quad (2.48)$$

Then $\forall l \in L$, $\bar{g}_l(m)$ can be decomposed into three parts: $\bar{g}_l(m) = g_l(m) + (E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m)) + (\bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m])$.

The first part is the exact subgradient $g_l(m)$.

The second part is the biased estimation error of $g_l(m)$, denoted by

$$B_l(m) \triangleq E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m) \quad (2.49)$$

The third part is a zero-mean martingale difference noise, denoted by

$$\eta_l(m) \triangleq \bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m] \quad (2.50)$$

Therefore,

$$\bar{g}_l(m) = g_l(m) + B_l(m) + \eta_l(m), \forall l \in L \quad (2.51)$$

Recall that $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ is the optimal solution to the problem **RA** (2.15). Thus $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ is a saddle point for $L_\beta(\mathbf{x}, \mathbf{r})$, it follows that

$$L_\beta(\mathbf{x}, \hat{\mathbf{r}}) \leq L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}}) \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}) \quad (2.52)$$

By using $\|\cdot\|$ to denote the *Euclidean* norm, we define the function $V(\cdot, \cdot)$ as follows:

$$V(\mathbf{x}, \mathbf{r}) \triangleq \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \|\mathbf{r} - \hat{\mathbf{r}}\|^2 \quad (2.53)$$

For any given $\mu > 0$, We also define the set

$$H_\mu \triangleq \{(\mathbf{x}, \mathbf{r}) : L_\beta(\hat{\mathbf{x}}, \mathbf{r}) - L_\beta(\mathbf{x}, \hat{\mathbf{r}}) \leq \mu\} \quad (2.54)$$

In the following, we need two steps to establish the convergence result.

- **Step 1:** we will show that $\forall \mu > 0$, H_μ is recurrent for $\{\mathbf{x}(m), \mathbf{r}(m)\}$.
- **Step 2:** we will show that for a sufficient large number m , and any $n \geq m + 1$, $\{\mathbf{x}(n), \mathbf{r}(n)\}$ will reside in H_μ almost surely.

Before the further illustrate of **Step 1** and **Step 2**, we need the following two lemmas. Lemma 2.11 is the condition (2.21) and is satisfied. Proofs of Lemma 2.12 is provided at Appendix 2.5.3.

Lemma 2.11. $\sum_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| < \infty$

Lemma 2.12. Let $W(n) \triangleq \sum_{i=1}^{n-1} \{\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T \boldsymbol{\eta}(i)\}$, then $W(n)$ converges with probability 1.

Step 1: Since

$$\begin{aligned} x_s(m+1) &= [x_s(m) + \epsilon(m) \cdot f_s(m)]_+, \quad \forall s \in S \\ r_l(m+1) &= [r_l(m) - \epsilon(m) \cdot \bar{g}_l(m)]_+, \quad \forall l \in L \end{aligned}$$

by using the fact that the projection $[\cdot]_+$ is non-expansive [5], we have

$$\begin{aligned} \|\mathbf{x}(m+1) - \hat{\mathbf{x}}\|^2 &\leq \|\mathbf{x}(m) + \epsilon(m) \cdot \mathbf{f}(m) - \hat{\mathbf{x}}\|^2 \\ &= \|\mathbf{x}(m) - \hat{\mathbf{x}}\|^2 + 2\epsilon(m) \cdot [\mathbf{x}(m) - \hat{\mathbf{x}}]^T \mathbf{f}(m) \\ &\quad + \epsilon^2(m) \|\mathbf{f}(m)\|^2. \end{aligned}$$

Similarly, with (2.78), we have

$$\begin{aligned} \|\mathbf{r}(m+1) - \hat{\mathbf{r}}\|^2 &\leq \|\mathbf{r}(m) - \epsilon(m) \cdot \bar{\mathbf{g}}(m) - \hat{\mathbf{r}}\|^2 \\ &= \|\mathbf{r}(m) - \hat{\mathbf{r}}\|^2 - 2\epsilon(m) \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T \bar{\mathbf{g}}(m) \\ &\quad + \epsilon^2(m) \|\bar{\mathbf{g}}(m)\|^2 \\ &= \|\mathbf{r}(m) - \hat{\mathbf{r}}\|^2 - 2\epsilon(m) \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T [\mathbf{g}(m) \\ &\quad + \mathbf{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2(m) \|\bar{\mathbf{g}}(m)\|^2 \end{aligned}$$

By assumption, we know that both $\|\mathbf{f}(m)\|^2$ and $\|\bar{\mathbf{g}}(m)\|^2$ are bounded, we can write that $\|\mathbf{f}(m)\|^2 \leq C_1$ and $\|\bar{\mathbf{g}}(m)\|^2 \leq C_2$, where C_1 and C_2 are positive constants. Using this and the above inequalities, we have that

$$\begin{aligned}
& V(\mathbf{x}(m+1), \mathbf{r}(m+1)) \\
&= \|\mathbf{x}(m+1) - \hat{\mathbf{x}}\|^2 + \|\mathbf{r}(m+1) - \hat{\mathbf{r}}\|^2 \\
&\leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2\epsilon(m) \cdot [(\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) \\
&\quad - (\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m)] - 2\epsilon(m) \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T [\mathbf{B}(m) + \boldsymbol{\eta}(m)] \\
&\quad + \epsilon^2(m) \cdot (C_1 + C_2)
\end{aligned} \tag{2.55}$$

Assuming that $(\mathbf{x}(m), \mathbf{r}(m)) \notin H_\mu$ (recall the definition of H_μ in (2.54)). Then we have

$$L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) - L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) \geq \mu \tag{2.56}$$

Since $L_\beta(\mathbf{x}, \mathbf{r})$ is concave in \mathbf{x} and convex in \mathbf{r} , $\mathbf{f}(m)$ and $\mathbf{g}(m)$ are the subgradient vectors of $L_\beta(\mathbf{x}, \mathbf{r})$ with respect to \mathbf{x} and \mathbf{r} respectively, it follows that

$$L_\beta(\mathbf{x}(m), \mathbf{r}(m)) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) \geq (\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) \tag{2.57}$$

$$L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\mathbf{x}(m), \mathbf{r}(m)) \geq -(\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m) \tag{2.58}$$

By the summation of (2.57) and (2.58), and combining (2.56), we have

$$\begin{aligned}
& (\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) - (\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m) \\
&\leq L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) \\
&\leq -\mu
\end{aligned} \tag{2.59}$$

Combining with (2.55) yields that

$$\begin{aligned}
& V(\mathbf{x}(m+1), \mathbf{r}(m+1)) \\
&\leq V(\mathbf{x}(m), \mathbf{r}(m)) - 2\epsilon(m)\mu \\
&\quad + 2\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T [\mathbf{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2(m) \cdot (C_1 + C_2)
\end{aligned} \tag{2.60}$$

Further,

$$\begin{aligned}
& E[V(\mathbf{x}(m+1), \mathbf{r}(m+1)) | \mathcal{F}_m] \\
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) - 2\epsilon(m)\mu \\
& \quad + 2\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T [\mathbf{B}(m)] + \epsilon^2(m) \cdot (C_1 + C_2)
\end{aligned} \tag{2.61}$$

By lemma 2.11 and condition (2.20), $|\sum_m \{\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)\}| < \infty$ and $\sum_m \epsilon^2(m) \cdot (C_1 + C_2) < \infty$. Then by supermartingale convergence lemma [4], we can conclude that the set H_μ is recurrent for $\{\mathbf{x}(m), \mathbf{r}(m)\}$.

Step 2: By (2.55) we have that for $n \geq m + 1$,

$$\begin{aligned}
& V(\mathbf{x}(n), \mathbf{r}(n)) \\
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2 \sum_{i=m}^{n-1} \{\epsilon(i) \cdot [(\mathbf{x}(i) - \hat{\mathbf{x}})^T \mathbf{f}(i) \\
& \quad - (\mathbf{r}(i) - \hat{\mathbf{r}})^T \mathbf{g}(i)]\} + 2 \sum_{i=m}^{n-1} \{\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T [\mathbf{B}(i) + \boldsymbol{\eta}(i)]\} \\
& \quad + (C_1 + C_2) \sum_{i=m}^{n-1} \epsilon^2(i)
\end{aligned} \tag{2.62}$$

Since $(C_1 + C_2) \sum_{i=1}^{\infty} \epsilon^2(i) < \infty$, $\sum_{i=1}^{\infty} |\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T \mathbf{B}(i)| < \infty$ by lemma 2.11, and $\sum_{i=1}^{\infty} |\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T \boldsymbol{\eta}(i)| < \infty$ by lemma 2.12, then

$$\lim_{m \rightarrow \infty} (C_1 + C_2) \sum_{i=m}^{\infty} \epsilon^2(i) = 0 \tag{2.63}$$

$$\lim_{m \rightarrow \infty} \sum_{i=m}^{\infty} |\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T \mathbf{B}(i)| = 0 \tag{2.64}$$

$$\lim_{m \rightarrow \infty} \sum_{i=m}^{\infty} |\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T \boldsymbol{\eta}(i)| = 0 \tag{2.65}$$

Combining (2.63), (2.64), and (2.65), we know that with probability 1, for any $\zeta > 0$, after $(\mathbf{x}(m), \mathbf{r}(m))$ returns to H_μ for some sufficiently large m (due to recurrence of H_μ),

$$\begin{aligned}
& 2 \sum_{i=m}^{n-1} \{\epsilon(i) \cdot [\hat{\mathbf{r}} - \mathbf{r}(i)]^T [\mathbf{B}(i) + \boldsymbol{\eta}(i)]\} \\
& \quad + (C_1 + C_2) \sum_{i=m}^{n-1} \epsilon^2(i) \leq \zeta
\end{aligned} \tag{2.66}$$

for any $n \geq m + 1$.

Combining (2.52) and (2.59), we have that

$$[(\mathbf{x}(i) - \hat{\mathbf{x}})^T \mathbf{f}(i) - (\mathbf{r}(i) - \hat{\mathbf{r}})^T \mathbf{g}(i)] \leq 0 \quad (2.67)$$

Therefore, applying (2.66) and (2.67) to (2.62), we have

$$V(\mathbf{x}(n), \mathbf{r}(n)) \leq V(\mathbf{x}(m), \mathbf{r}(m)) + \zeta, \forall n \geq m + 1.$$

Thus $(\mathbf{x}(n), \mathbf{r}(n))$ can not move far away from H_μ . Since this holds for H_μ with arbitrarily small $\mu > 0$ and any $\zeta > 0$, it follows that (\mathbf{x}, \mathbf{r}) converges to the optimal solution $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ with probability 1. This concludes the proof.

2.5.3 Proof of Lemma 2.12

First, we show that $W(n)$ is a martingale. By (2.48) and (2.50), we know that $\boldsymbol{\eta}(n-1) \in \mathcal{F}_n$, $E[\boldsymbol{\eta}(n-1)|\mathcal{F}_{n-1}] = \mathbf{0}$. Further, $\forall l \in L$, $|\eta_l(n)|$ is bounded and $|\eta_l(n)| < c_3$ for some $c_3 > 0$. Thus $W(n) \in \mathcal{F}_n$, $E|W(n)| < \infty, \forall n$ and $E(W(n)|\mathcal{F}_{n-1}) - W(n-1) = \epsilon(n-1) \cdot [\hat{\mathbf{r}} - \mathbf{r}(n-1)]^T E[\boldsymbol{\eta}(n-1)|\mathcal{F}_{n-1}] = 0$.

Then we prove that $\sup_n E(W(n)^2) < \infty$.

Since $\forall l \in L$, \hat{r}_l is bounded and $\hat{r}_l < \bar{r}$ for some $\bar{r} > 0$, then we have

$$|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \boldsymbol{\eta}(m)| \leq |L| \cdot c_3 [\bar{r} + r_{\max}]$$

Thus

$$\begin{aligned} & \sup_n E(W^2(n)) \\ &= \sup \sum_{m=1}^{n-1} E\{[\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T \boldsymbol{\eta}(m)]^2\} \\ &\leq \sum_{m=1}^{\infty} E\{[\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T \boldsymbol{\eta}(m)]^2\} \\ &\leq \sum_{m=1}^{\infty} \{\epsilon(m)^2 |L|^2 c_3^2 [\bar{r} + r_{\max}]^2\} \\ &= |L|^2 c_3^2 [\bar{r} + r_{\max}]^2 \sum_{m=1}^{\infty} \{\epsilon^2(m)\} \\ &< \infty \end{aligned}$$

where the last step follows from condition (2.20). By Martingale Convergence Theorem [41], $W(n)$ converges with probability 1.

2.5.4 Proof of Theorem 2.6

The key step is to show that condition 2.21 is satisfied if the lemma 2.5 is true.

In the following, we consider the period m , i.e., from t_m to t_{m+1} . At time t_m with $\mathbf{x}(m)$ and $\mathbf{r}(m)$, we denote the corresponding Markov chain as $Y(t)$, where $Y(t)$ is an irreducible continuous-time Markov chain.

To utilize the existing bounds on convergence to the stationary distribution of discrete-time Markov chain, we uniformize the continuous-time Markov chain $Y(t)$. Uniformization [38] plays the role of bridge between discrete-time Markov chain and continuous-time Markov chain.

Let the transition rate matrix of $Y(t)$ is denoted by $Q=\{Q(\mathbf{y}, \mathbf{y}')\}$, where $\mathbf{y}, \mathbf{y}' \in Y(t)$ are states. Construct a discrete-time Markov chain $Z(n)$ with its probability transition matrix $P= I + Q/v$, where I is the identity matrix. Then consider a system that successive states visited form a Markov chain $Z(n)$ and the times at which the system changes its state form a Poisson process $N(t)$. Here $N(t)$ is an independent Poisson process with rate v . Then the state of this system at time t is denoted by $Z(N(t))$, which is called a *subordinated Markov chain*.

Since the transition rates of $Y(t)$ are bounded, we have $\forall \mathbf{y}, \mathbf{y}', Q(\mathbf{y}, \mathbf{y}') \leq K_1$, where K_1 is a positive constant. Suppose \mathbf{y} can at most transit to M_1 other states, then Let $v = M_1 K_1$, we have $\sum_{\mathbf{y} \neq \mathbf{y}'} Q(\mathbf{y}, \mathbf{y}') \leq M_1 K_1 = v$. Therefore, by uniformization theorem [38], $Y(t)$ and $Z(N(t))$ has the same distribution, denoted by $Y(t) \stackrel{d}{=} Z(N(t))$.

Now let the vector $\boldsymbol{\omega}_m(t) = \{\omega_m(t, \mathbf{y})\}$ be the probabilities of all states at time $t_m + t$ ($0 \leq t \leq T_m$), given that the initial state at time t_m is $\mathbf{y}^0(m)$. Let

$\mathbf{y}(t_m + t)$ be the state at time $t_m + t$, then

$$\begin{aligned} & E[\bar{g}_l(m)|\mathcal{F}_m] \\ &= E\left[\int_0^{T_m} \sum_{g'} g' 1_{g_l(t_m+t)=g'} dt / T_m\right] \end{aligned} \quad (2.68)$$

$$= \int_0^{T_m} \sum_{g'} g' P(g_l(t_m + t) = g') dt / T_m \quad (2.69)$$

$$\begin{aligned} &= \int_0^{T_m} E[g_l(t_m + t)|\mathcal{F}_m] dt / T_m \\ &= \int_0^{T_m} \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot \omega_m(t, \mathbf{y}') dt / T_m \\ &= \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot \int_0^{T_m} \omega_m(t, \mathbf{y}') dt / T_m \\ &= \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot \bar{\omega}_m(\mathbf{y}') \end{aligned} \quad (2.70)$$

Where $g_l^{\mathbf{y}'}$ is the value of $g_l(m)$ at state \mathbf{y}' , $\bar{\omega}_m(\mathbf{y}') = \int_0^{T_m} \omega_m(t, \mathbf{y}') dt / T_m$ is the time-averaged probability of state \mathbf{y}' in the interval.

Since the initial distribution is concentrated at a single definite starting state $\mathbf{y}^0(m)$, we denote this distribution by $\delta_{\mathbf{y}^0}$. We let $\pi_{\mathbf{y}^0}$ be the probability of $\mathbf{y}^0(m)$ in the stationary distribution of $Y(t)$. Let $\boldsymbol{\pi}(m) \triangleq \{\pi_{\mathbf{y}}(m)\}$ be the stationary distribution of $Y(t)$, then by uniformization theorem [38], $\boldsymbol{\pi}(m)$ is also the stationary distribution of $Z(n)$.

We use $\|\cdot\|_{TV}$ to denote the total variation distance between two distributions [17, 46], which satisfies triangle inequality. Since discrete-time Markov chain $Z(n)$ with transition matrix P is irreducible and aperiodic [38], then for any $n \geq 0$, there exist constants $\rho \in (0, 1)$ and $K_3 > 0$ such that [46]:

$$\|\delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(m)\|_{TV} \leq K_3 \cdot \rho^n$$

Therefore,

$$\begin{aligned}
& \| \boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(m) \|_{TV} \\
&= \left\| \sum_{n=0}^{\infty} \frac{(vt)^n}{n!} \exp(-vt) \delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(m) \right\|_{TV} \\
&\leq \sum_{n=0}^{\infty} \frac{(vt)^n}{n!} \exp(-vt) \| \delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(m) \|_{TV} \\
&\leq K_3 \cdot \sum_{n=0}^{\infty} \frac{(vt\rho)^n}{n!} \exp(-vt) \\
&= K_3 \cdot \exp(-v(1-\rho)t)
\end{aligned}$$

Further,

$$\| \bar{\omega}_m - \boldsymbol{\pi}(m) \|_{TV} \tag{2.71}$$

$$\begin{aligned}
&= \left\| \int_0^{T_m} [\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(m)] dt / T_m \right\|_{TV} \\
&\leq \int_0^{T_m} \| \boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(m) \|_{TV} dt / T_m \\
&\leq \frac{K_3}{v(1-\rho)T_m}
\end{aligned} \tag{2.72}$$

Therefore, by (2.49) and (2.70), we have

$$\begin{aligned}
|B_l(m)| &= |E[\bar{g}_l(m) | \mathcal{F}_m] - g_l(m)| \\
&= \left| \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot \bar{\omega}_m(\mathbf{y}') - \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot \pi_{\mathbf{y}'}(m) \right| \\
&= \left| \sum_{\mathbf{y}'} g_l^{\mathbf{y}'} \cdot (\bar{\omega}_m(\mathbf{y}') - \pi_{\mathbf{y}'}(m)) \right| \\
&\leq 2g_{\max} \cdot \| \bar{\omega}_m - \boldsymbol{\pi}(m) \|_{TV} \\
&\leq \frac{2g_{\max} K_3}{v(1-\rho)T_m}, \forall l \in L
\end{aligned}$$

where $g_l^{\mathbf{y}'} \leq g_{\max}, \forall \mathbf{y}'$.

Since $\forall l \in L, m, \hat{r}_l$ is upper bounded by some $\bar{r} > 0$ and $r_l(m)$ is bounded by some $r_{\max} > 0$, then we have

$$\| \hat{r}_l - r_l(m) \| \leq \bar{r} + r_{\max}, \forall l \in L$$

Therefore,

$$|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \leq |L| \cdot (\bar{r} + r_{\max}) \cdot \frac{2g_{\max}K_3}{v(1-\rho)T_m} \quad (2.73)$$

$$= \frac{C_3}{T_m} \quad (2.74)$$

where $C_3 = \frac{2|L|(\bar{r}+r_{\max})g_{\max}K_3}{v(1-\rho)}$ is a positive constant.

Now we have

$$\begin{aligned} & \sum_{m=1}^{\infty} |\epsilon(m) \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \\ & \leq C_3 \sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} \\ & < \infty \end{aligned}$$

where the last step follows from condition (2.20).

2.5.5 Proof of Theorem 2.7

Part of the proof hinges on a lemma due to Robbins and Siegmund [57], which is restated as follows.

Lemma 2.13. *Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and let $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots$ be a sequence of sub σ -fields of \mathcal{F} . Let u_m and v_m , $m = 0, 1, 2, \dots$, be non-negative \mathcal{F}_m -measurable random variables. Assume that*

$$E[v_{m+1} | \mathcal{F}_m] \leq v_m - u_m$$

hold with probability 1. Then, with probability 1, the sequence $\{v_m\}$ converges to a non-negative random variable and $\sum_{m=0}^{\infty} u_m < \infty$.

Now we begin the proof of theorem 2.7.

Let $\mathbf{y}^0(m)$ be the state of the Markov chain at the beginning of period m . Define the random vector $U(m) = (\bar{\boldsymbol{\theta}}(m-1), \mathbf{x}(m), \mathbf{r}(m), \mathbf{y}^0(m))$ for $m \geq 1$ and $U(0) = (\mathbf{x}(0), \mathbf{r}(0), \mathbf{y}^0(0))$. For $m \geq 1$, let \mathcal{F}_m be the σ -field generated by $U(0), U(1), \dots, U(m)$, denoted by

$$\mathcal{F}_m = \sigma(U(0), U(1), \dots, U(m)). \quad (2.75)$$

Then $\forall l \in L$, $\bar{g}_l(m)$ can be decomposed into three parts: $\bar{g}_l(m) = g_l(m) + (E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m)) + (\bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m])$.

The first part is the exact subgradient $g_l(m)$.

The second part is the biased estimation error of $g_l(m)$, denoted by

$$B_l(m) \triangleq E[\bar{g}_l(m)|\mathcal{F}_m] - g_l(m) \quad (2.76)$$

The third part is a zero-mean martingale difference noise, denoted by

$$\eta_l(m) \triangleq \bar{g}_l(m) - E[\bar{g}_l(m)|\mathcal{F}_m] \quad (2.77)$$

Therefore,

$$\bar{g}_l(m) = g_l(m) + B_l(m) + \eta_l(m), \forall l \in L \quad (2.78)$$

Recall that $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ is the optimal solution to the problem **RA** (2.15). Thus $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ is a saddle point for $L_\beta(\mathbf{x}, \mathbf{r})$, it follows that

$$L_\beta(\mathbf{x}, \hat{\mathbf{r}}) \leq L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}}) \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}) \quad (2.79)$$

By using $\|\cdot\|$ to denote the *Euclidean* norm, we define the function $V(\cdot, \cdot)$ as follows:

$$V(\mathbf{x}, \mathbf{r}) \triangleq \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \|\mathbf{r} - \hat{\mathbf{r}}\|^2 \quad (2.80)$$

Since

$$\begin{aligned} x_s(m+1) &= [x_s(m) + \epsilon \cdot f_s(m)]_+, \quad \forall s \in S \\ r_l(m+1) &= [r_l(m) - \epsilon \cdot \bar{g}_l(m)]_+, \quad \forall l \in L \end{aligned}$$

by using the fact that the projection $[\cdot]_+$ is non-expansive [5], we have

$$\begin{aligned} \|\mathbf{x}(m+1) - \hat{\mathbf{x}}\|^2 &\leq \|\mathbf{x}(m) + \epsilon \cdot \mathbf{f}(m) - \hat{\mathbf{x}}\|^2 \\ &= \|\mathbf{x}(m) - \hat{\mathbf{x}}\|^2 + 2\epsilon \cdot [\mathbf{x}(m) - \hat{\mathbf{x}}]^T \mathbf{f}(m) \\ &\quad + \epsilon^2 \|\mathbf{f}(m)\|^2. \end{aligned}$$

Similarly, with (2.78), we have

$$\begin{aligned}
& \| \mathbf{r}(m+1) - \hat{\mathbf{r}} \|^2 \leq \| \mathbf{r}(m) - \epsilon \cdot \bar{\mathbf{g}}(m) - \hat{\mathbf{r}} \|^2 \\
& = \| \mathbf{r}(m) - \hat{\mathbf{r}} \|^2 - 2\epsilon \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T \bar{\mathbf{g}}(m) \\
& \quad + \epsilon^2 \| \bar{\mathbf{g}}(m) \|^2 \\
& = \| \mathbf{r}(m) - \hat{\mathbf{r}} \|^2 - 2\epsilon \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T [\mathbf{g}(m) \\
& \quad + \mathbf{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2 \| \bar{\mathbf{g}}(m) \|^2
\end{aligned}$$

By assumption, we know that both $\| \mathbf{f}(m) \|^2$ and $\| \bar{\mathbf{g}}(m) \|^2$ are bounded, we can write that $\| \mathbf{f}(m) \|^2 \leq C_1$ and $\| \bar{\mathbf{g}}(m) \|^2 \leq C_2$, where C_1 and C_2 are positive constants. Using this and the above inequalities, we have that

$$\begin{aligned}
& V(\mathbf{x}(m+1), \mathbf{r}(m+1)) \\
& = \| \mathbf{x}(m+1) - \hat{\mathbf{x}} \|^2 + \| \mathbf{r}(m+1) - \hat{\mathbf{r}} \|^2 \\
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2\epsilon \cdot [(\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) \\
& \quad - (\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m)] - 2\epsilon \cdot [\mathbf{r}(m) - \hat{\mathbf{r}}]^T [\mathbf{B}(m) + \boldsymbol{\eta}(m)] \\
& \quad + \epsilon^2 \cdot (C_1 + C_2)
\end{aligned} \tag{2.81}$$

Since $L_\beta(\mathbf{x}, \mathbf{r})$ is concave in \mathbf{x} and convex in \mathbf{r} , $\mathbf{f}(m)$ and $\mathbf{g}(m)$ are the subgradient vectors of $L_\beta(\mathbf{x}, \mathbf{r})$ with respect to \mathbf{x} and \mathbf{r} respectively, it follows that

$$L_\beta(\mathbf{x}(m), \mathbf{r}(m)) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) \geq (\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) \tag{2.82}$$

$$L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\mathbf{x}(m), \mathbf{r}(m)) \geq -(\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m) \tag{2.83}$$

By the summation of (2.82) and (2.83), we have

$$\begin{aligned}
& (\mathbf{x}(m) - \hat{\mathbf{x}})^T \mathbf{f}(m) - (\mathbf{r}(m) - \hat{\mathbf{r}})^T \mathbf{g}(m) \\
& \leq L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m))
\end{aligned} \tag{2.84}$$

Combining with (2.81) yields that

$$\begin{aligned}
& V(\mathbf{x}(m+1), \mathbf{r}(m+1)) \\
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2\epsilon \cdot [L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m))] \\
& \quad + 2\epsilon \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T [\mathbf{B}(m) + \boldsymbol{\eta}(m)] + \epsilon^2 \cdot (C_1 + C_2)
\end{aligned} \tag{2.85}$$

By methods similar to the proof of Lemma , we have

$$|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \leq \frac{C_3}{T_0}, \quad \forall m$$

where C_3 has the same form as (2.73).

Further,

$$\begin{aligned}
& E[V(\mathbf{x}(m+1), \mathbf{r}(m+1)) | \mathcal{F}_m] \\
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2\epsilon \cdot [L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m))] \\
& \quad + 2\epsilon \cdot [\hat{\mathbf{r}} - \mathbf{r}(m)]^T [\mathbf{B}(m)] + \epsilon^2(m) \cdot (C_1 + C_2)
\end{aligned} \tag{2.86}$$

$$\begin{aligned}
& \leq V(\mathbf{x}(m), \mathbf{r}(m)) + 2\epsilon \cdot [L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m))] \\
& \quad + 2\epsilon \cdot C_3/T_0 + \epsilon^2 \cdot (C_1 + C_2)
\end{aligned} \tag{2.87}$$

Now we define the set

$$\begin{aligned}
H_b & = \{(\mathbf{x}, \mathbf{r}) : 0 \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}) - L_\beta(\mathbf{x}, \hat{\mathbf{r}}) \\
& \quad < \frac{1}{b} + C_3/T_0 + \epsilon \frac{(C_1 + C_2)}{2}\}
\end{aligned} \tag{2.88}$$

Where b is a positive constant.

We also define the sequence $(\mathbf{x}'(m), \mathbf{r}'(m))$ as follows:

$$(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) \tag{2.89}$$

$$= \begin{cases} (\mathbf{x}(m+1), \mathbf{r}(m+1)) & \text{if } (\mathbf{x}'(m), \mathbf{r}'(m)) \notin H_b \\ (\hat{\mathbf{x}}, \hat{\mathbf{r}}) & \text{if } (\mathbf{x}'(m), \mathbf{r}'(m)) \in H_b \end{cases} \tag{2.90}$$

Thus the process $\{(\mathbf{x}'(m), \mathbf{r}'(m))\}$ is identical to the process $\{(\mathbf{x}(m), \mathbf{r}(m))\}$, until $\{(\mathbf{x}(m), \mathbf{r}(m))\}$ enters the set H_b .

Given m , we discuss two cases.

Case 1: $(\mathbf{x}'(m), \mathbf{r}'(m)) \in H_b$.

Since $(\mathbf{x}'(m), \mathbf{r}'(m)) \in H_b$ and $(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) = (\hat{\mathbf{x}}, \hat{\mathbf{r}})$, we have $V(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) = 0$ and $V(\mathbf{x}'(m), \mathbf{r}'(m)) \geq 0$, yielding

$$E[V(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) | \mathcal{F}_m] \leq V(\mathbf{x}'(m), \mathbf{r}'(m)) \quad (2.91)$$

Case 2: $(\mathbf{x}'(m), \mathbf{r}'(m)) \notin H_b$.

Then we have $(\mathbf{x}'(m), \mathbf{r}'(m)) = (\mathbf{x}(m), \mathbf{r}(m))$ and $(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) = (\mathbf{x}(m+1), \mathbf{r}(m+1))$. Using (2.87), we have

$$\begin{aligned} & E[V(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) | \mathcal{F}_m] \\ & \leq V(\mathbf{x}'(m), \mathbf{r}'(m)) + 2\epsilon[L_\beta(\mathbf{x}'(m), \hat{\mathbf{r}}) - L_\beta(\hat{\mathbf{x}}, \mathbf{r}'(m))] \\ & \quad + 2\epsilon \cdot C_3/T_0 + \epsilon^2 \cdot (C_1 + C_2) \end{aligned} \quad (2.92)$$

Observe that when $(\mathbf{x}'(m), \mathbf{r}'(m)) \notin H_b$,

$$L_\beta(\hat{\mathbf{x}}, \mathbf{r}'(m)) - L_\beta(\mathbf{x}'(m), \hat{\mathbf{r}}) \geq \frac{1}{b} + C_3/T_0 + \epsilon \frac{(C_1 + C_2)}{2} \quad (2.93)$$

Therefore, by combining (2.92) and (2.93), we obtain that

$$E[V(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) | \mathcal{F}_m] \leq V(\mathbf{x}'(m), \mathbf{r}'(m)) - \frac{2\epsilon}{b} \quad (2.94)$$

So from (2.91) and (2.94), we can write

$$E[V(\mathbf{x}'(m+1), \mathbf{r}'(m+1)) | \mathcal{F}_m] \leq V(\mathbf{x}'(m), \mathbf{r}'(m)) - \Delta(m) \quad (2.95)$$

where

$$\Delta(m) = \begin{cases} 0 & \text{if } (\mathbf{x}'(m), \mathbf{r}'(m)) \in H_b \\ \frac{2\epsilon}{b} & \text{if } (\mathbf{x}'(m), \mathbf{r}'(m)) \notin H_b \end{cases} \quad (2.96)$$

Observe that (2.95) satisfies the condition of Lemma 2.13. Therefore, it follows that with probability 1,

$$\sum_{m=0}^{\infty} \Delta(m) < \infty \quad (2.97)$$

However, this is possible only if $\Delta(m) = 0$ for all sufficiently large m . So for any given $b > 0$, with probability 1, we have $(\mathbf{x}'(m), \mathbf{r}'(m)) \in H_b$ for all sufficiently large m . This also means that for any given $b > 0$, with probability 1, $(\mathbf{x}(m), \mathbf{r}(m)) \in H_b$ for all sufficiently large m . By letting $b \rightarrow \infty$, we have that with probability 1, and for all sufficiently large m , $(\mathbf{x}(m), \mathbf{r}(m))$ belongs to the set

$$\{(\mathbf{x}, \mathbf{r}) : 0 \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}) - L_\beta(\mathbf{x}, \hat{\mathbf{r}}) \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\}$$

It follows that with probability 1, for all sufficiently large m ,

$$0 \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) - L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}$$

Then we have

$$L_\beta(\mathbf{x}(m), \mathbf{r}(m)) \leq L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) \quad (2.98)$$

$$\leq L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) + \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2} \quad (2.99)$$

$$\leq L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}}) + \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2} \quad (2.100)$$

and

$$L_\beta(\mathbf{x}(m), \mathbf{r}(m)) \geq L_\beta(\mathbf{x}(m), \hat{\mathbf{r}}) \quad (2.101)$$

$$\geq L_\beta(\hat{\mathbf{x}}, \mathbf{r}(m)) - \left(\frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\right) \quad (2.102)$$

$$\geq L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}}) - \left(\frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\right) \quad (2.103)$$

Therefore, with probability 1, for all sufficiently large m ,

$$|L_\beta(\mathbf{x}(m), \mathbf{r}(m)) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}. \quad (2.104)$$

So with probability 1, $\{(\mathbf{x}(m), \mathbf{r}(m))\}$ converge to the neighborhood of $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$.

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\} \quad (2.105)$$

Further, as $T_0 \rightarrow \infty$ and $\epsilon \rightarrow 0$, we know that $(\mathbf{x}(m), \mathbf{r}(m)) \rightarrow (\hat{\mathbf{x}}, \hat{\mathbf{r}})$ with probability 1. This concludes the proof.

2.5.6 Proof of Theorem 2.9

We denote M as the original configuration hopping Markov chain with exact system performances, and M' as the corresponding extended Markov chain with inaccurately observed system performances. For the convenience of expression, for all $f \in \mathcal{F}, j \in \{-n_f, \dots, n_f\}$, we use f_j to represent the state $(f, \phi(f) + \frac{j}{n_f} \Delta_f)$ in the extended Markov chain M' .

Therefor, given direct transitions between configuration f and f' in the original configuration hopping Markov chain M , there are direct transitions between states f_j and f'_k ($\forall j \in \{-n_f, \dots, n_f\}, k \in \{-n_{f'}, \dots, n_{f'}\}$) in the extended Markov chain M' . Following (2.31) and (2.32), we have the corresponding transition rates

$$q_{f_j, f'_k} = \eta_{f'_k} \alpha \frac{1}{\exp[\beta(\phi(f) + \frac{j}{n_f} \Delta_f)]} \quad (2.106)$$

$$q_{f'_k, f_j} = \eta_{f_j} \alpha \frac{1}{\exp[\beta(\phi(f') + \frac{k}{n_{f'}} \Delta_{f'})]} \quad (2.107)$$

$$\forall j \in \{-n_f, \dots, n_f\}, k \in \{-n_{f'}, \dots, n_{f'}\} \quad (2.108)$$

where $\sum_{j=-n_f}^{n_f} \eta_{f_j} = 1$ and $\sum_{k=-n_{f'}}^{n_{f'}} \eta_{f'_k} = 1$.

Now we compute the stationary distribution of states for Markov chain M' . By detailed balance equation, we have

$$p_{f_j} q_{f_j, f'_k} = p_{f'_k} q_{f'_k, f_j}, \forall j \in \{-n_f, \dots, n_f\}, k \in \{-n_{f'}, \dots, n_{f'}\} \quad (2.109)$$

Then we have

$$p_{f_j} \cdot \frac{1}{\eta_{f_j} \cdot \exp(\beta(\phi(f) + \frac{j}{n_f} \Delta_f))} = p_{f'_k} \cdot \frac{1}{\eta_{f'_k} \cdot \exp(\beta(\phi(f') + \frac{k}{n_{f'}} \Delta_{f'}))}, \quad (2.110)$$

$$\forall j \in \{-n_f, \dots, n_f\}, k \in \{-n_{f'}, \dots, n_{f'}\}$$

So

$$\frac{p_{f_0}}{\eta_{f_0} \cdot \exp(\beta\phi(f))} = \frac{p_{f'_0}}{\eta_{f'_0} \cdot \exp(\beta\phi(f'))} \quad (2.111)$$

and

$$\frac{p_{f'_k}}{p_{f'_0}} = \frac{\eta_{f'_k}}{\eta_{f'_0}} \cdot \exp\left(\beta \frac{k}{n_{f'}} \Delta_{f'}\right), \forall k \in \{-n_{f'}, \dots, n_{f'}\}. \quad (2.112)$$

Now consider an arbitrary state \hat{f}_0 in Markov chain M' , where $\hat{f} \in \mathcal{F}$ and $\hat{f} \neq f, f'$. Since state space of M' is connected, we can always find a path to connect \hat{f}_0 and f_0 through a series of adjacent states $\tilde{f}(1)_0, \dots, \tilde{f}(L)_0$, and $f_0 = \tilde{f}(1)_0, \tilde{f}(L)_0 = \hat{f}_0$. Therefore,

$$\frac{p_{\hat{f}_0}}{p_{f_0}} = \prod_{l=1}^{L-1} \frac{p_{\tilde{f}(l+1)_0}}{p_{\tilde{f}(l)_0}} \quad (2.113)$$

by (2.111) we have

$$\frac{p_{\tilde{f}(l+1)_0}}{\eta_{\tilde{f}(l+1)_0} \cdot \exp(\beta \phi(\tilde{f}(l+1)))} = \frac{p_{\tilde{f}(l)_0}}{\eta_{\tilde{f}(l)_0} \cdot \exp(\beta \phi(\tilde{f}(l)))} \quad (2.114)$$

Then

$$\frac{p_{\hat{f}_0}}{\eta_{\hat{f}_0} \cdot \exp(\beta \phi(\hat{f}))} = \frac{p_{f_0}}{\eta_{f_0} \cdot \exp(\beta \phi(f))} \quad (2.115)$$

By (2.112) and (2.115), we know that $\forall f \in \mathcal{F}$,

$$\frac{p_{f_0}}{\eta_{f_0} \cdot \exp(\beta \phi(f))} \text{ is a constant} \quad (2.116)$$

and

$$\frac{p_{f_j}}{p_{f_0}} = \frac{\eta_{f_j}}{\eta_{f_0}} \cdot \exp\left(\beta \frac{j}{n_f} \Delta_f\right), \forall j \in \{-n_f, \dots, n_f\}. \quad (2.117)$$

On the other hand, we have

$$\sum_{f \in \mathcal{F}} \sum_{j=-n_f}^{n_f} p_{f_j} = 1 \quad (2.118)$$

By (2.116), (2.117) and (2.118), we obtain the stationary distribution of Markov chain M' as follows:

$$\tilde{p}_{f_j} = \frac{\eta_{f_j} \cdot \exp(\beta(\phi(f) + \frac{j}{n_f} \Delta_f))}{\sum_{f' \in \mathcal{F}} \sum_{k=-n_{f'}}^{n_{f'}} \eta_{f'_k} \cdot \exp(\beta(\phi(f') + \frac{k}{n_{f'}} \Delta_{f'}))}, \forall f \in \mathcal{F}, j \in \{-n_f, \dots, n_f\}. \quad (2.119)$$

The stationary distribution for configuration $f \in \mathcal{F}$ in Markov chain M' is the probability distribution of aggregate states $f_j, j \in \{-n_f, \dots, n_f\}$, i.e.,

$$\bar{p}_f = \sum_{j=-n_f}^{n_f} \tilde{p}_{f_j} \quad (2.120)$$

Let

$$\alpha_f \triangleq \sum_{j=-n_f}^{n_f} \eta_{f_j} \cdot \exp(\beta \frac{j}{n_f} \Delta_f), \forall f \in \mathcal{F} \quad (2.121)$$

Then we have

$$\bar{p}_f = \frac{\alpha_f \exp(\beta \phi(f))}{\sum_{f' \in \mathcal{F}} \alpha_{f'} \exp(\beta \phi(f'))}, \forall f \in \mathcal{F}. \quad (2.122)$$

We know that

$$p_f^* = \frac{\exp(\beta \phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta \phi(f'))}, \forall f \in \mathcal{F}. \quad (2.123)$$

Let

$$\bar{\alpha} \triangleq \frac{\sum_{f' \in \mathcal{F}} \alpha_{f'} \exp(\beta \phi(f'))}{\sum_{f' \in \mathcal{F}} \exp(\beta \phi(f'))} \quad (2.124)$$

It is not hard to see that $\frac{p_f^*}{\bar{p}_f} = \frac{\bar{\alpha}}{\alpha_f}$, so

$$p_f^* \geq \bar{p}_f \text{ iff } \alpha_f \leq \bar{\alpha} \quad (2.125)$$

Since total variation distance

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) = \frac{1}{2} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \quad (2.126)$$

$$= \sum_{f \in A} (p_f^* - \bar{p}_f) \quad (2.127)$$

where $A \triangleq \{f \in \mathcal{F} : p_f^* \geq \bar{p}_f\}$.

By (2.125), we know $A = \{f \in \mathcal{F} : \alpha_f \leq \bar{\alpha}\} \subset \mathcal{F}$.

Therefore, $\forall f \in A$,

$$p_f^* - \bar{p}_f = \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} - \frac{\alpha_f \exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \alpha_{f'} \exp(\beta\phi(f'))} \quad (2.128)$$

$$= \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} - \frac{\alpha_f \exp(\beta\phi(f))}{\bar{\alpha} \sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} \quad (2.129)$$

$$= \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} \left[1 - \frac{\alpha_f}{\bar{\alpha}}\right] \quad (2.130)$$

Since $\sum_{j=-n_f}^{n_f} \eta_{f_j} = 1$ and $\forall j \in \{-n_f, \dots, n_f\}$

$$\exp(\beta \frac{j}{n_f} \Delta_f) \geq \exp(-\beta \Delta_f) \geq \exp(-\beta \Delta_{\max}) \quad (2.131)$$

$$\exp(\beta \frac{j}{n_f} \Delta_f) \leq \exp(\beta \Delta_f) \leq \exp(\beta \Delta_{\max}), \quad (2.132)$$

by (2.121) we know that $\forall f \in \mathcal{F}$

$$\alpha_f \geq \sum_{j=-n_f}^{n_f} \eta_{f_j} \cdot \exp(-\beta \Delta_{\max}) = \exp(-\beta \Delta_{\max}) \quad (2.133)$$

$$\alpha_f \leq \sum_{j=-n_f}^{n_f} \eta_{f_j} \cdot \exp(\beta \Delta_{\max}) = \exp(\beta \Delta_{\max}) \quad (2.134)$$

Then by (2.124) we have $\bar{\alpha} \leq \exp(\beta \Delta_{\max})$. Therefore,

$$1 - \frac{\alpha_f}{\bar{\alpha}} \leq 1 - \frac{\exp(-\beta \Delta_{\max})}{\exp(\beta \Delta_{\max})} = 1 - \exp(-2\beta \Delta_{\max}), \forall f \in A \subset \mathcal{F}. \quad (2.135)$$

So by (2.130), we have $\forall f \in A$,

$$p_f^* - \bar{p}_f = \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} \left[1 - \frac{\alpha_f}{\bar{\alpha}}\right] \quad (2.136)$$

$$\leq \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} (1 - \exp(-2\beta \Delta_{\max})). \quad (2.137)$$

Then

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) = \sum_{f \in A} (p_f^* - \bar{p}_f) \quad (2.138)$$

$$\leq \sum_{f \in A} \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} (1 - \exp(-2\beta\Delta_{\max})) \quad (2.139)$$

$$\leq \sum_{f \in \mathcal{F}} \frac{\exp(\beta\phi(f))}{\sum_{f' \in \mathcal{F}} \exp(\beta\phi(f'))} (1 - \exp(-2\beta\Delta_{\max})) \quad (2.140)$$

$$= 1 - \exp(-2\beta\Delta_{\max}) \quad (2.141)$$

Therefore,

$$|\mathbf{p}^* \mathbf{x}^T - \bar{\mathbf{p}} \mathbf{x}^T| = \left| \sum_{f \in \mathcal{F}} (p_f^* - \bar{p}_f) x_f \right| \quad (2.142)$$

$$\leq x_{\max} \sum_{f \in \mathcal{F}} |p_f^* - \bar{p}_f| \quad (2.143)$$

$$= 2x_{\max} d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \quad (2.144)$$

$$\leq 2x_{\max} (1 - \exp(-2\beta\Delta_{\max})) \quad (2.145)$$

This concludes the proof.

2.5.7 Proof of Theorem 2.10

From the proof of Theorem 2.9, we know that to tighten the bound, we need to tighten the lower bound of α_f (2.121).

By AM-GM inequality, we know that the arithmetic mean of a list of non-negative real numbers is greater than or equal to the geometric mean of the

same list. So we can see that $\forall f \in \mathcal{F}$,

$$\alpha_f = \sum_{j=-n_f}^{n_f} \eta_{f_j} \cdot \exp\left(\beta \frac{j}{n_f} \Delta_f\right) \quad (2.146)$$

$$\geq (2n_f + 1) \left[\prod_{j=-n_f}^{n_f} \eta_{f_j} \cdot \exp\left(\beta \frac{j}{n_f} \Delta_f\right) \right]^{\frac{1}{2n_f+1}} \quad (2.147)$$

$$= (2n_f + 1) \left[\prod_{j=-n_f}^{n_f} \eta_{f_j} \right]^{\frac{1}{2n_f+1}} \quad (2.148)$$

Therefore, finding the maximum lower bound is equivalent to solving the following optimization problem:

$$\text{MLB : } \max \prod_{j=-n_f}^{n_f} \eta_{f_j} \quad (2.149)$$

$$\text{s.t. } \sum_{j=-n_f}^{n_f} \eta_{f_j} = 1, \eta_{f_j} \geq 0, \forall j \quad (2.150)$$

It is not hard to see the optimal solution is that: $\forall f \in \mathcal{F}$, $\eta_{f_j} = \frac{1}{2n_f+1}$, $\forall j \in \{-n_f, \dots, n_f\}$. So the optimal error distribution is a uniform distribution. Given this uniform distribution, the lower bound of α_f , $\forall f \in \mathcal{F}$ is 1. Then following the same line as proof for Theorem 2.9, we can see that

$$d_{TV}(\mathbf{p}^*, \bar{\mathbf{p}}) \leq 1 - \exp(-\beta \Delta_{\max}). \quad (2.151)$$

Chapter 3

Wireless Networks With Deterministic Channel Models

All exact science is dominated by
the idea of approximation.

Bertrand Russell

In this chapter, we apply Markov approximation framework to cross-layer optimization of wireless networks with deterministic channel model. Proposed in [2, 3], this deterministic channel model is a simple abstraction of the physical layer that effectively captures the effect of channel strength, broadcast and superposition in wireless channels. In contrast, existing work on cross-layer optimization for wireless networks adopts simple physical-layer models, i.e., treating interference as noise. Within the Network Utility Maximization (NUM) framework, we study the cross-layer optimization for wireless networks based on this deterministic channel model. First, we extend the well-studied conflict graph model to capture the flow interactions over the deterministic channels and characterize the feasible rate region. Then guided by Markov approximation framework, we design distributed algorithms for general wireless multi-hop networks with both link-centric formulation and node-centric formulation. The convergence of algorithms is proved by applying Lyapunov stability

theorem and stochastic approximation method. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant step and constant update interval. Our numerical evaluations validate the analytical results and show the advantage of deterministic channel model over simple physical layer models such as treating interference as noise.

3.1 Introduction

Since Kelly's seminal work [35], the network utility maximization (NUM) framework [12] has attracted significant attentions for cross-layer optimization of wireless networks. In this framework, network protocols are understood as distributed algorithms that maximize aggregate user utility under network resource constraints.

Existing studies on wireless NUM [12] mainly base on a simple physical layer assumptions: treating interference as noise. However, it is known that "treating interference as noise" is in general sub-optimal and higher rates can be achieved by using more advanced physical layer coding techniques, such as superposition coding [14], successive interference cancellation [14] and interference alignment [7].

Recently, a deterministic channel model was proposed in [2,3]. This model quantizes the transmitted signal into different signal strength levels and represents the noise by a deterministic cut-off threshold rather than a random variable. Consequently, the bits received below the noise threshold are discarded by the receiver. This deterministic channel model simplifies the wireless flow interaction by eliminating the noise and allows studies to focus on interferences among transmissions. The deterministic channel provides an accurate, yet simple, abstraction of the physical layer that can be utilized effectively for the cross-layer protocol design [3].

On the other hand, the optimal scheduling is still a challenging problem for

wireless networks [49, 62], irrespective of the underlying physical layer model. Though Maximum Weight Scheduling (MWS) algorithm proposed in the seminal work [62] is shown to be throughput-optimal, it has the exponential computational complexity [49] even in a centralized manner, i.e., NP-complete. Further, the MWS is not amenable to distributed implementation. Some low-complexity alternatives [48, 49] of MWS have been proposed. However, they achieve only a small fraction of maximum throughput region or maximum utility.

An adaptive CSMA based distributed scheduling algorithm was proposed in [31] recently. This adaptive CSMA algorithm is based on the product-form distribution of the idealized CSMA Markov chain and adaptively changing the channel access rate. It has been shown to be throughput-optimal and utility-optimal asymptotically [10, 30, 31, 50, 55]. Inspired by this series of work [10, 30, 31, 50, 55, 58, 59, 66], we propose in [10] a Markov approximation framework for synthesizing distributed algorithms for general combinatorial network optimization problems, including the Maximum Weight Scheduling problem as a special case.

Motivated by this progress, in this chapter, we study the cross-layer optimization for wireless networks with deterministic channel models, i.e., jointly optimizing flow control and scheduling. The key results are as follows:

- **Characterization of the feasible rate region by extended conflict graph based model:** Existing conflict-graph model [28, 49] are based on simple physical-layer assumptions. We extend the conflict-graph model to capture the flow interactions over the deterministic channels. In general, the resulted feasible rate region is a subset of the information-theoretic capacity region. In some special cases such as single-hop multiple-access networks, this feasible rate region is shown to be the same as the information-theoretic capacity region.

- **Distributed solutions for NUM over general wireless multi-hop networks with deterministic channel models:** We consider wireless networks with both link-centric formulation and node-centric formulation [49]. By applying Markov approximation framework [10] and standard Lagrange dual decomposition method, we construct distributed algorithms to approximately solve the cross-layer optimization problem within the NUM framework.
- **Convergence of the primal-dual flow control algorithm with or without time-scale separation assumption:** Existing work [30, 31, 50] focuses on the dual algorithm and proves its convergence. In this paper, we focus on the primal-dual algorithm because it has smoother dynamics than that of the dual algorithm. With the time-scale separation assumption that Markov chain converges to its stationary distribution instantaneously, the proposed primal-dual algorithm is shown to converge to the optimal solutions globally asymptotically. Without such time-scale separation assumption, the resulted stochastic primal-dual algorithm is shown to converge to a bounded neighborhood of the same optimal solutions with probability one under constant step size and constant update interval.

The remainder of this chapter is organized as follows. In Section 3.2, we introduce the system model and the problem formulation. In Section 3.3 and Section 3.4, we propose the distributed solution for general multi-hop wireless networks with link-centric formulation and node-centric formulation respectively. Numerical results are provided in Section 3.5, and conclusions are drawn in Section 3.6.

3.2 System Model and Problem Formulation

We consider a wireless network with a set of N users (source-destination pairs), denoted as $S = \{1, \dots, N\}$. Each user is associated with a sending rate x_s , as well as a utility function $U_s(x_s)$ that measures the efficiency and fairness of resource allocation algorithms. We assume the utility function to be twice differentiable, increasing and strictly concave [12]. For example, the α -fairness utility function [53] is defined in the following:

$$U^\alpha(x) = \begin{cases} \frac{x^{1-\alpha}}{1-\alpha} & \text{if } \alpha \neq 1 \\ \log x & \text{Otherwise,} \end{cases}$$

where $\alpha = 1$ corresponds to proportional fairness and $\alpha \rightarrow \infty$ corresponds to max-min fairness.

The utility maximization problem is as follows

$$\begin{aligned} \mathbf{P1:} \quad & \max_{\mathbf{x} \geq 0} \sum_{s \in S} U_s(x_s) & (3.1) \\ \text{s.t.} \quad & \mathbf{x} \in \{\text{feasible rate region}\}. \end{aligned}$$

where \mathbf{x} is the vector of user's rate (we use bold symbols to denote vectors through the whole paper) and the feasible rate region is a function of information transmission constraints. In general, there are two different ways of stating the feasible rate region: *link-centric formulation* and *node-centric formulation* [49]. In link-centric formulation, the capacity constraints are stated as balance equations for each link and the routes for each user are predetermined. While in node-centric formulation, the capacity constraints are stated as balance equations of incoming rates and outgoing rates for each node.

3.2.1 Deterministic Channel Model

In this paper, we study the problem **P1** over the feasible rate region based on the deterministic channel model. A formal definition of this channel model is given as follows.

Definition 3.1. (Definition of the deterministic model [2,3]) Consider a wireless network consisting of a set of nodes V and a set of channels, where $K = |V|$ is the number of nodes. Communication from node i to node j has a non-negative integer gain $\rho_{(i,j)}$ associated with it. This number models the channel gain in a corresponding Gaussian setting. At each time t , node i transmits a vector $\mathbf{x}_i[t] \in \mathbb{F}_2^\omega$ and receives a vector $\mathbf{y}_i[t] \in \mathbb{F}_2^\omega$ where $\omega = \max_{i,j}(\rho_{(i,j)})$. The received signal at each node is a deterministic function of the transmitted signals at the other nodes, with the following input-output relation: if the nodes in the network transmit $\mathbf{x}_1[t], \mathbf{x}_2[t], \dots, \mathbf{x}_K[t]$ then the received signal at node j , $1 \leq j \leq K$ is: $\mathbf{y}_j[t] = \sum_{k=1}^K \mathbf{W}^{\omega - \rho_{k,j}} \mathbf{x}_k[t]$ for all $1 \leq k \leq V$, where \mathbf{W} is the $\omega \times \omega$ shift matrix and the summation and multiplication is in \mathbb{F}_2 .

An example is shown in Fig. 3.1. In this figure, each signal is a sequence of bits at different signal levels, with the highest signal level being the most significant bit (MSB) and the lowest level being the least significant bit (LSB). The transmit and received signal levels are sorted from MSB (top) to LSB (bottom). The channel gain between two nodes i and j indicates how many of the first MSB transmitted signal levels of node i are received at destination node j .

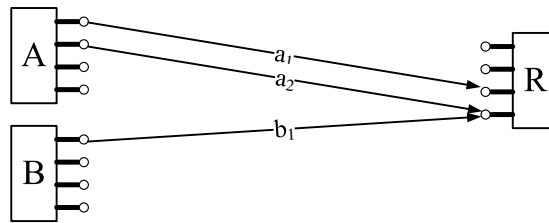


Figure 3.1: One example of deterministic channel based wireless network with channel gains $\rho_{AR} = 2$ and $\rho_{BR} = 1$. Each rectangle represents a transmitter or receiver node, and each knob attaching to a rectangle represents a signal level and what is sent on it is a bit.

There is a constraint on configurations of sub-links belonging to the same channel [3]: activated sub-links are consecutive from high signal levels to low signal levels, *not necessary* starting from MSB. When non-consecutive signal

levels are used in the transmission, we need to use superposition code to encode the disjoint set of sublinks belonging to the same channel. Consequently, the complex code generation induces the loss of energy efficiency and the approximation accuracy to the Gaussian model [3], which depends on the number of “abruptones” in signal levels. There is a tradeoff between 1) the deterministic model’s rate region reduction caused by the consecutive signal level constraint, and 2) its approximation accuracy to the Gaussian model. This fundamental tradeoff deserves its own investigation [3]. In this paper, we focus on achieving a good network-wise performance with practically implementable schemes. As such, we cast the consecutive signal level constraint to get a higher approximation accuracy. So for the channel l with the channel gain ρ_l , there are $\frac{\rho_l(\rho_l+1)}{2}$ configurations of channel l , i.e., possible combinations of consecutive sub-links within the channel l . For example, for a channel l with channel gain 3 and three sub-links l_1, l_2, l_3 , there are 6 configurations: $\{l_1\}$, $\{l_1, l_2\}$, $\{l_1, l_2, l_3\}$, $\{l_2\}$, $\{l_2, l_3\}$, and $\{l_3\}$.

3.2.2 Conflict Graph Based Model

The wireless network is represented as a graph $G=(V, L)$, where V is the set of nodes and L is the set of links(channels) between nodes. Each link(channel) $(i, j) \subset L$ consists of $\rho_{(i,j)}$ sub-links (each with one-unit capacity), where $i, j \in V$ and $\rho_{(i,j)}$ is the channel gain from node i to node j .

Note that for a general wireless network with physical layer assumptions such as treating interference as noise, the existing conflict graph model [28, 49] is equivalent to the collision model, where if two interfering links transmit packets simultaneously, both packets are dropped. However, the interference itself actually carries information and has structure that can be potentially be exploited in mitigating its effect.

On the other hand, for wireless networks under deterministic channel modeling, we can exploit the structure of interference by superposition coding and interference cancelation [3]. Unlike the collision model where entire messages are lost when there is collision, the most significant bits of the stronger users remain intact.

Therefore, by treating interference as noise, the physical layer model deals the interference in the link level, a coarse granularity of interference; while deterministic channel model deals the interference in the sub-link level, a fine granularity of interference. Since existing conflict graph model can only be applied to interference in the link level, we need an extended conflict graph model, which can be applied to interference in the sub-link level.

We propose an extended conflict graph model to capture the interference over the deterministic channels, shown as follows:

Definition 3.2. *The conflict graph G_c of a graph G is an undirected graph $G_c = (V_c, A)$, where vertex set V_c corresponds to possible configurations for all channels(links) in G , and edge set A represents connections of adjacent vertices, i.e., conflicts of configurations. Two vertices are adjacent if for the corresponding two configurations, either they belong to the same set of configurations for one channel(link) or there exist two sub-links within two configurations, one for each, conflicting with each other, i.e., intersecting at the same signal level of one node.*

Note that all possible configurations for the same channel are conflict with others, since given any one channel, only one configuration for this channel can be activated at a time. Then we introduce the definition of independent set, shown as follows:

Definition 3.3. *An independent set of the conflict graph $G_c = (V_c, A)$ is a subset $M \subset V_c$ of vertices that no two of which are adjacent (i.e., $(s, t) \notin A$ for*

all $s, t \in M$), i.e., the set of channel (link) configurations that can be activated simultaneously.

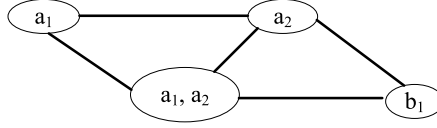


Figure 3.2: Corresponding conflict graph for the network shown in Fig. 1. All independent sets are: \emptyset , $\{a_1\}$, $\{a_2\}$, $\{b_1\}$, $\{a_1, a_2\}$, and $\{\{a_1\}, \{b_1\}\}$.

Here we give an example. For the wireless network shown in Fig. 3.1, the corresponding conflict graph and independent sets are shown in Fig. 3.2. Note that in Fig. 3.1, sub-link a_2 conflicts with sub-link b_1 , i.e., intersecting at the same signal level of node R .

3.2.3 Extended Conflict Graph Model vs. Existing Conflict Graph Model.

In this subsection, we discuss the advantage of using link-configuration based extended conflict graph model over sublink based existing conflict graph model.

First, we discuss the fundamental difference in complexity between these two models. For the constraint of consecutive signal levels, the extended conflict graph model can easily handle it based on link-configurations, while the existing conflict graph model needs global coordination between nodes. Now we consider a k -hop flow and suppose each link has ρ sublinks. By the extended conflict graph model, each link maintains $\frac{\rho(\rho+1)}{2}$ link configurations, and there are only one k -hop path based on link configurations. While by the existing conflict graph model, each link maintains ρ sublinks and there are at least ρ^k k -hop paths based on sublinks. An example is shown in Fig.3.3, where a flow transverses from A to E , $k = 4$ and $\rho = 3$.

In general, given a deterministic network with $\Theta(\rho)$ sublinks per link and $\Theta(k)$ hops per-flow, by extended conflict graph model, we have $\Theta(\rho^2)$ sublinks

per-link and $\Theta(1)$ path per-flow; whereas by existing conflict graph model, we have $\Theta(\rho)$ sublinks per-link and $\Theta(\rho^k)$ paths per-flow.

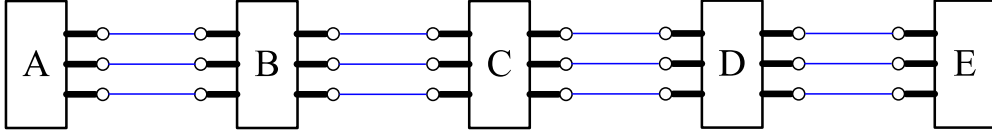


Figure 3.3: A deterministic wireless network with four links AB , BC , CD and DE . Each link has three sublinks. There is a flow transverses from A to E . By the extended conflict graph model, each link maintains six link configurations and there is only one four-hop path. In contrast, by the existing conflict graph model on sublinks, each link maintains three sublinks and there are at least eighty-one four-hop paths.

Second, we discuss the impacts of running primal, dual, and primal-dual resource allocation algorithms over the underlying conflict graph model. In general, the dual algorithm and the primal-dual algorithm solve the resource allocation problem exactly, whereas the primal algorithm solves it only approximately [61]. Further, given the underlying multipath setting, the objective functions are not strictly concave (or convex). Standard dual gradient algorithms fail to work since the gradient is not everywhere defined [11, 61]. Dual subgradient algorithms are proposed as alternatives. However, convergence of dual variables in these subgradient algorithms are typically slow, and recovering optimal primal variables from optimal dual variables requires solving another optimization problem [61]. What is more, it is well known that multipath setting usually incurs the instability of primal-dual algorithms [11, 65].

By utilizing existing conflict graph model, we have the multipath setting, which incurs troubles for running resource allocation algorithms. Therefore, we propose the extended conflict graph model instead of adopting the existing conflict graph model.

Now we take into account the broadcast advantage of wireless medium, i.e., a single packet transmission might be overheard by a subset of receiver nodes within range of the transmitter. Therefore a one-hop wireless transmission

with broadcast advantage can be represented by a hyperlink $(s, \{R_s\})$, where s is the transmitter, $\{R_s\}$ is the set of receivers, and $|R_s| \geq 1$. By modeling wireless networks with broadcast advantages as hypergraphs, we can directly extend the definitions of conflict graph and associated independent set to conflict hypergraph model. When $|R_s| = 1, \forall s$, the conflict hypergraph model degenerates into the above conflict graph model. We omit the formal definition of conflict hypergraph here and show an example instead. The scenario is shown in Fig. 3.4. Corresponding conflict graph and independent sets are shown in Fig. 3.5. Note that in Fig. 3.4, when subnode b_1 sends one packet to subnode d_4 , the subnode c_4 overhears this packet. So if at this time subnode a_2 also sends one packet to c_4 , then c_4 receives two corrupted packets, i.e., two simultaneous transmissions $b_1 \rightarrow d_4$ and $a_2 \rightarrow c_4$ collide with each other.

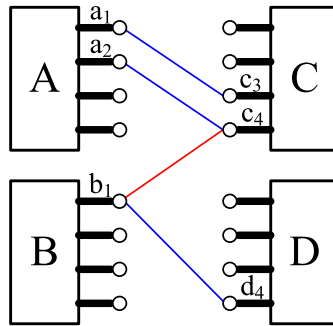


Figure 3.4: Example of wireless networks with broadcast advantages. For transmission initiated by a_1 , its receiver is c_3 , for transmission initiated by a_2 , its receiver is c_4 , and for transmission initiated by b_1 , its receivers are c_4 and d_4 . Possible transmissions: $(a_1, \{c_3\})$, $(a_2, \{c_4\})$, $[(a_1, \{c_3\}), (a_2, \{c_4\})]$, $(b_1, \{c_4\})$, $(b_1, \{d_4\})$ and $(b_1, \{c_4, d_4\})$.

3.2.4 Feasible Rate Region

The feasibility of simultaneous transmissions can be captured by the extended conflict graph for wireless networks with deterministic channel model. As a result, the feasible rate region is characterized as a convex hull of the feasible rates supported by possible individual independent sets on this extended

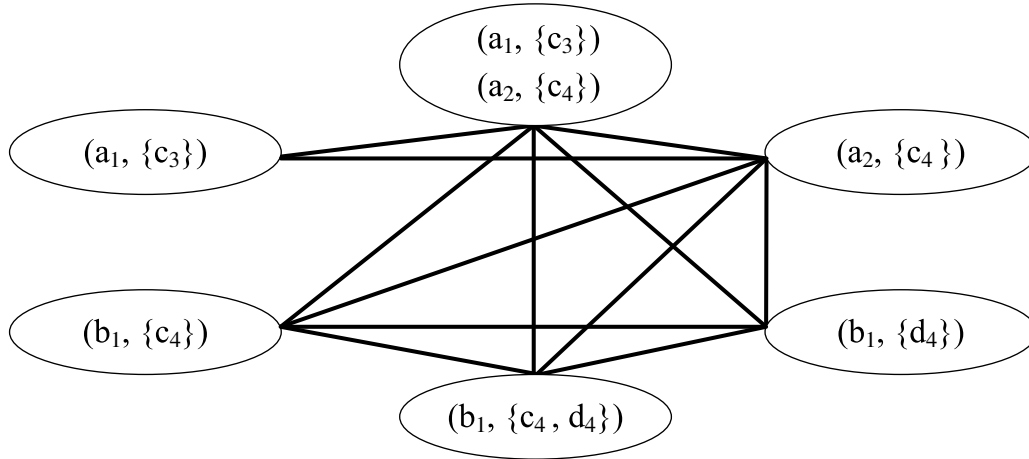


Figure 3.5: Corresponding conflict graph for the network shown in Fig. 3. All independent sets are: \emptyset , $\{(a_1, \{c_3\})\}$, $\{(a_2, \{c_4\})\}$, $\{(a_1, \{c_3\}), (a_2, \{c_4\})\}$, $\{(b_1, \{c_4\})\}$, $\{(b_1, \{d_4\})\}$, $\{(b_1, \{c_4, d_4\})\}$, $\{(a_1, \{c_3\}), (b_1, \{c_4\})\}$, $\{(a_1, \{c_3\}), (b_1, \{d_4\})\}$, and $\{(a_1, \{c_3\}), (b_1, \{c_4, d_4\})\}$.

conflict graph [28].

For some important cases such as deterministic multiple-access channels, we have the following result:

Proposition 3.4. *For the deterministic multiple-access channel, the conflict graph based rate region is equal to the information-theoretic capacity region.*

The proof is relegated to Appendix 3.7.1.

In general, the conflict graph based rate region is the subset of the information-theoretic capacity region. One example is shown in Fig. 3.6 [6]. It is not hard to see that the rate tuple (3,2) is out of the conflict graph based rate region, since in our conflict graph model, two bits arrive simultaneously at the same signal levels of the receiver are dropped.

For general wireless networks, the characterization of the information-theoretic capacity region is open. Therefore, in this paper, we focus on the conflict graph based rate region. To save the heavy notation, we focus on the conflict graph model. The extension to conflict hypergraph model is straightforward.

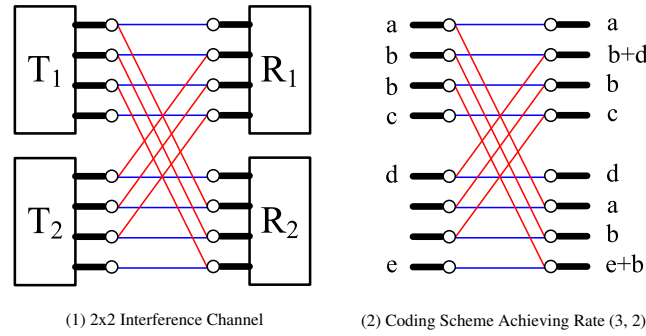


Figure 3.6: [6]. Subfig 1 shows a 2x2 deterministic interference channel with channel gains $\rho_{T_1 R_1} = \rho_{T_2 R_2} = 4$ and $\rho_{T_1 R_2} = \rho_{T_2 R_1} = 3$. Subfig 2 shows a coding scheme achieving the rate tuple (3,2), where T_1 sends a, b, b, c to R_1 and T_2 sends $d, \emptyset, \emptyset, e$ to R_2 . R_1 can decode and obtain a, b, c , while R_2 can decode and obtain d, e .

3.3 NUM Over General Multi-Hop Network: Link-Centric Formulation

In this section, we focus on the link-centric formulation based feasible rate region. Consider a multi-hop network $G=(V, L)$, where each user is associated with a single path. Let H be the set of all independent sets over the corresponding conflict graph G_c . Let $\mathbf{q} = [q_h, h \in H]^T$ be the vector of probability (or time fraction) of all independent sets. Let $\mathbf{x} = [x_s, s \in S]^T$ be the vector of sending rates of users. Let $\lambda_{l,h}$ be the capacity of link l within the independent set h . $\lambda_{l,h} = 0$ means link l is not activated within the independent set h . We also let $\{s : l \in s\}$ denote the set of users that sharing the link l .

Consider the following master utility maximization problem over G_c .

$$\mathbf{MP} : \max_{\mathbf{x} \geq 0, \mathbf{q} \geq 0} \sum_{s \in S} U_s(x_s) \quad (3.2)$$

$$\text{s.t.} \quad \sum_{s: l \in s} x_s \leq \sum_{h \in H} \lambda_{l,h} q_h, \quad \forall l \in L \quad (3.3)$$

$$\sum_{h \in H} q_h = 1.$$

Solving the master problem \mathbf{MP} (3.2) is very challenging because the scheduling subproblem is NP-hard in general. To see that, first, by relaxing

the first set of inequality constraints (3.3) in problem **MP**, we get its partial Lagrangian:

$$L(\mathbf{x}, \mathbf{q}, \mathbf{r}) = \sum_{s \in S} U_s(x_s) - \sum_{l \in L} r_l \left(\sum_{s: l \in s} x_s - \sum_{h \in H} \lambda_{l,h} q_h \right),$$

where $\mathbf{r} = [r_l, l \in L]^T$ is the vector of Lagrange multipliers. We notice that

$$\begin{aligned} \sum_{l \in L} r_l \sum_{s: l \in s} x_s &= \sum_{s \in S} x_s \sum_{l: l \in s} r_l \\ \sum_{l \in L} r_l \sum_{h \in H} \lambda_{l,h} q_h &= \sum_{h \in H} q_h \sum_{l \in L} \lambda_{l,h} r_l. \end{aligned}$$

Since the Slater constraint qualification conditions hold for convex optimization problems with concave objective functions and linear constraints [5], the strong duality holds for problem **MP**. Thus the problem **MP** can be solved by finding the saddle points of $L(\mathbf{x}, \mathbf{q}, \mathbf{r})$ via solving the following problem:

$$\begin{aligned} \mathbf{DP} : \min_{\mathbf{r} \geq 0} & \left(\max_{\mathbf{x} \geq 0} \left(\sum_{s \in S} U_s(x_s) - \sum_{s \in S} x_s \sum_{l: l \in s} r_l \right) \right. \\ & \left. + \max_{\mathbf{q} \geq 0} \left(\sum_{h \in H} q_h \sum_{l \in L} \lambda_{l,h} r_l \right) \right) \\ \text{s.t.} & \sum_{h \in H} q_h = 1. \end{aligned} \quad (3.4)$$

The above problem can be solved successively in $\mathbf{q}, \mathbf{x}, \mathbf{r}$. The key challenge lies in solving the sub-problem in \mathbf{q} , which is the Maximum Weighted Independent Set (MWIS) problem [64]:

$$\begin{aligned} \mathbf{MWIS} : \max_{\mathbf{q} \geq 0} & \sum_{h \in H} q_h \sum_{l \in L} \lambda_{l,h} r_l \\ \text{s.t.} & \sum_{h \in H} q_h = 1. \end{aligned} \quad (3.5)$$

The optimal value of the **MWIS** problem is given by computing the max function: $\max_{h \in H} \sum_{l \in L} \lambda_{l,h} r_l$.

3.3.1 Markov Approximation

The **MWIS** problem is NP-hard [64] and hard to approximate even in a centralized manner [64]. Here we apply the Markov approximation framework [10] to solve the problem in a distributed way. There are two steps of Markov approximation framework [10]: log-sum-exp approximation and distributed construction of Markov chain.

First, we apply the log-sum-exp approximation

$$\max_{h \in H} \sum_{l \in L} \lambda_{l,h} r_l \approx \frac{1}{\beta} \log \left[\sum_{h \in H} \exp \left(\beta \sum_{l \in L} \lambda_{l,h} r_l \right) \right], \quad (3.6)$$

where β is a positive constant.

Let $|H|$ denote the size of the set H , then the approximation accuracy is known as follows:

Proposition 3.5.

$$\max_{h \in H} \sum_{l \in L} \lambda_{l,h} r_l \leq \frac{1}{\beta} \log \left[\sum_{h \in H} \exp \left(\beta \sum_{l \in L} \lambda_{l,h} r_l \right) \right] \leq \max_{h \in H} \sum_{l \in L} \lambda_{l,h} r_l + \frac{1}{\beta} \log |H| \quad (3.7)$$

Proof. Given $\beta > 0$, let

$$\phi(h) \triangleq \sum_{l \in L} \lambda_{l,h} r_l, \forall h \in H,$$

we have

$$\exp(\max_{h \in H} \beta \phi(h)) \leq \sum_{h \in H} \exp[\beta \phi(h)] \leq |H| \exp(\max_{h \in H} \beta \phi(h)).$$

So

$$\max_{h \in H} \beta \phi(h) \leq \log \left(\sum_{h \in H} \exp[\beta \phi(h)] \right) \leq \max_{h \in H} \beta \phi(h) + \log |H|$$

By dividing β in both sides, we obtain the inequality (3.7). \square

As $\beta \rightarrow \infty$, there is no gap between log-sum-exp approximation and the max function. Then we have some important observations in the following proposition.

Proposition 3.6. $\frac{1}{\beta} \log [\sum_{h \in H} \exp(\beta \sum_{l \in L} \lambda_{l,h} r_l)]$ is the optimal value of the following optimization problem

$$\begin{aligned} \mathbf{MWIS} - \beta : \max_{\mathbf{q} \geq 0} & -\frac{1}{\beta} \sum_{h \in H} q_h \log q_h + \sum_{h \in H} q_h \sum_{l \in L} \lambda_{l,h} r_l & (3.8) \\ \text{s.t.} & \sum_{h \in H} q_h = 1. \end{aligned}$$

and the corresponding unique optimal solution is

$$q_h(\beta \mathbf{r}) = \frac{\exp(\beta \sum_{l \in L} \lambda_{l,h} r_l)}{\sum_{h \in H} \exp(\beta \sum_{l \in L} \lambda_{l,h} r_l)}, \forall h \in H. \quad (3.9)$$

Proof. By solving equations of KKT(Karush-Kuhn-Tucker) condition [5] for problem (3.8), we attain the desired results. \square

By time-sharing among different configurations h according to their portions $q_h(\beta \mathbf{r})$, we can solve the problem $\mathbf{MWIS} - \beta$, and hence the problem \mathbf{MWIS} , approximately. Therefore, by the *log-sum-exp* approximation in (3.6), we are implicitly solving an approximated version of the problem \mathbf{MWIS} , off by an *entropy* term $-\frac{1}{\beta} \sum_{h \in H} q_h \log q_h$.

Second, the $q_h(\beta \mathbf{r}), h \in H$ in (3.9) can be interpreted as the stationary distribution of a time reversible Markov chain, whose states are the independent sets in H . Therefore, the second step of Markov approximation framework is to design and implement such a Markov chain in a distributed manner.

3.3.2 Design and Implementation of Markov Chain

To construct a time-reversible Markov chain with its stationary distribution $q_h(\beta \mathbf{r}), h \in H$ in (3.9), we let $h \in H$ be the state of the Markov chain. Denote

$h, h' \in H$ as any two states of Markov chain, and denote $q_{h,h'}$ as the non-negative transition rate from state h to h' . According to Chapter 2, it is sufficient to design $q_{h,h'}$ so that

- the resulting Markov chain is irreducible, i.e., any two states are reachable from each other, either directly or via other states.
- the detailed balance equation is satisfied: for all h and h' in H and $h \neq h'$, $q_h(\beta \mathbf{r}) q_{h,h'} = q_{h'}(\beta \mathbf{r}) q_{h',h}$,

Let link configuration l_k denote the link l under its configuration k , and $\lambda_{l_k,h}$ denote the capacity of link configuration l_k within independent set h . Assume that for link(channel) l with channel gain ρ_l , there are N_l configurations of link(channel) l . Here $1 \leq N_l \leq 2^{N_l}$.

We start by only allowing direct transitions between two “adjacent” states (independent sets) h and h' that differ by one and only one link configuration. Note that doing so will not affect the stationary distribution for time-reversible Markov chains [10, Section II]. Then by this design, the transition from h' to $h = h' \cup \{l_k\}$ corresponds to link configuration l_k starting its transmission. Similarly, the transition from h to h' corresponds to link configuration l_k finishing its on-going transmission.

Now, consider two states h and h' where $h = h' \cup \{l_k\}$. We set $q_{h,h'}$ to $\lambda_{l_k,h}$, and

$$\begin{aligned} q_{h',h} &= \lambda_{l_k,h} \exp\left(\beta\left(\sum_{l \in h} \lambda_{l,h} r_l - \sum_{l \in h'} \lambda_{l,h'} r_l\right)\right) \\ &= \lambda_{l_k,h} \exp\left(\beta \lambda_{l_k,h} r_l\right). \end{aligned}$$

To achieve transition rate $q_{h',h}$, the transmitter of link l sets a timer $T_{l,k}$ for link configuration l_k , which counts down according to an exponential distribution with rate $\lambda_{l_k,h} \exp(\beta \lambda_{l_k,h} r_l)$. When the timer $T_{l,k}$ expires, link l starts to transmit under the configuration k . During the count-down process, if the

transmitter of link l determines that another interfering link configuration is in transmission, link configuration l_k will freeze its count-down process. This could be done in various ways, for instance by the receiver of link l communicating busy/idle notification to the transmitter using a dedicated low-rate feedback channel.

When the transmission of interfering link is over, timer $T_{l,k}$ counts down according to the residual back-off time, which is still exponential distributed with the same rate, because of the memoryless property of exponential distribution.

The transition rate $q_{h,h'}$ can be achieved by the transmitter of link l under configuration k setting its transmission time to follow exponential distribution with rate $\lambda_{l_k,h}$.

The corresponding pseudocode is shown in Algorithm 1. Then we establish the following result:

Proposition 3.7. *Algorithm 1 in fact implements a time-reversible Markov chain with stationary distribution in (3.9).*

The proof is relegated to Appendix 3.7.2

3.3.3 Solving the Approximated Problem by the Primal-Dual Algorithm

By the log-sum-exp approximation, it is not hard to see that we are actually solving a problem close to the original problem **MP**:

$$\begin{aligned} \mathbf{MP} - \beta : \max_{\mathbf{x} \geq 0, \mathbf{q} \geq 0} \quad & \sum_{s \in S} U_s(x_s) - \frac{1}{\beta} \sum_{h \in H} q_h \log q_h & (3.10) \\ \text{s.t.} \quad & \sum_{s: l \in s} x_s \leq \sum_{h \in H} \lambda_{l,h} q_h, \quad \forall l \in L \\ & \sum_{h \in H} q_h = 1. \end{aligned}$$

Denote the optimal solution of master problem **MP** (3.2) by \mathbf{x}^* , and the optimal solution of problem **MP** - β (3.10) by $\hat{\mathbf{x}}$, then $|\sum_{s \in S} (U_s(\hat{x}_s) - U_s(x_s^*))| \leq \frac{\log |H|}{\beta}$. As $\beta \rightarrow \infty$, $\hat{\mathbf{x}}$ approaches \mathbf{x}^* .

Algorithm 1 Implementation of Markov Chain

- 1: The following procedure runs on each individual link(channel) independently. We focus on a particular link(channel) l with the channel gain ρ_l .
 - 2: **procedure** INITIALIZATION
 - 3: Obtains r_l based on queue-length of link(channel) l
 - 4: index $\leftarrow 0$
 - 5: Invoke Procedure Wait-and-Transmit(l)
 - 6: **end procedure**
 - 7: **procedure** WAIT-AND-TRANSMIT(l)
 - 8: generates N_l timers $T_{l,k}$, $k = 1, 2, \dots, N_l$ following exponential distributions with rates $\lambda_{l_k} \exp(\beta \lambda_{l_k} r_l)$ respectively and begin counting down. Each timer k is associated with link configuration l_k , a possible configuration of link l , and λ_{l_k} is the corresponding capacity of link configuration l_k .
 - 9: **while** the timer T_{l,k^*} with $k^* = \arg \min_k T_{l,k}$ does not expire **do**
 - 10: **if** Determines the transmission of interfering links **then**
 - 11: index $\leftarrow 1$
 - 12: break
 - 13: **end if**
 - 14: **end while**
 - 15: Terminates all current countdown processes
 - 16: **if** index = 1 **then**
 - 17: index $\leftarrow 0$
 - 18: Invoke Procedure Wait-and-Transmit(l)
 - 19: **else**
 - 20: Sets the transmit time to follow an exponential distribution with rate $\lambda_{l_{k^*}}$ and transmits
 - 21: **end if**
 - 22: **end procedure**
-

MP – β is equivalent to the following problem:

$$\begin{aligned} \mathbf{DP} - \beta : \min_{\mathbf{r} \geq 0} \max_{\mathbf{x} \geq 0} L_\beta(\mathbf{x}, \mathbf{r}) &= \sum_{s \in S} (U_s(x_s) - x_s \sum_{l:l \in s} r_l) \\ &+ \frac{1}{\beta} \log \left[\sum_{h \in H} \exp(\beta \sum_{l \in L} \lambda_{l,h} r_l) \right] \end{aligned} \quad (3.11)$$

The problem **DP** – β (3.11) can be solved by either a dual algorithm or a primal-dual algorithm. Existing work [31, 50] all focused on dual algorithms. We prefer the primal-dual algorithm because of its fast convergence rate (only one time-scale) and smoothness of changes in parameters.

Define the user rates as $x_s, s \in S$ and the link prices as $r_l, l \in L$, we propose a primal-dual algorithm as follows:

$$\left\{ \begin{array}{l} \dot{x}_s = \alpha_s [U'_s(x_s) - \sum_{l:l \in s} r_l]_{x_s}^+ \\ \forall s \in S \text{ user rates updating} \\ \dot{r}_l = k_l [\sum_{s:l \in s} x_s - \sum_{h \in H} \lambda_{l,h} q_h]_{r_l}^+ \\ \forall l \in L \text{ link prices updating} \end{array} \right. , \quad (3.12)$$

where $k_s (s \in S)$ and $\alpha_s (s \in S)$ are positive constants, and function $[b]_a^+ = \max(0, b)$ if $a \leq 0$ and equals b otherwise. With the time-scale separation assumption that Markov chain converges to its stationary distribution instantaneously compared to the time-scale of adaption of \mathbf{x} and \mathbf{r} , we have the following result:

Theorem 3.8. *The primal-dual algorithm (3.12) is globally asymptotically stable.*

The proof is relegated to Appendix 3.7.3

Since the equilibrium point of primal-dual algorithm (3.12) solves the problem **DP** – β (3.11) exactly, it also solves the problem **MP** – β (3.10) exactly. When $\beta \rightarrow \infty$, the primal-dual algorithm (3.12) solves the master problem **MPN** (3.2) in a distributed way.

In practice, however, we are working on a discrete system instead of the continuous one. The discrete-time primal-dual algorithm is shown as follows:

$$\left\{ \begin{array}{l} x_s(m+1) = [x_s(m) + \alpha_s (U'_s(x_s(m)) - \sum_{l:l \in S} r_l(m))]_+ \\ \forall s \in S \quad \text{user rates updating} \\ r_l(m+1) = [r_l(m) - k_l (\sum_{h \in H} \lambda_{l,h} q_h(m) - \sum_{s:l \in S} x_s(m))]_+ \\ \forall l \in L \quad \text{link prices updating} \end{array} \right. , \quad (3.13)$$

where $[\cdot]_+ \triangleq \max(\cdot, 0)$.

It can be viewed as an approximated version of the continuous-time version. The convergence of this discrete-time primal-dual algorithm in (3.13) can be established by using methods similar to [52]. Note that this primal-dual algorithm is a synchronous distributed algorithm, which assumes that updates at links and nodes are synchronized to occur at times $m = 1, 2, \dots$. In practice, especially in large scale networks, because of large network delays, we prefer asynchronous settings. In particular, the set of update time sequence for any user $s \in S$ (primal) and any link $l \in L$ (dual) are denoted by $T_s = \{T_s(1), T_s(2), \dots\}$ and $T_l = \{T_l(1), T_l(2), \dots\}$ respectively. For any time $m \in T_s$, user s updates its rate according to (3.13). At time $m \notin T_s$, user s keeps its rate fixed. Update behavior of link l is similar. Under this asynchronous setting, we can obtain similar convergence results by adopting methods similar to [52].

3.3.4 Convergence of Stochastic Primal-Dual Algorithm

Without the time-scale separation assumption on Markov chain, the above primal-dual algorithm (3.12) turns to a stochastic primal-dual algorithm, given

as follows:

$$\begin{cases} x_s(m+1) = [x_s(m) + \epsilon(m) (U'_s(x_s(m)) - \sum_{l:l \in \mathcal{E}_s} r_l(m))]_+ \\ \forall s \in S \quad \text{user rates updating} \\ r_l(m+1) = [r_l(m) - \epsilon(m) (\bar{\theta}_l(m) - \sum_{s:l \in \mathcal{E}_s} x_s(m))]_+ \\ \forall l \in L \quad \text{link prices updating} \end{cases}, \quad (3.14)$$

where without loss of generality, we assume both primal and dual updates have the same step size $\epsilon(m)$, $\bar{\theta}_l(m)$ is the average link rate measured by link l within the update interval T_m , and T_m is the time interval between the system updating $(\mathbf{x}(m-1), \mathbf{r}(m-1))$ and $(\mathbf{x}(m), \mathbf{r}(m))$.

We omit convergence result of the stochastic primal-dual algorithm (3.14) under diminishing step sizes and increasing update intervals because we want to emphasis the following convergence result under constant step sizes and constant update intervals.

Theorem 3.9. *Assume that $U'_s(0) < \infty, \forall s \in S$, $\max_{s,m} x_s(m) < \infty$ and $\max_{l,m} r_l(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:*

$$T_m = T_0 > 0 \quad \forall m \quad (3.15)$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \quad (3.16)$$

By running the stochastic primal-dual algorithm (3.14), $(\mathbf{x}(m), \mathbf{r}(m))$ converges with probability 1 to the bounded neighborhood of $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$, i.e., optimal solutions of $\mathbf{DP} - \beta$ (3.11) as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\},$$

where C_1, C_2, C_3 are positive constants.

3.4 NUM Over General Multi-Hop Network: Node-Centric Formulation

In this section, we focus on the node-centric formulation based feasible rate region. There are several differences between the node-centric formulation and the link-centric formulation [49]. First, in the node-centric formulation, each user does *not* use any pre-specified set of paths, while in the link-centric formulation, each user uses a pre-specified set of paths. Second, the scheduling component of the node-centric formulation incorporates the routing functionality, while the scheduling component of the link-centric formulation does *not*. Third, the node-centric formulation does *not* require end-to-end feedback, while the link-centric formulation does (aggregate link prices along the paths).

In spite of these difference, the applications of Markov approximation framework to the node-centric formulation and the link-centric formulation are very similar. So in the following, we state the results only and omit details of all proofs for clarity.

The wireless network is represented as a graph $G = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes, and \mathcal{L} is the set of links. S denotes the user set and $\mathbf{x} = [x_s, s \in S]^T$ denotes the vector of sending rates of users.

Let H be the set of all independent sets over the corresponding conflict graph G_c . Let $\mathbf{q} = [q_h, h \in H]^T$ be the vector of probability (or time fraction) of all independent sets. Let $\lambda_{i,j,h}$ be the capacity of link (i, j) within the independent set h . $\lambda_{i,j,h} = 0$ means link (i, j) is not activated within the independent set h .

Given a user $s \in S$, we let $f_{i,j}^s$ denote the flow rate from source s to destination t_s over link $l = (i, j)$. Let

$$1_A = \begin{cases} 1 & \text{if event A is true} \\ 0 & \text{otherwise} \end{cases} \quad (3.17)$$

Then we consider the following master utility maximization problem:

$$\text{MPN} : \max_{\mathbf{x}, \mathbf{f}, \mathbf{q} \geq 0} \sum_{s \in S} U_s(x_s) \quad (3.18)$$

$$\text{s.t. } x_s \mathbf{1}_{i=s} + \sum_{j:(j,i) \in \mathcal{L}} f_{j,i}^s \leq \sum_{j:(i,j) \in \mathcal{L}} f_{i,j}^s, \quad \forall i \in \mathcal{N} - \{t_s\}, s \in S \quad (3.19)$$

$$\sum_{s \in S} f_{i,j}^s \leq \sum_{h \in H} \lambda_{i,j,h} q_h \quad \forall (i,j) \in \mathcal{L} \quad (3.20)$$

$$\sum_{h \in H} q_h = 1. \quad (3.21)$$

Similar to the previous link-centric formulation, we apply the Markov approximation framework. Then we are actually solving a problem close to the original problem **MPN**:

$$\text{MPN} - \beta : \max_{\mathbf{x}, \mathbf{f}, \mathbf{q} \geq 0} \sum_{s \in S} U_s(x_s) - \frac{1}{\beta} \sum_{h \in H} q_h \log q_h \quad (3.22)$$

$$\text{s.t. } x_s \mathbf{1}_{i=s} + \sum_{j:(j,i) \in \mathcal{L}} f_{j,i}^s \leq \sum_{j:(i,j) \in \mathcal{L}} f_{i,j}^s, \quad \forall i \in \mathcal{N} - \{t_s\}, s \in S \quad (3.23)$$

$$\sum_{s \in S} f_{i,j}^s \leq \sum_{h \in H} \lambda_{i,j,h} q_h \quad \forall (i,j) \in \mathcal{L} \quad (3.24)$$

$$\sum_{h \in H} q_h = 1. \quad (3.25)$$

where β is a positive constant.

Associated the first set of inequality constraints (3.23) with Lagrange multipliers $\mathbf{r} = [r_i^s, s \in S, i \in \mathcal{N}]^T$, where $r_{t_s}^s = 0, \forall s \in S$. By using standard Lagrange dual decomposition method, we can show that solving the problem **MPN** - β (3.22) is equal to finding the saddle point of the following problem

$$\text{DDPN} - \beta : \min_{\mathbf{r} \geq 0} \max_{\mathbf{x} \geq 0} L_\beta(\mathbf{x}, \mathbf{r}) = \sum_{s \in S} [U_s(x_s) - r_s^s x_s] + \frac{1}{\beta} \log \left[\sum_{h \in H} \exp(\beta \sum_{(i,j) \in \mathcal{L}} \lambda_{i,j,h} w_{i,j}) \right] \quad (3.26)$$

where

$$w_{i,j} = \max_{s \in S} [r_i^s - r_j^s]_+, \quad \forall (i,j) \in \mathcal{L}$$

The corresponding (unique) optimal solution of $q_h(\beta\mathbf{r}), h \in H$ is

$$q_h = \frac{\exp(\beta \sum_{(i,j) \in \mathcal{L}} \lambda_{i,j,h} w_{i,j})}{\sum_{h' \in H} \exp(\sum_{(i,j) \in \mathcal{L}} \lambda_{i,j,h'} w_{i,j})}, \forall h \in H \quad (3.27)$$

We explore algorithm design in the following subsections. The $q_h(\beta\mathbf{r}), h \in H$ in (3.27) can be interpreted as the stationary distribution of a time reversible Markov chain, whose states are the independent sets in H . We first discuss how to design and implement such a Markov chain in a distributed manner, then we design primal-dual algorithms to solve the problem **DDPN** – β .

3.4.1 Design and Implementation of Markov Chain

The design method of Markov chain, corresponding pseudocode and related proof are very similar to their counterparts for link-centric formulation in Section 3.3.2. So we omit them for clarity.

3.4.2 Solving the Approximated Problem by the Primal-Dual Algorithm

Since $L_\beta(\mathbf{x}, \mathbf{r})$ is not differential with respect to \mathbf{r} , we need to use the corresponding sub-gradients. In the following, we propose a primal-dual algorithm to solve .

Define the user (source) rates as $x_s, s \in S$ and the node prices as $r_i^s, s \in S$. To solve the problem **DDPN** – β , we propose a primal-dual subgradient based flow control algorithm and associated scheduling policy as follows:

Flow-Control: The primal-dual flow-control algorithm is given as follows:

$$\left\{ \begin{array}{l} \dot{x}_s = \alpha_s [U'_s(x_s) - r_s^s]^+ \\ \forall s \in S \text{ user rates updating} \\ \dot{r}_i^s = k_i^s \left[x_s 1_{i=s} + \sum_{j:(j,i) \in \mathcal{L}} f_{j,i}^s - \sum_{j:(i,j) \in \mathcal{L}} f_{i,j}^s \right]_{r_i^s}^+ \\ \forall i \in \mathcal{N} - \{t_s\}, s \in S, \text{ non-destination node prices updating} \\ \dot{r}_{t_s}^s = r_{t_s}^s = 0, \forall s \in S \text{ destination node prices updating} \end{array} \right., \quad (3.28)$$

where $k_i^s (i \in \mathcal{N} - \{t_s\}, s \in S)$ and $\alpha_s (s \in S)$ are positive constants, and function

$$[b]_a^+ = \begin{cases} \max(0, b) & a \leq 0 \\ b & a > 0 \end{cases}$$

Scheduling: for each link $(i, j) \in \mathcal{L}$, find the user $s_{i,j}^*$ such that

$$s_{i,j}^* = \arg \max_{s \in S} (r_i^s - r_j^s), \quad (3.29)$$

and the link transmission rate is defined as follows:

$$f_{ij}^s = \begin{cases} \sum_{h \in H} \lambda_{i,j,h} q_h & \text{if } s = s_{i,j}^*, \text{ and } w_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.30)$$

When the link (i, j) gets opportunity to transmit, if $w_{i,j} > 0$, it will serve flows of the user $s_{i,j}^*$ with the service rate $\sum_{h \in H} \lambda_{i,j,h} q_h$; otherwise, it will transmit NULL bits.

Note that $w_{i,j}$ represent the maximum differential backlog between nodes and (maximized over all users). The policy thus uses back-pressure in an effort to equalize differential backlog. We emphasize that this scheme does not require knowledge of the arrival rates or channel statistics, and does not use any pre-specified set of routes. The route for each unit of data is found dynamically.

3.4.3 Convergence Properties

We use \mathbf{x}^* and $\hat{\mathbf{x}}$ to denote the optimal user rates of master problem **MPN** (3.18) and problem **MPN** $-\beta$ (3.22) respectively. Then we can see that

$$\sum_{s \in S} U_s(\hat{x}_s) \leq \sum_{s \in S} U_s(x_s^*) \leq \sum_{s \in S} U_s(\hat{x}_s) + \frac{1}{\beta} \log |H| \quad (3.31)$$

As $\beta \rightarrow \infty$, $\hat{\mathbf{x}} \rightarrow \mathbf{x}^*$.

With the time-scale separation assumption that Markov chain converges to its stationary distribution instantaneously compared to the time-scale of adaption of the Markov chain parameters, we have the following result:

Theorem 3.10. *The primal-dual algorithm (3.28) is globally asymptotically stable.*

Without the time-scale separation assumption on CSMA Markov chain, the above primal-dual algorithm (3.28) turns to a stochastic primal-dual algorithm modulated by the underlying Markov chain, shown in the following:

$$\left\{ \begin{array}{l} x_s(m+1) = [x_s(m) + \epsilon(m) (U'_s(x_s(m)) - r_s^s(m))]_+ \\ \forall s \in S \quad \text{user rates updating} \\ r_i^s(m+1) = \left[r_i^s(m) + \epsilon(m) \left(x_s(m) 1_{i=s} + \sum_{j:(j,i) \in \mathcal{L}} \bar{f}_{j,i}^s(m) - \sum_{j:(i,j) \in \mathcal{L}} \bar{f}_{i,j}^s(m) \right) \right]_+ \\ \forall i \in \mathcal{N} - \{t_s\}, s \in S, \quad \text{Non-destination node prices updating} \\ r_{t_s}^s(m+1) = r_{t_s}^s(m) = 0, \forall s \in S \quad \text{destination node prices updating} \end{array} \right. , \quad (3.32)$$

where $\epsilon(m)$ is the step size, $\bar{f}_{i,j}^s(m)$ is the average flow rate of user s over link (i, j) measured within the update interval T_m .

Similar to the link-centric formulation, we have the following convergence result of the stochastic primal-dual algorithm (3.32) under constant step size and constant update interval.

Theorem 3.11. *Assume that $U'_s(0) < \infty, \forall s \in S, \max_{s,m} x_s(m) < \infty$ and $\max_{i,m} r_i^s(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:*

$$T_m = T_0 > 0 \quad \forall m \tag{3.33}$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \tag{3.34}$$

By running the stochastic primal-dual algorithm (3.32), $(\mathbf{x}(m), \mathbf{r}(m))$ converges with probability 1 to the bounded neighborhood of $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$ as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C'_3}{T_0} + \epsilon \frac{(C'_1 + C'_2)}{2}\},$$

where C'_1, C'_2, C'_3 are positive constants.

3.5 Numerical Examples

In this section, we present numerical experiments to illustrate the performance of the primal-dual algorithms.

3.5.1 Link-centric Formulation

We consider a network scenario shown in Fig. 3.7, where user 1 chooses the path $A \rightarrow B \rightarrow C$, user 2 chooses the path $B \rightarrow C \rightarrow G$, user 3 chooses the path $D \rightarrow E \rightarrow F$, and user 4 chooses the path $E \rightarrow F \rightarrow G$. By focusing on proportional-fairness, we choose utility function $U(\cdot) = \log(\cdot + 0.1)$. By running the primal-dual algorithm (3.14) with constant step size $\epsilon = 0.01$, constant update interval $T_0 = 100$ and approximating factor $\beta = 100$, we have the corresponding rates and prices shown in Fig. 3.8 and Fig. 3.9 respectively. We compare the numerical results with theoretically optimal values in Table 3.1.

Advantage of Not Treating Interference As Noise: By treating interference as noise (TIAN), we can only obtain aggregate utilities -0.8310

(maximum in theory) and user rates 1, 0.5, 1, 0.5 (maximum in theory). In contrast, by deterministic channel model and **not** treating interference as noise, we obtain larger aggregate utilities -0.0816(maximum in theory), as well as larger user rates 1.5, 0.5, 1.5, 0.5 (maximum in theory). Further, by running our primal-dual algorithms, we obtain aggregate utilities -0.08136 and user rates 1.5, 0.5, 1.5, 0.5. The numerical results are close to the optimal solutions with **not** treating interference as noise, all within 2.45% of the optimal values. Therefore, numerical results illustrate the convergence and optimality of our joint scheduling and primal-dual flow control algorithms.

Now for three parameters β , T_0 and ϵ , we fix any two of them and study the impacts of the remaining one on user rates and aggregate user utilities. We omit studies on link prices because the observations are very similar. We run the primal-dual algorithm (3.14) for 1000 iterations and corresponding numerical results on β , T_0 and ϵ are shown in Table 3.2, Table 3.3 and Table 3.4 respectively. We can see that

- As β *decreases*, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates. Too small β results in deterioration of performance.
- As T_0 *decreases*, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates. Too small T_0 results in deterioration of performance.
- As ϵ *increases*, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates.

So to reduce the gap between the numerical result and the optimal value, we should increase β , T_0 and decrease ϵ . However, too large T_0 and too small ϵ cause much slower convergence.

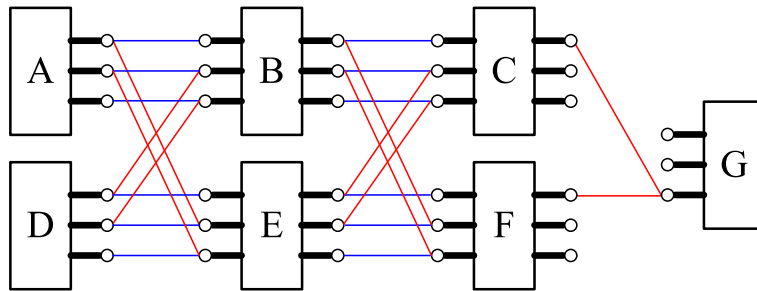


Figure 3.7: A wireless deterministic network with channel gains $\rho_{AB} = \rho_{BC} = \rho_{DE} = \rho_{EF} = 3$, $\rho_{AE} = \rho_{DB} = \rho_{BF} = \rho_{EC} = 2$, and $\rho_{CG} = \rho_{FG} = 1$. Four users are associated with paths $A \rightarrow B \rightarrow C$, $B \rightarrow C \rightarrow G$, $D \rightarrow E \rightarrow F$, and $E \rightarrow F \rightarrow G$ respectively.

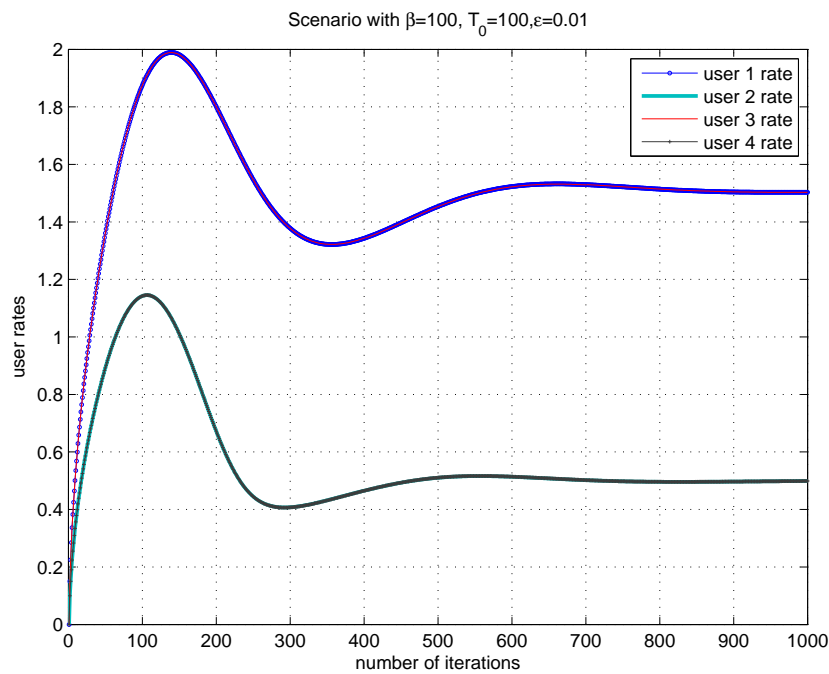


Figure 3.8: Performance of the primal-dual algorithm on user rates with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Initial values of user rates are all θ . Because of symmetricity, not only rate of user 1 and rate of user 3 evolves nearly in the same way, but also rate of user 2 and rate of user 4 evolves nearly in the same way.

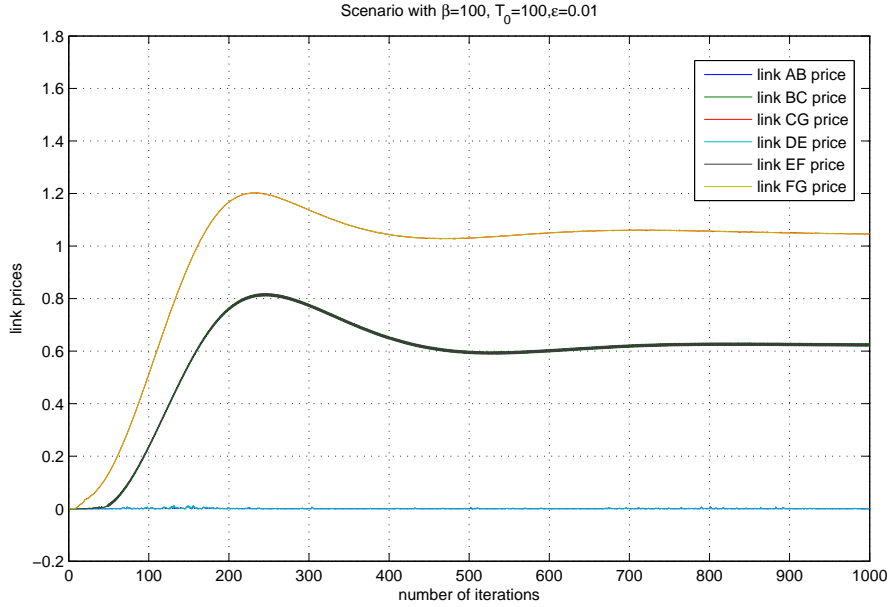


Figure 3.9: Performance of the primal-dual algorithm on link prices with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Initial values of link prices are all 0. Because of symmetricity, price pairs such as price of Link AB and price of Link DE, price of Link BC and price of Link EF, price of Link CG and price of Link FG, evolve nearly in the same way respectively.

	TIAN	Optimal	Approximation	Gap	Relative Error
Sum of Utility	-0.8310	-0.0816	-0.0836	0.0020	2.45%
Rate of User 1	1.0000	1.5000	1.5004	0.0004	0.03%
Rate of User 2	0.5000	0.5000	0.4992	0.0008	0.16%
Rate of User 3	1.0000	1.5000	1.5004	0.0004	0.03%
Rate of User 4	0.5000	0.5000	0.4993	0.0007	0.14%
Price of Link AB	0.0000	0.0000	0.0005	0.0005	
Price of Link BC	0.9091	0.6250	0.6191	0.0059	0.94%
Price of Link CG	0.7576	1.0417	1.0442	0.0025	0.24%
Price of Link DE	0.0000	0.0000	0.0006	0.0006	
Price of Link EF	0.9091	0.6250	0.6292	0.0042	0.67%
Price of Link FG	0.7576	1.0417	1.0434	0.0017	0.16%

Table 3.1: Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.01$. Here 'TIAN' denotes 'Treating Interference As Noise'.

	Optimal	$\beta = 100$	$\beta = 50$	$\beta = 10$	$\beta = 1$
Sum of Utility	-0.0816	-0.0836	-0.0827	-0.0922	-0.5573
Rate of User 1	1.5000	1.5004	1.5016	1.5046	1.6678
Rate of User 2	0.5000	0.4992	0.4990	0.4950	0.3275
Rate of User 3	1.5000	1.5004	1.5017	1.5044	1.6747
Rate of User 4	0.5000	0.4993	0.4991	0.4954	0.3271

Table 3.2: Performance comparison with $T_0 = 100, \epsilon = 0.01$

	Optimal	$T_0 = 100$	$T_0 = 50$	$T_0 = 10$	$T_0 = 1$
Sum of Utility	-0.0816	-0.0836	-0.0783	-0.0551	0.3430
Rate of User 1	1.5000	1.5004	1.5032	1.5117	1.665
Rate of User 2	0.5000	0.4992	0.4999	0.5034	0.5712
Rate of User 3	1.5000	1.5004	1.5031	1.5125	1.6685
Rate of User 4	0.5000	0.4992	0.4998	0.5035	0.5714

Table 3.3: Performance comparison with $\beta = 100, \epsilon = 0.01$

	Optimal	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
Sum of Utility	-0.0816	-0.0836	-0.0781	-0.0766	-0.0755
Rate of User 1	1.5000	1.5004	1.5016	1.5010	1.5030
Rate of User 2	0.5000	0.4992	0.5007	0.5004	0.5042
Rate of User 3	1.5000	1.5004	1.5012	1.5009	1.5001
Rate of User 4	0.5000	0.4992	0.5004	0.5020	0.4984

Table 3.4: Performance comparison with $\beta = 100, T_0 = 100$

3.5.2 Node-centric Formulation

We consider a butterfly network with three users shown in Fig. 3.10. We choose utility function of user 1, 2 and 3 to be $U(\cdot) = \log(\cdot + 0.1)$, $U(\cdot) = \log(\cdot + 0.3)$ and $U(\cdot) = \log(\cdot + 0.5)$ respectively. By running the primal-dual algorithm (3.14) with constant step size $\epsilon = 0.05$, constant update interval $T_0 = 100$ and approximating factor $\beta = 100$, we have the corresponding user rates and node prices shown in Fig. 3.11 and Fig. 3.12 respectively. Focusing on user rates, we compare the numerical results with theoretically optimal values in Table 3.5. The numerical results of all users rates converges to the optimal solutions, all within 2.58% of the optimal values. We omit studies on node prices since the observations are very similar.

Now for three parameters β , T_0 and ϵ , we fix two of them at a time and study the impacts of the remaining one on user rates. We run the primal-dual algorithm (3.14) for 1000 iterations. The numerical results on β , T_0 and ϵ are shown in Table 3.6, Table 3.7 and Table 3.8 respectively. The observations are similar to the counterparts for Link-centric formulation and we omit them for clarity.

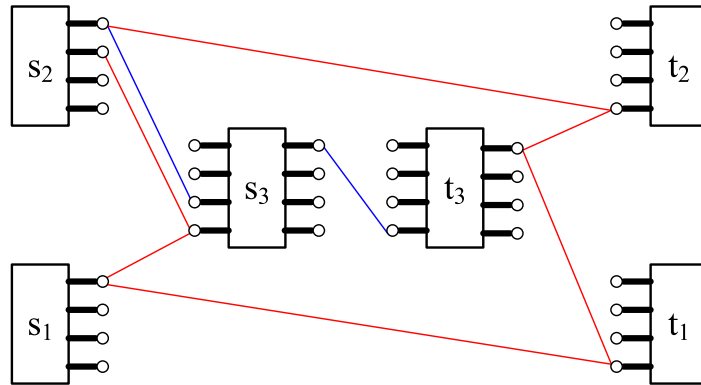


Figure 3.10: A wireless butterfly network with channel gains $\rho_{s_1 s_3} = \rho_{s_1 t_1} = 1$, $\rho_{s_2 s_3} = 2$, $\rho_{s_2 t_2} = 1$, $\rho_{s_3 t_3} = 1$, and $\rho_{t_3 t_2} = \rho_{t_3 t_1} = 1$. There are three users (source-destination pairs): user 1 (s_1, t_1), user 2 (s_2, t_2) and user 3 (s_3, t_3).

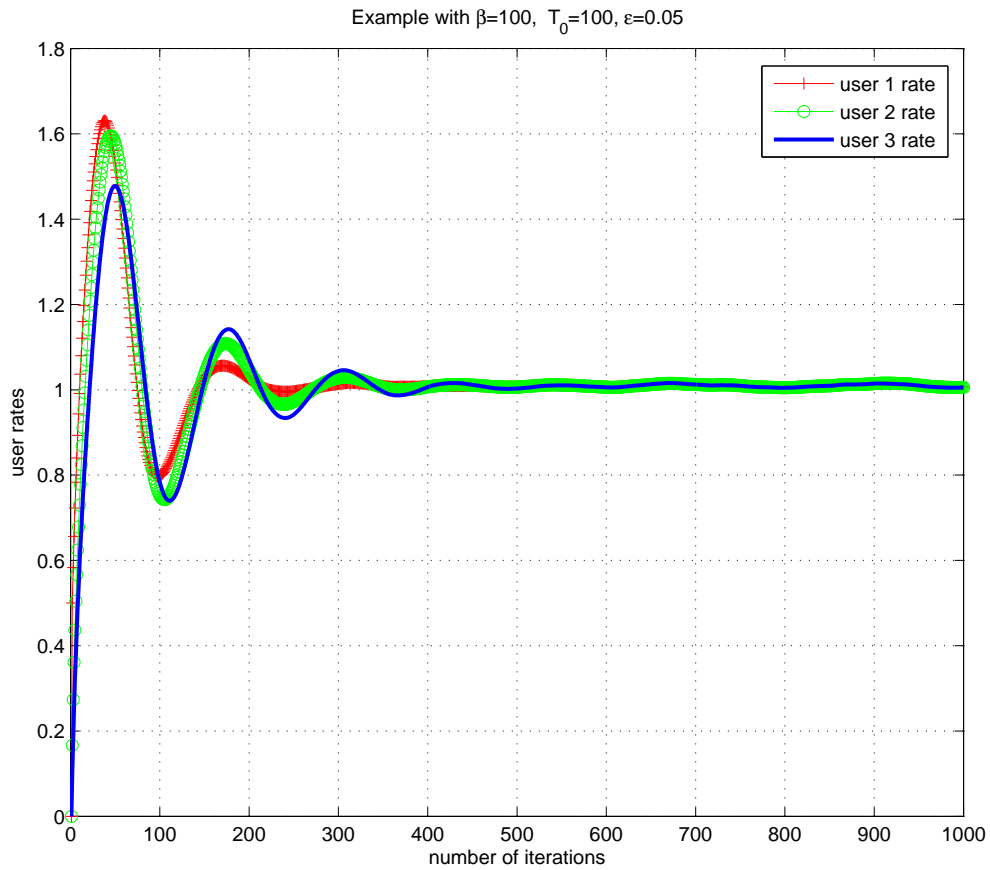


Figure 3.11: Performance of the primal-dual algorithm on user rates with $\beta = 100, T_0 = 100, \epsilon = 0.05$. Initial values of all user rates are θ . All user rates converge to the neighborhood of 1, the optimal value.

	Optimal	Approximation	Gap	Relative Error
Sum of Utility	0.7631	0.7800	0.0197	2.58%
Rate of User 1	1.0000	1.0076	0.0076	0.76%
Rate of User 2	1.0000	1.0070	0.0070	0.70%
Rate of User 3	1.0000	1.0070	0.0070	0.70%

Table 3.5: Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$

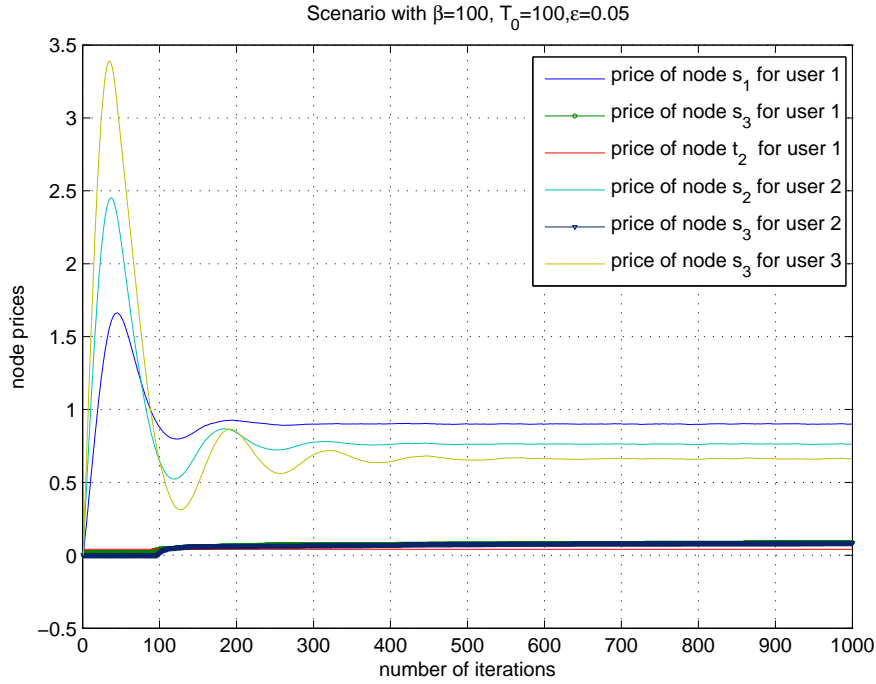


Figure 3.12: Performance of the primal-dual algorithm on node prices with $\beta = 100, T_0 = 100, \epsilon = 0.05$. Initial values of all node prices for user 1, 2, 3 are 0. For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are θ all the time.

	Optimal	$\beta = 100$	$\beta = 50$	$\beta = 10$	$\beta = 1$
Sum of Utility	0.7631	0.7800	0.7880	0.7896	-0.1577
Rate of User 1	1.0000	1.0076	1.0107	1.0131	0.9118
Rate of User 2	1.0000	1.0070	1.0108	1.0133	0.9072
Rate of User 3	1.0000	1.0070	1.0104	1.0068	0.1992

Table 3.6: Performance comparison with $T_0 = 100, \epsilon = 0.05$

	Optimal	$T_0 = 100$	$T_0 = 50$	$T_0 = 10$	$T_0 = 1$
Sum of Utility	0.7631	0.7800	0.8086	0.9510	1.6322
Rate of User 1	1.0000	1.0076	1.0195	1.0826	1.4285
Rate of User 2	1.0000	1.0070	1.0197	1.0823	1.4106
Rate of User 3	1.0000	1.0070	1.0195	1.0833	1.4563

Table 3.7: Performance comparison with $\beta = 100, \epsilon = 0.05$

	Optimal	$\epsilon = 0.05$	$\epsilon = 0.08$	$\epsilon = 0.1$	$\epsilon = 0.2$
Sum of Utility	0.7631	0.7800	0.7869	0.7882	0.7919
Rate of User 1	1.0000	1.0076	1.0103	1.0113	1.0125
Rate of User 2	1.0000	1.0070	1.0107	1.0110	1.0125
Rate of User 3	1.0000	1.0070	1.0094	1.0097	1.0119

Table 3.8: Performance comparison with $\beta = 100, T_0 = 100$

3.6 Conclusions

In this chapter, within the NUM framework, we study the cross-layer optimization problem for wireless networks with deterministic channel models. We present an extended conflict graph model to characterize the feasible rate region. In the important cases such as single-hop multiple access wireless networks, we show that the feasible rate region is equal to the information theoretic capacity region. Then we consider general multi-hop wireless networks with both link-centric formulation and node-centric formulation. We approximately solve the NUM problem in a distributed manner by Markov approximation framework, including the scheduling algorithm and primal-dual flow-control algorithm. The primal-dual algorithm is shown to converge to the optimal solutions of the NUM problem with or without time-scale separation assumption. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant step size and constant update interval. Numerical results illustrate not only the advantage of deterministic channel model over treating interference as noise, but also the convergence and optimality of our joint scheduling and primal-dual flow control algorithms.

3.7 Appendix of Chapter 3

3.7.1 Proof of Proposition 3.4

For the deterministic multiple-access channel, there are N users communicate to a single receiver. For each user $j \in S = \{1, \dots, N\}$, let ρ_j denote channel gain of the channel from the user j to the receiver.

On one hand, let C_{MAC} denote the information-theoretic capacity region for deterministic multiple-access channel, then by [3], we have

$$C_{MAC} = \left\{ \mathbf{x} : \sum_{j \in M} x_j \leq \max_{j \in M} \rho_j \right. \\ \left. \forall M \subseteq \{1, \dots, N\} \right\}$$

Note that this region is characterized by $2^N - 1$ constraints, each corresponding to a nonempty subset of users.

On the other hand, let R_{MAC} denote the conflict graph based rate region for deterministic multiple-access channel. Let L_s be the set of sub-links from source node s to the common receiver. Each sub-link in graph G have unit capacity. Let H be the set of all independent sets over conflict graph G_c . Let $\mathbf{q} = [q_h, h \in H]^T$ be the vector of probability (or time fraction) of all independent sets. Then we have

$$R_{MAC} = \bigcup_{\mathbf{q} \in \mathcal{Q}} R_{\mathbf{q}}$$

where

$$\mathcal{R}(\mathbf{q}) = \left\{ \mathbf{x} : x_s \leq \sum_{l \in L_s} \sum_{h: l \in h} q_h \right. \\ \left. \forall s \in S = \{1, \dots, N\} \right\}$$

and

$$\mathcal{Q} = \left\{ \mathbf{q} : \sum_{h \in H} q_h \leq 1 \right. \\ \left. q_h \geq 0, \forall h \in H \right\}$$

We will show that any set of rates achievable in R_{MAC} is also achievable in C_{MAC} , and *vice versa*.

First, we have $R_{MAC} \subseteq C_{MAC}$ since conflict graph model uses only a subset of possible information-theoretic schemes to achieve the rate region.

Second, we will show that $C_{MAC} \subseteq R_{MAC}$. The capacity region C_{MAC} is a N -dimensional polyhedron, and successive decoding with interference cancellation can achieve all corner points of C_{MAC} [3, 13]. Every decoding order corresponds to a different corner point of C_{MAC} , and consequently, there are $N!$ corner points in the capacity region C_{MAC} . By the convexity of C_{MAC} and R_{MAC} (due to the convex hull operation), it is sufficient to show that all corner points (i.e, the successive decoding points) of C_{MAC} are in the capacity region R_{MAC} .

Given a decoding order $\pi_1, \pi_2, \dots, \pi_N$ in which user π_1 is decoded first, user π_2 is decoded second, \dots , user π_N is decoded last. Without loss of generality, we assume that $\rho_{\pi_1} \leq \rho_{\pi_2} \leq \dots \leq \rho_{\pi_N}$, then the rate vector of the corresponding corner point $\mathbf{x}(\pi)$ is equal to

$$\begin{aligned} x_{\pi_1} &= \rho_{\pi_1} \\ x_{\pi_2} &= \rho_{\pi_2} - \rho_{\pi_1} \\ &\vdots \\ x_{\pi_N} &= \rho_{\pi_N} - \rho_{\pi_{N-1}} \end{aligned}$$

Now we will show $\mathbf{x}(\pi) \in R_{MAC}$. We construct an independent set H_π of the corresponding conflict graph, which includes ρ_{π_1} sub-links of user π_1 , $\rho_2 - \rho_1$ sub-links of user π_2 , \dots , $\rho_{\pi_N} - \rho_{\pi_{N-1}}$ sub-links of user π_N . Sub-links belonging to different users are disjoint. Also we let probability (or time fraction) of this independent set $q_{H_\pi} = 1$, thus the probability of all other independent sets are zero. Then we can see that for any user π_j , $j \in \{1, 2, \dots, N\}$, $x_{\pi_j} = (\rho_{\pi_j} - \rho_{\pi_{j-1}}) q_{H_\pi} = \rho_{\pi_j} - \rho_{\pi_{j-1}}$.

Thus $\mathbf{x}(\pi) \in R_{MAC}$. Similarly, all corner points (i.e, the successive decoding points) of C_{MAC} are in the capacity region R_{MAC} . Therefore, $C_{MAC} \subseteq R_{MAC}$. Combining two directions, we have $C_{MAC} = R_{MAC}$.

3.7.2 Proof of Proposition 3.7

First, let state 0 denote the state where no link is transmitting. Since direct transitions are allowed only between two “adjacent” states that differ by one and only one link configuration, it is not hard to verify that state 0 can reach any other state in a finite number of transitions and vice versa. Therefore, all two states h, h' can reach each other within a finite number of transitions, either by direct transitions (if h and h' are adjacent states), or $h(h')$ reaches 0 first, then starts from 0 to reach $h'(h)$. Therefore, the constructed Markov chain is irreducible. Further, it is a finite state ergodic Markov chain with a unique stationary distribution. We now show that the stationary distribution is indeed (3.9).

According to the Algorithm 1, only one link and only one of its configurations can be activated at a time. Thus direct transitions occur only between two “adjacent” states. Denote this link configuration as l_k , which belongs to $\frac{\rho_l(\rho_l+1)}{2}$ configurations of link l . Now given any two adjacent states h and h' where $|h| = h' \cup \{l_k\}$. Since the timer for link configuration l_k counters down according to an exponential distribution with rate $\lambda_{l_k,h} \exp(\beta \lambda_{l_k,h} r_l)$, we have $q_{h',h} = \lambda_{l_k,h} \exp(\beta \lambda_{l_k,h} r_l)$. On the other hand, transmit time of l_k follows an exponential distribution with rate $\lambda_{l_k,h}$, thus $q_{h,h'} = \lambda_{l_k,h}$.

With (3.9), we have $q_h(\beta \mathbf{r}) \cdot q_{h,h'} = q_{h'}(\beta \mathbf{r}) \cdot q_{h,h}$, i.e., the detailed balance equations hold. For any two non-adjacent states, the detailed balance equations hold trivially. Thus the constructed Markov chain is time-reversible and its stationary distribution is indeed (3.9) according to Theorem 1.3 and Theorem 1.14 in [34].

3.7.3 Proof of Theorem 3.8

Let \hat{x}_s ($s \in S$) and \hat{r}_l ($l \in L$) be the optimal solutions of the problem **DP** – β (3.11). Since the log-sum-exponential function in (3.11) is convex with respect to \mathbf{r} [5], thus $L_\beta(\mathbf{x}, \mathbf{r})$ is concave in \mathbf{x} and convex in \mathbf{r} . By properties of gradients for convex functions [5], we have

$$\begin{aligned} (\hat{\mathbf{x}} - \mathbf{x})^T \cdot \nabla_{\mathbf{x}} L_\beta(\mathbf{x}, \mathbf{r}) &\geq 0 \\ (\hat{\mathbf{r}} - \mathbf{r})^T \cdot \nabla_{\mathbf{r}} L_\beta(\mathbf{x}, \mathbf{r}) &\leq 0 \end{aligned}$$

So

$$\sum_{s \in S} (\hat{x}_s - x_s) \left[U'_s(x_s) - \sum_{l: l \in s} r_l \right] \geq 0 \quad (3.35)$$

$$\sum_{l \in L} (\hat{r}_l - r_l) \left[\sum_{h \in H} \lambda_{l,h} q_h(\beta \mathbf{r}) - \sum_{s: l \in s} x_s \right] \leq 0 \quad (3.36)$$

Consider the following Lyapunov function

$$V(\mathbf{x}, \mathbf{r}) = \sum_{s \in S} \int_{\hat{x}_s}^{x_s} \frac{1}{\alpha_s} (v - \hat{x}_s) dv + \sum_{l \in L} \int_{\hat{r}_l}^{r_l} \frac{1}{k_l} (u - \hat{r}_l) du. \quad (3.37)$$

Then

$$\begin{aligned} \dot{V} &= \frac{dV}{dt} \\ &= \sum_{l \in L} (r_l - \hat{r}_l) \left[\sum_{s: l \in s} x_s - \sum_{h \in H} \lambda_{l,h} q_h(\beta \mathbf{r}) \right]_{r_l}^+ \\ &\quad + \sum_{s \in S} (x_s - \hat{x}_s) \left[U'_s(x_s) - \sum_{l: l \in s} r_l \right]_{x_s}^+ \\ &\leq \sum_{l \in L} (r_l - \hat{r}_l) \left[\sum_{s: l \in s} x_s - \sum_{h \in H} \lambda_{l,h} q_h(\beta \mathbf{r}) \right] \\ &\quad + \sum_{s \in S} (x_s - \hat{x}_s) \left[U'_s(x_s) - \sum_{l: l \in s} r_l \right] \\ &\leq 0 \text{ because of (3.35) and (3.36)} \end{aligned}$$

Further, $\frac{dV}{dt} = 0$ only when $r_l = \hat{r}_l, l \in L$ and $x_s = \hat{x}_s, s \in S$. We also have that the Lyapunov function $V(\mathbf{x}, \mathbf{r})$ is radially unbounded. $V(\hat{\mathbf{x}}, \hat{\mathbf{r}}) = 0$ and $V(\mathbf{x}, \mathbf{r}) > 0, \forall \mathbf{x} > \hat{\mathbf{x}}, \mathbf{r} > \hat{\mathbf{r}},$.

Then the globally asymptotic stability comes from the Lyapunov theorem [36]. Thus the dynamic system defined by the primal-dual algorithm (3.12) converges to the unique optimal solution of the problem $\mathbf{DP} - \beta$ (3.11).

Chapter 4

Wireless Networks With Network Coding

Information is not a commodity.

Network Coding Theory [68]

In this chapter, we apply Markov approximation framework to cross-layer optimization of wireless networks with network coding and broadcast advantage. We focus on two scenarios of network coding: multiple multicast sessions with intra-session network coding and multiple unicast sessions with inter-session network coding. We develop distributed back-pressure scheduling strategies based on primal-dual subgradient algorithms. Further, we show stochastic convergence and asymptotic utility-optimal results under various rules on step sizes and update intervals. Therefore, our distributed schemes provide a systematic way for the cross-layer design of wireless networks with network coding.

4.1 Introduction

Most work on applying the NUM framework for wireless networks is built around the existing routing (non-network-coding) approach, i.e., store and forward. Network coding, introduced in the seminal work [1], allows intermediate

nodes to perform coding operation. Advantages of network coding over routing are shown both theoretically and empirically, especially when the broadcast nature of wireless media is properly exploited [33]. However, for general multi-hop wireless networks with network coding, applying NUM framework is very challenging mainly because of two reasons.

One is the difficulty of characterizing the throughput region of multi-hop wireless networks with network coding. There are two types of network coding: intra-session network coding (where coding is restricted to packets belonging to the same session or connection) and inter-session network coding (where coding is allowed among packets belonging to possibly different sessions). Though polynomial-time algorithms exist for intra-session network coding [27, 39], it is well-known that intra-session network coding is suboptimal [45]. Inter-session network coding is necessary to achieve optimal throughput region. However, performing inter-session network coding is very difficult and linear coding operations are not sufficient for optimal throughput region [18]. Even if we limit ourselves to the simplest form of linear coding, i.e, scalar linear coding, deciding what operations to perform is an NP-hard problem [45].

Therefore, lots of work turns to develop low-complexity methods for inter-session network coding [9, 16, 19, 22, 33, 37, 60, 63] that, though not optimal, achieves significant throughput gains over intra-session network coding.

The other reason is that even if the exact throughput region with either intra-session network coding or suboptimal inter-session network coding is known, the throughput-optimal scheduling on the given throughput region is still a challenging problem. Though back-pressure scheduling algorithm (maximum differential backlog and max-weight scheduling) proposed in the seminal work [62] is shown to be throughput-optimal for multi-hop wireless networks, it is computationally intractable and typically requires centralized computation [49]. Various low-complexity alternatives [8, 32, 49] of back-pressure scheduling algorithm are proposed. In general, they achieve a small fraction of

the capacity region and their signaling overhead result that they are not fully distributed algorithms. Further, in a cross-layer setting, they achieve a small fraction of maximum network utilities [49].

Inspired by recent breakthrough on wireless scheduling [31] and its follow-ups [30, 31, 50, 55, 58, 59], The Markov approximation framework [10] is proposed for design distributed approximation algorithms for general combinatorial optimization problems. In this chapter, guided by Markov approximation framework, we propose distributed back-pressure scheduling algorithms for cross-layer design of wireless networks with network coding and broadcast advantage. The key results are provided as follows

- **Unified scheme for multi-hop wireless networks with network coding:** By modeling the wireless network with broadcast advantages as a hypergraph, we study wireless NUM problems with network coding. All existing work on network coding uses either centralized back-pressure scheduling or distributed maximal scheduling. In contrast, we develop distributed back-pressure strategies to approximately solve the scheduling subproblem and achieve the utility-optimality asymptotically. By doing so, we give a unified distributed scheme for wireless NUM problems with various network coding scenarios, including multicast sessions with intra-session network coding [24] and unicast sessions with session decomposition based suboptimal inter-session network coding approaches [16]. Extensions to wireless NUM with other suboptimal inter-session network coding approaches such as the “poison-remedy flow” [19,22,63] and PINC (pairwise intersession network coding) [37] are pretty straightforward.
- **Convergence of the primal-dual algorithm without time-scale separation assumption:** Existing work [30,31,50] focuses on the dual algorithm. In contrast, we focus on the primal-dual algorithm because

it is smoother than that of the dual algorithm. We prove the convergence of the primal-dual algorithm as follows. Without such time-scale separation assumption, the resulted stochastic primal-dual algorithm is shown to converge to the same optimal solutions with probability one under suitable choices of step sizes and update intervals. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant steps and constant update intervals.

The remainder of this chapter is organized as follows. In Section 4.2, we discuss the related work. In Section 4.3, we introduce preliminaries on system model and the problem formulation. Then in Section 4.4 and Section 4.5, we develop distributed back-pressure scheduling schemes for scenarios of multicast sessions with intra-session network coding, and unicast sessions with suboptimal inter-session network coding respectively. We also show the related convergence and utility-optimal results of primal-dual algorithms. We conclude with a summary and directions for future work in Section 4.6.

4.2 Related Work

Due to rich history of network coding and scheduling, it will be impossible for us to provide a complete history. We will describe a few of these results that are closer to our result. First, we describe results on wireless scheduling algorithms.

Link Scheduling for Wireless Networks. It is well-known that the queue-length based back-pressure scheduling (or called maximum weight scheduling) algorithm [62] is throughput-optimal. However, for general interference models, performing back-pressure scheduling needs centralized operations and is an NP-hard problem. The call for distributed scheduling algorithms with guaranteed throughput has given rise to two main lines of research.

One approach is to adopt special interference models, then apply low-complexity distributed scheduling rules. For example, maximal scheduling [8, 49, 67] and greedy maximal scheduling [32, 44] have been shown to achieve a guaranteed fraction of the maximum throughput region. A few recent work can achieves a throughput region arbitrarily close to the maximum throughput region under certain interference models, e.g., algorithms proposed in [54] under the primary interference model and [20] under two-hop interference model.

Another line of approach develops CSMA-type random access scheduling algorithms for general interference models. [31] developed a distributed algorithm to adaptively choose CSMA parameters to construct a continuous-time Markov chain and solve the maximum weight scheduling problem. It was shown [31, 58] that the adaptive CSMA algorithm achieves maximum throughput region. With this observation, various extensions are proposed, including discrete-time CSMA Markov chain with collisions [55], utility optimization on top of adaptive CSMA [31, 50], and aloha-like algorithms with appropriate function of the queue-size as the weight [59]. Further, [30] adopted techniques from stochastic approximation to prove the convergence, rate stability, and utility-optimality of the algorithms. On the other hand, [50] showed the utility-optimality via the ODE (Ordinary Differential Equation) approach.

Independently, network coding has been extensively studied in both wireline and wireless networks. It can be further classified into two sub-categories: intra-session network coding and inter-session network coding.

Intra-session Network Coding. In [1], the concept of network coding was introduced. Then it was shown that linear network coding [47] is sufficient for achieving the multicast capacity. Further, an algebraic formulation of linear network coding was provided in [39] and polynomial-time linear coding algorithms were found in [27]. In [23], it was shown that distributed random linear network codes can achieve multicast capacity with high probability provided that the field size is large enough.

Distributed random linear network coding was applied for lots of work on practical wireless networks with intra-session network coding. The associated back-pressure algorithms were studied in [24] (with centralized scheduling) and [15] (distributed maximal scheduling) respectively.

Inter-session Network Coding. Inter-session network coding is still in its infancy. It was shown [18] that nonlinear inter-session coding are needed to achieve the capacity region. However, doing such coding is complex and largely open [45]. Actually, even with linear inter-session network coding, determining the capacity is an NP-hard problem [45].

Therefore, low-complexity suboptimal inter-session network coding is preferred. Most work on wireless networks with suboptimal inter-session network coding restricts to multi-session unicast scenarios and focuses on two approaches: search the local butterfly structure and exploit wireless one-hop coding opportunity. For the butterfly-based approach, the achievable rate region was studied in [63] and the associated back-pressure algorithms were studied in [19, 22]. The one-hop coding opportunity approach was proposed in COPE protocol [33]. Then it was followed by numerous work including characterization of throughput region with centralized scheduling [60], energy efficient scheduling with opportunistic coding [16], the power and throughput trade-off between multicast and unicast [9], and binary XOR based pairwise inter-session network coding [37].

4.3 Preliminaries

4.3.1 Network Model

In this paper, we exploit the *broadcast advantage* [15, 60] of wireless networks. Specifically, because of the broadcast nature of wireless medium, a single packet transmission might be overheard by a subset of receiver nodes within range of

the transmitter. Therefore, we model a multi-hop wireless network as a hypergraph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes and \mathcal{L} is the set of hyperlinks. A hyperlink $(i, J) \in \mathcal{L}$ represents a one-hop broadcast transmission, where $i \in \mathcal{N}$ is the transmitter and $J \subseteq \mathcal{N}$ is the set of receivers. Any two hyperlinks either interfere with each other, or they can be activated simultaneously [60].

We adopt the conflict graph model [28,60] to represent this binary interference relationship. The conflict graph $\mathcal{G}_c = (\mathcal{N}_c, \mathcal{L}_c)$ is constructed as follows: for each hyperlink $l \in \mathcal{L}$, there is a corresponding vertex $v_l \in \mathcal{N}_c$. There is an edge between v_l and $v_{l'}$ if the hyperlinks l and $l' \in \mathcal{L}$ interfere with each other. An independent set [28] of the conflict graph \mathcal{G}_c is a set of non-adjacent nodes in \mathcal{G}_c , representing a set of hyperlinks that can be activated simultaneously without interfering each other.

4.3.2 Network Coding

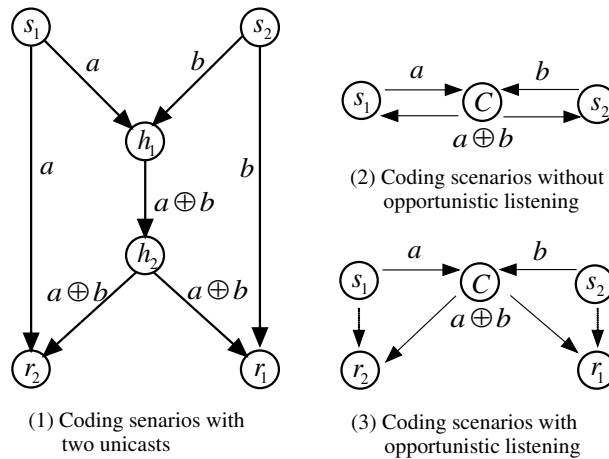


Figure 4.1: Examples of network coding for both wireline and wireless networks.

Figure 4.1 shows the basic scenarios where network coding works. For wireline networks, network coding is demonstrated through the famous butterfly example [1,47] shown in Figure 4.1-(1). The source s_1 wants to deliver packet a to destination r_1 , and source s_2 wants to send packet b to destination r_2 . All

links are assumed to have a capacity of one packet per second. Routing (store and forward) can not achieve the unicast rate pair $(1,1)$ because of bottleneck $h_1 h_2$. However, by network coding, we can achieve the unicast rate pair $(1,1)$ with correspond network code shown in Figure 4.1-(1).

For wireless networks, network coding is demonstrated through the famous cross example [33, 60] shown in Figure 4.1-(2) and Figure 4.1-(3).

In Figure 4.1-(2), source s_1 and s_2 needs to exchange packets a and b via the relay node C . Three transmissions are needed with network coding and wireless broadcast advantage, while four transmissions are needed with routing.

In Figure 4.1-(3), source s_1 and s_2 needs to exchange packets a and b via the relay node C . The dashed arrows $s_1 \dashrightarrow r_2$ and $s_2 \dashrightarrow r_1$ mean that r_2, r_1 are within the transmission range of s_1, s_2 respectively. Because of the wireless broadcast advantage, r_1 and r_2 can perform opportunistic listening. For example, when s_1 sends a to node C , r_2 can overhear the transmission and store the packet a . When r_2 receives the coded packet $a \oplus b$ from C , r_2 can decode it and obtain both packets a and b . Similarly, r_1 obtains both packets a and b . Thus compared to routing approach, using network coding saves one transmission with opportunistic listening and wireless broadcast advantage.

4.4 Intra-session Network Coding

In this Section, we apply the Markov approximation framework to the wireless utility maximization problems with multiple multicast sessions and intra-session network coding.

4.4.1 Wireless NUM

We consider a wireless network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with a set of multicast sessions. To save the heavy notation, we assume that each session is associated with only one source and only one set of destinations. However, extensions to multiple

sources per session is straightforward. The set of sources is denoted by S . Each source node $s \in S \subset \mathcal{N}$ multicasts packets to its destination node set \mathcal{T}_s at a rate of x_s . Let $\mathbf{x} = [x_s, s \in S]^T$ denote the vector of multicast rates of users.

For a single multicast transmission from source s at a rate of x_s , information must flow at a rate x_s to each destination. With intra-session network coding, the actual physical flows on each link need only be the maximum of the individual destination's flows. With this setting, let f_{iJj}^{st} denote the information flow rate from source s to destination node $t \in \mathcal{T}_s$ over hyperlink (i, J) and is intended to node $j \in J$. We also define g_{iJ}^s as the physical flow rate from source s to the set of destination nodes \mathcal{T}_s over hyperlink (i, J) .

Let \mathcal{H} be the set of all independent sets over the corresponding conflict hypergraph \mathcal{G}_c . $\mathbf{p} = [p_h, h \in \mathcal{H}]^T$ is denoted to be the vector of probability (or time fraction) of all independent sets. Let $\mathbb{1}_{\{\cdot\}}$ denote the indicator function, λ_{iJ} denote the capacity of hyperlink (i, J) when it is activated, and λ_{iJ}^h denotes the capacity of hyperlink (i, J) within the independent set h . Then $\lambda_{iJ}^h = \lambda_{iJ} \mathbb{1}_{(i,J) \in h}$.

Then we consider the following master utility maximization problem

$$\mathbf{MP} : \max_{\mathbf{x}, \mathbf{f}, \mathbf{g}, \mathbf{p} \geq 0} \sum_{s \in S} U_s(x_s) \quad (4.1)$$

$$\text{s.t. } x_s \mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} f_{jIi}^{st} \leq \sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} f_{iJj}^{st},$$

$$\forall i \in \mathcal{N} - \{t\}, s \in S, t \in \mathcal{T}_s \quad (4.2)$$

$$\sum_{j \in J} f_{iJj}^{st} \leq g_{iJ}^s, \forall (i, J) \in \mathcal{L}, s \in S, t \in \mathcal{T}_s \quad (4.3)$$

$$\sum_{s \in S} g_{iJ}^s \leq \sum_{h \in \mathcal{H}} p_h \lambda_{iJ}^h, \forall (i, J) \in \mathcal{L} \quad (4.4)$$

$$\sum_{h \in \mathcal{H}} p_h = 1 \quad (4.5)$$

where the constraint (4.2) comes from the property of multi-commodity

flow balance, the constraints (4.3)-(4.5) come from the flow-sharing property of network coding and time-sharing capacity-constraints [24].

Solving the master problem **MP** (4.1) is very challenging because the scheduling subproblem is NP-hard in general. To see that, first, by relaxing the first set of inequality constraints (4.2) in problem **MP**, we get its partial Lagrangian:

$$\begin{aligned}
L(\mathbf{x}, \mathbf{f}, \mathbf{g}, \mathbf{p}, \mathbf{r}) &= \sum_{s \in S} U_s(x_s) + \\
&\sum_{s \in S, t \in \mathcal{T}_s, i \in \mathcal{N} - \{t\}} r_i^{st} \left(\sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} f_{iJj}^{st} - \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} f_{jIi}^{st} - x_s \mathbb{1}_{i=s} \right) \\
&= \sum_{s \in S} U_s(x_s) + \sum_{s \in S, t \in \mathcal{T}_s, i \in \mathcal{N}} r_i^{st} \left(\sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} f_{iJj}^{st} - \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} f_{jIi}^{st} - x_s \mathbb{1}_{i=s} \right)
\end{aligned} \tag{4.6}$$

where $\{r_i^{st}\}$, $s \in S, t \in \mathcal{T}_s, i \in \mathcal{N} - \{t\}$ are Lagrange multipliers and $r_t^{st} = 0$, $\forall s \in S, t \in \mathcal{T}_s$. $\mathbf{r} = [r_i^{st}, s \in S, t \in \mathcal{T}_s, i \in \mathcal{N}]^T$ is the vector.

Note that

$$\sum_{s \in S, t \in \mathcal{T}_s, i \in \mathcal{N}} r_i^{st} x_s \mathbb{1}_{i=s} = \sum_{s \in S} x_s \sum_{t \in \mathcal{T}_s} r_s^{st},$$

and

$$\begin{aligned}
&\sum_{s \in S, t \in \mathcal{T}_s, i \in \mathcal{N}} r_i^{st} \left(\sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} f_{iJj}^{st} - \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} f_{jIi}^{st} \right) \\
&= \sum_{(i,J) \in \mathcal{L}, s \in S} \sum_{t \in \mathcal{T}_s, j \in J} f_{iJj}^{st} (r_i^{st} - r_j^{st}).
\end{aligned} \tag{4.7}$$

Since the Slater constraint qualification conditions are always satisfied for a problem with a concave objective function and linear constraints [5], the strong duality holds for the problem **MP**. Thus it can be solved by finding

the saddle points of $L(\mathbf{x}, \mathbf{f}, \mathbf{g}, \mathbf{p}, \mathbf{r})$ via solving the following problem:

$$\begin{aligned} \mathbf{DDP} : & \min_{\mathbf{r} \geq 0} \left[\max_{\mathbf{x} \geq 0} \left(\sum_{s \in S} U_s(x_s) - \sum_{s \in S} x_s \sum_{t \in \mathcal{T}_s} r_s^{st} \right) + \right. \\ & \left. \max_{\mathbf{f}, \mathbf{g}, \mathbf{p} \geq 0} \left(\sum_{(i, J) \in \mathcal{L}, s \in S} \sum_{t \in \mathcal{T}_s, j \in J} f_{iJj}^{st} (r_i^{st} - r_j^{st}) \right) \right] \quad (4.8) \\ & \text{s.t. (4.3) - (4.5)} \end{aligned}$$

The above problem can be solved successively in $\mathbf{x}, \mathbf{f}, \mathbf{g}, \mathbf{p}, \mathbf{r}$. The key challenge lies in solving the scheduling sub-problem in $\mathbf{f}, \mathbf{g}, \mathbf{p}$ given \mathbf{r} , shown in the following:

$$\begin{aligned} \mathbf{SSP} : & \max_{\mathbf{f}, \mathbf{g}, \mathbf{p} \geq 0} \sum_{(i, J) \in \mathcal{L}, s \in S} \sum_{t \in \mathcal{T}_s, j \in J} f_{iJj}^{st} (r_i^{st} - r_j^{st}) \quad (4.9) \\ & \text{s.t. (4.3) - (4.5)} \end{aligned}$$

Note that \mathbf{SSP} is a linear programming problem on $\mathbf{f}, \mathbf{g}, \mathbf{p}$, one optimal solution is the extreme point solution. Let $[\cdot]_+$ denote the projection onto \mathbb{R}^+ , i.e., $\max(\cdot, 0)$, then after solving \mathbf{SSP} on \mathbf{f}, \mathbf{g} successively, we obtain the subproblem of \mathbf{SSP} on \mathbf{p} in the following:

$$\begin{aligned} \mathbf{SSP} - \mathbf{1} : & \max_{\mathbf{p} \geq 0} \sum_{h \in \mathcal{H}} p_h \sum_{(i, J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ} \quad (4.10) \\ & \text{s.t. } \sum_{h \in \mathcal{H}} p_h = 1. \end{aligned}$$

where

$$w_{iJ} = \max_{s \in S} \sum_{t \in \mathcal{T}_s} \max_{j \in J} [r_i^{st} - r_j^{st}]_+, \forall (i, J) \in \mathcal{L}. \quad (4.11)$$

This is the Maximum Weight Independent Set (MWIS) problem, where the associated weight is $\sum_{(i, J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ}$ for each independent set $h \in \mathcal{H}$.

Therefore the scheduling subproblem is very challenging because the MWIS problem is NP-hard. We utilize the Markov approximation framework to approximately solving this problem in a distributed way.

4.4.2 Markov Approximation

First, we approximate the max function by the log-sum-exp function, i.e.,

$$\max_{h \in \mathcal{H}} \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ} \approx \frac{1}{\beta} \log \left(\sum_{h \in \mathcal{H}} \exp(\beta \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ}) \right). \quad (4.12)$$

Where β is a positive constant. According to Proposition 2.2 in Chapter 2, we are implicitly solving an approximated version of of the problem **SSP – 1** (4.10), off by an entropy term $-\frac{1}{\beta} \sum_{h \in \mathcal{H}} p_h \log p_h$, shown as follows:

$$\begin{aligned} \mathbf{SSP - 1A} : \max_{\mathbf{p} \geq 0} & \sum_{h \in \mathcal{H}} p_h \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ} - \frac{1}{\beta} \sum_{h \in \mathcal{H}} p_h \log p_h \\ \text{s.t.} & \sum_{h \in \mathcal{H}} p_h = 1. \end{aligned}$$

and the corresponding (unique) optimal solution is and the corresponding (unique) optimal solution is

$$p_h = \frac{\exp(\beta \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ})}{\sum_{h' \in \mathcal{H}} \exp(\beta \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^{h'} w_{iJ})}, \forall h \in \mathcal{H} \quad (4.13)$$

Now we are solving a problem close to the original problem **MP** (4.1) as $\beta \rightarrow \infty$:

$$\begin{aligned} \mathbf{MPA} : \max_{\mathbf{x}, \mathbf{f}, \mathbf{g}, \mathbf{p} \geq 0} & \sum_{s \in \mathcal{S}} U_s(x_s) - \frac{1}{\beta} \sum_{h \in \mathcal{H}} p_h \log p_h \\ \text{s.t.} & (4.2) - (4.5) \end{aligned} \quad (4.14)$$

By using standard Lagrange dual decomposition method, we can show that solving the problem **MPA** (4.14) is equal to finding the saddle point of the following problem

$$\begin{aligned} \mathbf{DDPA} : \min_{\mathbf{r} \geq 0} \max_{\mathbf{x} \geq 0} L_\beta(\mathbf{x}, \mathbf{r}) &= \sum_{s \in \mathcal{S}} [U_s(x_s) - x_s \sum_{t \in \mathcal{T}_s} r_s^{st}] \\ &+ \frac{1}{\beta} \log \left[\sum_{h \in \mathcal{H}} \exp(\beta \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ}) \right] \end{aligned} \quad (4.15)$$

We explore algorithm design in the following subsections. The $p_h, h \in \mathcal{H}$ in (4.13) can be interpreted as the stationary distribution of a time reversible Markov chain, whose states are the independent sets in \mathcal{H} . We first discuss how to design and implement such a Markov chain in a distributed manner, then we design primal-dual algorithms to solve the problem **DDPA**.

4.4.3 Design and Implementation of Markov Chain

To construct a time-reversible Markov chain with its stationary distribution $p_h, h \in \mathcal{H}$ in (4.13), we let $h \in \mathcal{H}$ be the state of the Markov chain. Denote $q_{h',h}$ as the transition rate from h' to h , for any two states $h, h' \in \mathcal{H}$. According to Chapter 2, it is sufficient to design $q_{h,h'}$ so that

- the resulting Markov chain is irreducible, i.e., any two states are reachable from each other, either directly or via other states.
- the detailed balance equation is satisfied: for all h and h' in H and $h \neq h'$, $q_h(\beta \mathbf{r}) q_{h,h'} = q_{h'}(\beta \mathbf{r}) q_{h',h}$,

We start by only allowing direct transitions between two “adjacent” states (independent sets) h and h' that differ by one and only one hyperlink. Note that doing so will not affect the stationary distribution for time-reversible Markov chains. By this design, the transition from h' to $h = h' \cup \{(i, J)\}$ corresponds to hyperlink (i, J) starting its transmission. Similarly, the transition from h to h' corresponds to hyperlink (i, J) finishing its on-going transmission.

Now, consider two states h and h' where $h = h' \cup \{(i, J)\}$. Recall that $\lambda_{iJ}^h = \lambda_{iJ} \mathbb{1}_{(i,J) \in h} = \lambda_{iJ}$. We set $q_{h,h'}$ to $\lambda_{iJ}^h = \lambda_{iJ}$, and

$$\begin{aligned} q_{h',h} &= \lambda_{iJ}^h \exp(\beta(\sum_{(i',J') \in \mathcal{L}} \lambda_{i'J'}^h w_{i'J'} - \sum_{(i',J') \in \mathcal{L}} \lambda_{i'J'}^{h'} w_{i'J'})) \\ &= \lambda_{iJ}^h \exp(\beta \lambda_{iJ}^h w_{iJ}) . \\ &= \lambda_{iJ} \exp(\beta \lambda_{iJ} w_{iJ}) . \end{aligned}$$

To achieve transition rate $q_{h',h}$, the node i set a timer for hyperlink (i, J) , which counts down according to an exponential distribution with rate $\lambda_{iJ} \exp(\beta \lambda_{iJ} w_{iJ})$. When the timer expires, hyperlink (i, J) starts to transmit. During the count-down, if the node i senses that another interfering node is in transmission, node i will freeze its count-down process.

When the transmission is over, node i counts down according to the residual back-off time, which is still exponential distributed with the same rate, because of the memoryless property of exponential distribution.

The transition rate $q_{h,h'}$ can be achieved by node i setting its transmission time to follow exponential distribution with rate $\lambda_{i,J}$.

The corresponding pseudocode is shown in Algorithm 2.

Then we establish the following result:

Proposition 4.1. *Algorithm 2 in fact implements a time-reversible Markov chain with stationary distribution in (4.13).*

The proof is relegated to Appendix 4.8.1

4.4.4 Solving the Approximated Problem by the Primal-Dual Algorithm

The problem **DDPA** can be solved by either a dual subgradient algorithm or a primal-dual subgradient algorithm. Existing work [31, 50] focused on the dual subgradient algorithm. We prefer the primal-dual subgradient algorithm because of its smoothness of changes in parameters. Denote the user (source) rates as $x_s, s \in S$ and the node prices as $r_i^{st}, s \in S, t \in \mathcal{T}_s, i \in \mathcal{N}$, we propose a primal-dual congestion control algorithm with scheduling and coding policy as follows:

Algorithm 2 Implementation of Markov Chain

- 1: The following procedures run on each individual hyperlink independently.
We focus on a particular hyperlink (i, J) with the capacity $\lambda_{i,J}$.
 - 2: **procedure** INITIALIZATION((i, J))
 - 3: Obtains $w_{i,J}$ based on back-pressure of hyperlink (i, J)
 - 4: index $\leftarrow 0$
 - 5: Invokes Procedure Wait-and-Transmit((i, J))
 - 6: **end procedure**
 - 7: **procedure** WAIT-AND-TRANSMIT((i, J))
 - 8: generates a timer $T_{i,J}$ following an exponential distribution with rates $\lambda_{i,J} \exp(\beta \lambda_{i,J} w_{i,J})$ respectively and begin counting down.
 - 9: **while** the timer $T_{i,J}$ does not expire **do**
 - 10: **if** Senses the transmission of interfering hyperlinks **then**
 - 11: index $\leftarrow 1$
 - 12: break
 - 13: **end if**
 - 14: **end while**
 - 15: Terminates current countdown process
 - 16: **if** index = 1 **then**
 - 17: index $\leftarrow 0$
 - 18: Invokes Procedure Wait-and-Transmit((i, J))
 - 19: **else**
 - 20: Sets the transmit time to follow an exponential distribution with rate $\lambda_{i,J}$ and transmits
 - 21: **end if**
 - 22: **end procedure**
-

Primal-dual Congestion Control: The primal-dual subgradient algorithm is given as follows:

$$\begin{cases} \dot{x}_s = \alpha_s [U'_s(x_s) - \sum_{t \in \mathcal{T}_s} r_s^{st}]_+^+, \forall s \in S \\ \dot{r}_i^{st} = k_i^{st} [x_s \mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{(j,I) \in \mathcal{L}, i \in I\}} f_{jI}^{st} - \sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} f_{iJj}^{st}]_{r_i^{st}}^+, \forall i \in \mathcal{N} - \{t\}, s \in S, t \in \mathcal{T}_s \\ \dot{r}_t^{st} = r_t^{st} = 0, \forall s \in S, t \in \mathcal{T}_s \end{cases}, \quad (4.16)$$

Where k_i^{st} ($i \in \mathcal{N} - \{t\}, s \in S, t \in \mathcal{T}_s$) and α_s ($s \in S$) are positive constants, and function

$$[b]_a^+ = \begin{cases} \max(0, b) & a \leq 0 \\ b & a > 0 \end{cases}$$

Session(Source)-Scheduling: for each hyperlink $(i, J) \in \mathcal{L}$, one session (source)

$$s_{i,J} = \arg \max_{s \in S} \sum_{t \in \mathcal{T}_s} \max [r_i^{st} - r_j^{st}]_+ \quad (4.17)$$

is chosen and

$$w_{iJ} = \sum_{t \in \mathcal{T}_{s_{i,J}}} \max_{j \in J} [r_i^{s_{i,J}t} - r_j^{s_{i,J}t}]_+ \quad (4.18)$$

Note that w_{iJ} represents the maximum aggregate differential backlog (maximized over all sessions) over hyperlink (i, J) . The distributed back-pressure scheduling algorithm thus uses back-pressure in an effort to equalize aggregate differential backlog. We emphasize that this algorithm does not use any pre-specified set of routes. The route for each unit of data is found dynamically.

Hyperlink-Scheduling: According to Algorithm 2.

Coding: for all hyperlinks $(i, J) \in \mathcal{L}$, physical transmission rate for any session(source) $s \in S$ is defined as follows:

$$g_{iJ}^s = \begin{cases} \sum_{h \in \mathcal{H}} \lambda_{iJ}^h p_h & \text{if } s = s_{i,J}, \text{ and } w_{i,J} > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (4.19)$$

For all hyperlinks $(i, J) \in \mathcal{L}$, let $j_{iJ} = \arg \max_{j \in J} (r_i^{st} - r_j^{st})$. Then the information flow rate over link $(i, J) \in \mathcal{L}$ for any session(source) $s \in S$, destination node $t \in \mathcal{T}_s$, and receiver $j \in J$ is defined as follows:

$$f_{iJ}^{st} = \begin{cases} \sum_{h \in \mathcal{H}} \lambda_{iJ}^h p_h & \text{if } s = s_{i,J}, j = j_{iJ}, \text{ and } r_i^{st} - r_j^{st} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.20)$$

When the hyperlink (i, J) gets opportunity to transmit, if $w_{iJ} > 0$, node i will perform a random linear network coding of packets of session(source) $s_{i,J}$ with destination $t \in \mathcal{T}_{s_{i,J}}$ satisfying $r_i^{s_{i,J}t} - r_j^{s_{i,J}t} > 0$. The random linear network coding means random linear combination with coefficients chosen from a finite field \mathbb{F} with sufficient large field sizes [23]. Then node i will transmit the coded packets over the hyperlink (i, J) (broadcasts to all nodes in J) at a rate of $\sum_{h \in \mathcal{H}} \lambda_{iJ}^h p_h$. Otherwise, $w_{iJ} = 0$, node i will transmit NULL bits.

4.4.5 Convergence Properties of Primal-dual Algorithms

We use \mathbf{x}^* and $\hat{\mathbf{x}}$ to denote the optimal user rates of master problem **MP** (4.1) and problem **MPA** (4.14) respectively. Then we can see that

$$\sum_{s \in S} U_s(\hat{x}_s) \leq \sum_{s \in S} U_s(x_s^*) \leq \sum_{s \in S} U_s(\hat{x}_s) + \frac{1}{\beta} \log |\mathcal{H}| \quad (4.21)$$

As $\beta \rightarrow \infty$, $\hat{\mathbf{x}} \rightarrow \mathbf{x}^*$.

Now given β , we consider the properties of the proposed algorithm (4.16). Observe that there are two time scales, one is the time-scale of the primal-dual subgradient algorithm (4.16), another is the underlying Markov chain. With the *time-scale separation assumption* that CSMA Markov chain converges to its stationary distribution instantaneously, we have the following result:

Theorem 4.2. *The distributed algorithm (4.16) is globally asymptotically stable and converges to $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$, i.e., the optimal solution of the problem **DDPA** (4.15).*

The proof uses LaSalle's invariance principle [36] and the Lyapunov function

$$V(\mathbf{x}, \mathbf{r}) = \sum_{s \in S} \frac{(x_s - \hat{x}_s)^2}{2\alpha_s} + \sum_{s \in S} \sum_{t \in \mathcal{T}_s} \sum_{i \in \mathcal{N}} \frac{(r_i^{st} - \hat{r}_i^{st})^2}{2k_i^{st}} \quad (4.22)$$

We omit the details here because the proof follows a standard routine [61].

Since the equilibrium point of primal-dual algorithm (4.16) solves the problem **DDPA** (4.15) exactly, it also solves the problem **MPA** (4.14) exactly. When $\beta \rightarrow \infty$, the primal-dual algorithm (4.16) solves the master problem **MP** (4.1) in a distributed way.

Without the time-scale separation assumption on CSMA Markov chain, the above primal-dual algorithm (4.16) turns to a stochastic primal-dual algorithm modulated by the underlying Markov chain, shown in the following:

$$\begin{cases} x_s(m+1) = [x_s(m) + \epsilon(m)(U'_s(x_s(m)) - \sum_{t \in \mathcal{T}_s} r_s^{st}(m))]_+, \forall s \in S \\ r_i^{st}(m+1) = [r_i^{st}(m) + \epsilon(m)(x_s(m)\mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} \bar{f}_{jIi}^{st}(m) - \sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} \bar{f}_{iJj}^{st}(m))]_+ \\ \forall i \in \mathcal{N} - \{t\}, s \in S, t \in \mathcal{T}_s \\ r_t^{st}(m+1) = r_t^{st}(m) = 0, \forall s \in S, t \in \mathcal{T}_s \end{cases} \quad (4.23)$$

where $[\cdot]_+ \triangleq \max(\cdot, 0)$, $m \in \mathbb{Z}^+$, \mathbb{Z}^+ denotes the set of all positive integers, $\epsilon(m)$ is the step size within the update interval T_m , $\bar{f}_{iJj}^{st}(m)$, $\forall (i, J)$, s, t, j is the average flow rate measured by hyperlink (i, J) within the update interval T_m , and T_m is the time interval between the system updating $(\mathbf{x}(m-1), \mathbf{r}(m-1))$ and $(\mathbf{x}(m), \mathbf{r}(m))$.

For time-variant step sizes and update intervals, we have the following convergence properties.

Theorem 4.3. *Assume that $U'_s(0) < \infty, \forall s \in S$, $\max_{s,m} x_s(m) < \infty$ and $\max_{i,s,t,m} r_i^{st}(m) < \infty$. Then the stochastic primal-dual algorithm in (4.23) converges asymptotically with probability one to the optimal solution of problem*

$(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ under the following conditions on step sizes and update intervals:

$$\{T_m\} \text{ is non-decreasing with } m \quad (4.24)$$

$$\epsilon(m) > 0 \forall m, \sum_{m=1}^{\infty} \epsilon(m) = \infty, \sum_{m=1}^{\infty} \epsilon^2(m) < \infty \quad (4.25)$$

$$\sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \quad (4.26)$$

Further, the setting $\epsilon(m) = \frac{1}{m}$, $T_m = m$, $m \geq 1$ is one specific choice of step sizes and update intervals satisfying conditions (4.24)-(4.26). This setting depends only on the time index m .

The proof is based on the Theorem 2.4, we omit the details here because we want to emphasize the following convergence result on the constant step size and constant update interval.

Theorem 4.4. Assume that $U'_s(0) < \infty, \forall s \in S$, $\max_{s,m} x_s(m) < \infty$ and $\max_{i,s,t,m} r_i^{st}(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:

$$T_m = T_0 > 0 \quad \forall m \quad (4.27)$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \quad (4.28)$$

By running the stochastic primal-dual algorithm (4.23), $(\mathbf{x}(m), \mathbf{r}(m))$ converges with probability 1 to the bounded neighborhood of $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$ as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C_3}{T_0} + \epsilon \frac{(C_1 + C_2)}{2}\}$$

where C_1, C_2, C_3 are positive constants. Choosing sufficiently large T_0, m , and sufficiently small ϵ , $(\mathbf{x}(m), \mathbf{r}(m))$ can approach $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$ arbitrarily close.

The proof is relegated to Appendix 4.8.2.

4.5 Inter-session Network coding

In this Section, we apply the Markov approximation framework to the wireless NUM problems with suboptimal inter-session network coding. We adopt

one suboptimal inter-session network coding strategy similar to that in [16], i.e., opportunistic unicast with XOR inter-session coding. However, our approach differs from [16] in that we apply distributed back-pressure scheduling for general interference model, while [16] assumes special primal-interference model and uses maximal-scheduling. Further, our approach achieves utility-optimality asymptotically while [16] achieves a small fraction of maximum utility.

Our strategy exploits one-hop coding opportunity where each coded packet is decoded at the immediate next-hop node; and searches the local-butterfly structures by session decomposition approach [16]. This strategy includes the “XOR poison-remedy flow” approach [19, 22, 63] and COPE approach [33] as special cases.

The session decomposition approach decomposes multiple unicast sessions into a superposition of multicast and unicast sessions. Here we provide an example. Observe that in Figure 4.1-(1), source s_1 (s_2) actually multicasts packet a (b) to both r_1 and r_2 . Therefore, the two unicast sessions u_1 and u_2 can be regarded as a multicast session m' with two sources s_1, s_2 and two common destinations r_1 and r_2 . Then the coding scheme in Figure 4.1-(1) becomes the intra-session coding within the multicast session m' . Further, if each link capacity in Figure 4.1-(1) is arbitrary, we can consider that three sessions exist, where the two unicast sessions \hat{u}_1 and \hat{u}_2 share the same sources and destinations as u_1 and u_2 respectively and one multicast session m' . Note that there are no inter-session coding across unicast sessions \hat{u}_1 and \hat{u}_2 . General decomposition method is similar and can be found in [16].

4.5.1 Wireless NUM

We consider a wireless network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with a set of unicast sessions. Each unicast session is associated with one commodity, including one source

and one destination. The set of commodities is denoted by S , where within each commodity $s \in S$, a source s sends packets to its destination t_s (unicast) at a rate of x_s . Let $\mathbf{x} = [x_s, s \in S]^T$ denote the vector of unicast rates.

Then after session decomposition, we have a set of multicast sessions \mathcal{M} and a set of new unicast sessions \mathcal{V} , where $|\mathcal{V}| = |S|$. Any new unicast session $v_s \in \mathcal{V}$ includes only one commodity $s \in S$, while any multicast session $m \in \mathcal{M}$ includes multiple commodities belonging to set S . There are some heuristic methods to exploit one-hop coding opportunity and construct set \mathcal{M} [33, 60]. Here we assume that \mathcal{M} has been constructed and fixed.

For each hyperlink $(i, J) \in \mathcal{L}$, let $f_{iJj}^{v_s}$ denote the flow rate of unicast session v_s over (i, J) and is intended to node $j \in J$; let $f_{iJj}^{m_s}$ denotes the flow rate of multicast session m (including the commodity s) over (i, J) and is intended to node $j \in J$.

Let \mathcal{H} be the set of all independent sets over the corresponding conflict hypergraph \mathcal{G}_c . $\mathbf{p} = [p_h, h \in \mathcal{H}]^T$ is denoted to be the vector of probability (or time fraction) of all independent sets. $\lambda_{i,J}^h$ denotes the capacity of hyperlink (i, J) within the independent set h .

Given the hyperlink $(i, J) \in \mathcal{L}$, let z_{iJ}^m denote the physical rate of multicast session m over (i, J) , and $z_{iJ}^{v_s}$ denote the physical rate of unicast session v_s over (i, J) , $\forall m \in \mathcal{M}, v_s \in \mathcal{V}$.

Let $\mathbb{1}_{\{i,s\}}$ denotes the indicator function and event $i = s$ denote that node i is the source of commodity s , then we consider the following master utility

maximization problem

$$\mathbf{MP2} : \max_{\mathbf{x}, \mathbf{f}, \mathbf{z}, \mathbf{p} \geq 0} \sum_{s \in S} U_s(x_s) \quad (4.29)$$

$$\begin{aligned} \text{s.t. } x_s \mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{(j, I) \in \mathcal{L}, i \in I\}} \left(\sum_{m \in \mathcal{M}} f_{jI}^{ms} + f_{jI}^{v_s} \right) \leq \\ \sum_{\{J | (i, J) \in \mathcal{L}\}} \sum_{j \in J} \left(\sum_{m \in \mathcal{M}} f_{iJ}^{ms} + f_{iJ}^{v_s} \right), \forall i \in \mathcal{N} - \{t_s\}, s \in S \end{aligned} \quad (4.30)$$

$$\sum_{j \in J} f_{iJ}^{v_s} \leq z_{iJ}^{v_s}, \quad \forall (i, J) \in \mathcal{L}, v_s \in \mathcal{V} \quad (4.31)$$

$$\sum_{j \in J} f_{iJ}^{ms} \leq z_{iJ}^m, \quad \forall (i, J) \in \mathcal{L}, s \in S, m \in \mathcal{M} \quad (4.32)$$

$$\sum_{m \in \mathcal{M}} z_{iJ}^m + \sum_{v_s \in \mathcal{V}} z_{iJ}^{v_s} \leq \sum_{h \in \mathcal{H}} p_h \lambda_{iJ}^h, \quad \forall (i, J) \in \mathcal{L} \quad (4.33)$$

$$\sum_{h \in \mathcal{H}} p_h = 1 \quad (4.34)$$

where the constraint (4.30) comes from the property of multi-commodity flow balance, the constraints (4.31)-(4.34) come from the flow-sharing property of network coding and time-sharing capacity-constraints [16].

Solving the master problem **MP2** (4.29) is very challenging because the scheduling subproblem is NP-hard in general. To see that, we relax the first set of inequality constraints (9) in problem (4.30) in problem **MP2** with Lagrange multipliers $\{r_i^s\}, s \in S, i \in \mathcal{N} - \{t_s\}$. $r_{t_s}^s = 0, \forall s \in S$ and $\mathbf{r} = [r_i^s, s \in S, i \in \mathcal{N}]^T$ is the vector. Then similarly to Section 5, by Lagrange dual decomposition method and solving some lineal programming problems, we obtain the following scheduling subproblem:

$$\begin{aligned} \mathbf{SSP} - \mathbf{2} : \max_{\mathbf{p} \geq 0} \sum_{h \in \mathcal{H}} p_h \sum_{(i, J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ} \\ \text{s.t. } \sum_{h \in \mathcal{H}} p_h = 1. \end{aligned} \quad (4.35)$$

where $\forall(i, J) \in \mathcal{L}$,

$$w_{iJ} = \max(w_{iJ}^1, w_{iJ}^2) \quad (4.36)$$

$$w_{iJ}^1 = \max_{v_s \in \mathcal{V}} \max_{j \in J} (r_i^s - r_j^s)_+ \quad (4.37)$$

$$w_{iJ}^2 = \max_{m \in \mathcal{M}} \sum_{s: s \in m} \max_{j \in J} (r_i^s - r_j^s)_+ \quad (4.38)$$

This is also the Maximum Weight Independent Set (MWIS) problem. We utilize the Markov approximation framework to approximately solving this problem in a distributed way.

4.5.2 Markov Approximation

Given a positive constant β , by log-sum-exp approximation it is not hard to show that we are solving a problem close to the original problem **MP2** (4.29) as $\beta \rightarrow \infty$:

$$\mathbf{MP2A} : \max_{\mathbf{x}, \mathbf{f}, \mathbf{z}, \boldsymbol{\theta}, \mathbf{p} \geq 0} \sum_{s \in S} U_s(x_s) - \frac{1}{\beta} \sum_{h \in \mathcal{H}} p_h \log p_h \quad (4.39)$$

$$\text{s.t. (4.30) - (4.34)}$$

By standard Lagrange dual decomposition method, we show that solving the problem **MP2A** (4.39) is equal to finding the saddle point of the following problem

$$\begin{aligned} \mathbf{DDP2A} : \min_{\mathbf{r} \geq 0} \max_{\mathbf{x} \geq 0} L_\beta(\mathbf{x}, \mathbf{r}) &= \sum_{s \in S} [U_s(x_s) - x_s r_s^s] \\ &+ \frac{1}{\beta} \log \left[\sum_{h \in \mathcal{H}} \exp(\beta \sum_{(i, J) \in \mathcal{L}} \lambda_{iJ}^h w_{iJ}) \right] \end{aligned} \quad (4.40)$$

then we solve the problem **DDPA** by a primal-dual subgradient algorithm. Define the user (source) rates as $x_s, s \in S$ and the node prices as $r_i^s, s \in S, i \in \mathcal{N}$, we propose a primal-dual congestion control algorithm with scheduling and coding policy as follows:

Primal-dual Congestion Control: The primal-dual subgradient algorithm is given as follows:

$$\begin{cases} \dot{x}_s = \alpha_s [U'_s(x_s) - r_s^s]_+^+, \forall s \in S \\ \dot{r}_i^s = k_i^s [\text{sub}_i^s]_{r_i^s}^+, \forall i \in \mathcal{N} - \{t_s\}, s \in S \\ \dot{r}_{t_s}^s = r_{t_s}^s = 0, \forall s \in S \end{cases}, \quad (4.41)$$

Where

$$\begin{aligned} \text{sub}_i^s &= x_s \mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} \left(\sum_{m \in \mathcal{M}} f_{jIi}^{ms} + f_{jIi}^{v_s} \right) \\ &- \sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} \left(\sum_{m \in \mathcal{M}} f_{iJj}^{ms} + f_{iJj}^{v_s} \right), \forall i \in \mathcal{N}, s \in S \end{aligned}$$

is subgradient and $k_i^s (i \in \mathcal{N} - \{t_s\}, s \in S)$ and $\alpha_s (s \in S)$ are positive constants.

Hyperlink-Scheduling: Very similar to Algorithm 2, we omit details here for clarity.

Session-Scheduling and Coding: for each hyperlink $(i, J) \in \mathcal{L}$, let

$$v_{i,J} = \arg \max_{v_s \in V} \max_{j \in J} (r_i^s - r_j^s)_+ \quad (4.42)$$

$$m_{i,J} = \arg \max_{m \in \mathcal{M}} \sum_{s: s \in m} \max_{j \in J} (r_i^s - r_j^s)_+ \quad (4.43)$$

When $w_{i,J} = 0$, node i will transmit NULL bits. When $w_{i,J} > 0$, if $w_{i,J} = w_{i,J}^1$, a unicast session $v_{i,J}$ is chosen. Node i broadcasts a packet from commodity $v_{i,J}$ to all receivers $j \in J$ at rate $\sum_{h \in \mathcal{H}} p_h \lambda_{i,J}^h$.

Otherwise, $w_{i,J} = w_{i,J}^2$, a multicast session $m_{i,J}$ is chosen. Node i codes a packet by XOR-ing together packets from all commodities in $m_{i,J}$ (one packet per commodity), then broadcasts the coded packet to all receivers $j \in J$ at rate $\sum_{h \in \mathcal{H}} p_h \lambda_{i,J}^h$.

Opportunistic Listening: for all hyperlinks $(i, J) \in \mathcal{L}$, let

$$j_{i,J}^s = \arg \max_{j \in J} (r_i^s - r_j^s) \quad (4.44)$$

Each node maintains a virtual queue for each commodity. It also maintains a *side information buffer* containing uncoded packets obtained from transmissions or overhearing.

When node i broadcasts a packet from commodity s to all receivers $j \in J$, only the receiver j_{iJ}^s puts the packet into its virtual queue corresponding to commodity s . The other nodes $J - \{j_{iJ}^s\}$ put the packet in their side information buffers (overhearing).

When node i broadcasts a coded packet from a multicast session m to all receivers $j \in J$, for each commodity $s \in m$, only the receiver j_{iJ}^s decodes the packet using overheard packets in its side information buffer and puts the packet into its virtual queue corresponding to commodity s .

4.5.3 Convergence Properties

With the same proof technique adopted in Session 4.4, we can obtain similar convergence results. We state the results only and omit proof details for clarity.

We use \mathbf{x}^* and $\hat{\mathbf{x}}$ to denote the optimal user rates of master problem **MP2** (4.29) and problem **MP2A** (4.39) respectively. Then we can see that

$$\sum_{s \in S} U_s(\hat{x}_s) \leq \sum_{s \in S} U_s(x_s^*) \leq \sum_{s \in S} U_s(\hat{x}_s) + \frac{1}{\beta} \log |H| \quad (4.45)$$

As $\beta \rightarrow \infty$, $\hat{\mathbf{x}} \rightarrow \mathbf{x}^*$.

With the time-scale separation assumption that Markov chain converges to its stationary distribution instantaneously compared to the time-scale of adaption of the Markov chain parameters, we have the following result:

Theorem 4.5. *The primal-dual algorithm (4.41) is globally asymptotically stable.*

Without the time-scale separation assumption on CSMA Markov chain, the above primal-dual algorithm (4.41) turns to a stochastic primal-dual algorithm

modulated by the underlying Markov chain, shown in the following:

$$\left\{ \begin{array}{l} x_s(m+1) = [x_s(m) + \epsilon(m) (U'_s(x_s(m)) - r_s^s(m))]_+ \\ \forall s \in S \quad \text{user rates updating} \\ r_i^s(m+1) = [r_i^s(m) + \epsilon(m) (In(m) - Out(m))]_+ \\ \forall i \in \mathcal{N} - \{t_s\}, s \in S, \quad \text{Node prices updating} \\ r_{t_s}^s(m+1) = r_{t_s}^s(m) = 0, \forall s \in S \end{array} \right. \quad (4.46)$$

where

$$In(m) = x_s(m) \mathbb{1}_{i=s} + \sum_{j \in \mathcal{N}} \sum_{\{i|(j,I) \in \mathcal{L}, i \in I\}} \left(\sum_{m \in \mathcal{M}} \bar{f}_{jIi}^{ms} + \bar{f}_{jIi}^{v_s} \right),$$

$$Out(m) = \sum_{\{J|(i,J) \in \mathcal{L}\}} \sum_{j \in J} \left(\sum_{m \in \mathcal{M}} \bar{f}_{iJj}^{ms} + \bar{f}_{iJj}^{v_s} \right),$$

$\epsilon(m)$ is the step size, $\bar{f}_{iJj}^{ms}(m)$ and $\bar{f}_{iJj}^{v_s}(m)$ are the average flow rates of user s over hyperlink (i, J) measured within the update interval T_m .

Then we have the following results. Since proof of them are very similar to the proof of Theorem 4.3 and 4.4, we omit details here for clarity.

Theorem 4.6. *Assume that $U'_s(0) < \infty, \forall s \in S, \max_{s,m} x_s(m) < \infty$ and $\max_{i,s,m} r_i^s(m) < \infty$. Then the stochastic primal-dual algorithm in (4.46) converges asymptotically with probability one to the optimal solution of problem $(\hat{\mathbf{x}}, \hat{\mathbf{r}})$ under the following conditions on step sizes and update intervals:*

$$\{T_m\} \text{ is non-decreasing with } m \quad (4.47)$$

$$\epsilon(m) > 0 \forall m, \sum_{m=1}^{\infty} \epsilon(m) = \infty, \sum_{m=1}^{\infty} \epsilon^2(m) < \infty \quad (4.48)$$

$$\sum_{m=1}^{\infty} \frac{\epsilon(m)}{T_m} < \infty \quad (4.49)$$

Further, the setting $\epsilon(m) = \frac{1}{m}, T_m = m, m \geq 1$ is one specific choice of step sizes and update intervals satisfying conditions (4.47)-(4.49).

Theorem 4.7. *Assume that $U'_s(0) < \infty, \forall s \in S$, $\max_{s,m} x_s(m) < \infty$ and $\max_{i,s,m} r_i^s(m) < \infty$. If the sequence of step size $\{\epsilon(m)\}$ and the sequence of update interval $\{T_m\}$ satisfy the following conditions:*

$$T_m = T_0 > 0 \quad \forall m \quad (4.50)$$

$$\epsilon(m) = \epsilon > 0 \quad \forall m \quad (4.51)$$

By running the stochastic primal-dual algorithm (4.46), $(\mathbf{x}(m), \mathbf{r}(m))$ converges with probability 1 to the bounded neighborhood of $\hat{\mathbf{x}}$ and $\hat{\mathbf{r}}$ as follows:

$$\{(\mathbf{x}, \mathbf{r}) : |L_\beta(\mathbf{x}, \mathbf{r}) - L_\beta(\hat{\mathbf{x}}, \hat{\mathbf{r}})| \leq \frac{C'_3}{T_0} + \epsilon \frac{(C'_1 + C'_2)}{2}\},$$

where C'_1, C'_2, C'_3 are positive constants.

Remark. In a pretty straightforward way, our primal-dual subgradient based distributed scheme can be applied to wireless NUM with other suboptimal inter-session coding schemes as well, including the “XOR poison-remedy flow” approach [19, 22, 63] and binary XOR based PINC (pairwise intersession network coding) scheme [37].

4.6 Numerical Examples

In this section, we present numerical experiments to illustrate the performance of the primal-dual algorithms.

4.6.1 Multicast Sessions With Intra-session Network Coding

We consider a wireless butterfly network shown in Fig. 4.2, where two multicast sessions exist: session 1 includes source A multicasting to destinations D and E, and session 2 includes source B multicasting to destinations D and E. We choose utility function of session 1 and 2 to be $U(\cdot) = \log(\cdot + 0.1)$ and

$U(\cdot) = \log(\cdot + 0.3)$ respectively. By running the primal-dual algorithm (4.23) with constant step size $\epsilon = 0.05$, constant update interval $T_0 = 100$ and approximating factor $\beta = 100$, we have the corresponding session rates and node prices shown in Fig. 4.3 and Fig. 4.4 respectively. Focusing on session rates, we compare the numerical results with theoretically optimal values in Table 4.1. The numerical results of all users rates converges to the optimal solutions, all within 4.07% of the optimal values. We omit studies on node prices since the observations are very similar. Therefore, numerical results illustrate the convergence and optimality of our joint scheduling and primal-dual flow control algorithms.

Now for three parameters β , T_0 and ϵ , we fix any two of them and study the impacts of the remaining one on session rates and aggregate utilities. We omit studies on node prices because the observations are very similar. We run the primal-dual algorithm (4.23) for 1000 iterations and corresponding numerical results on β , T_0 and ϵ are shown in Table 4.2, Table 4.3 and Table 4.4 respectively. We can see that

- As β decreases, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates. Too small β results in deterioration of performance.
- As T_0 decreases, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates. Too small T_0 results in deterioration of performance.
- As ϵ increases, the gap between the numerical result and the optimal value is *increasing*, though not monotonically increasing for all flow rates.

So to reduce the gap between the numerical result and the optimal value, we should increase β , T_0 and decrease ϵ . However, too large T_0 and too small ϵ cause much slower convergence.

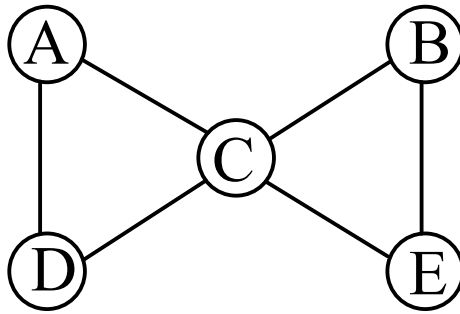


Figure 4.2: A wireless butterfly network with three hyperlinks, $(A, \{C, D\})$, $(B, \{C, E\})$, and $(C, \{D, E\})$. All hyperlinks have one unit capacity. There are two multicast sessions, session 1 includes source A multicasting to destinations D and E, and session 2 includes source B multicasting to destinations D and E.

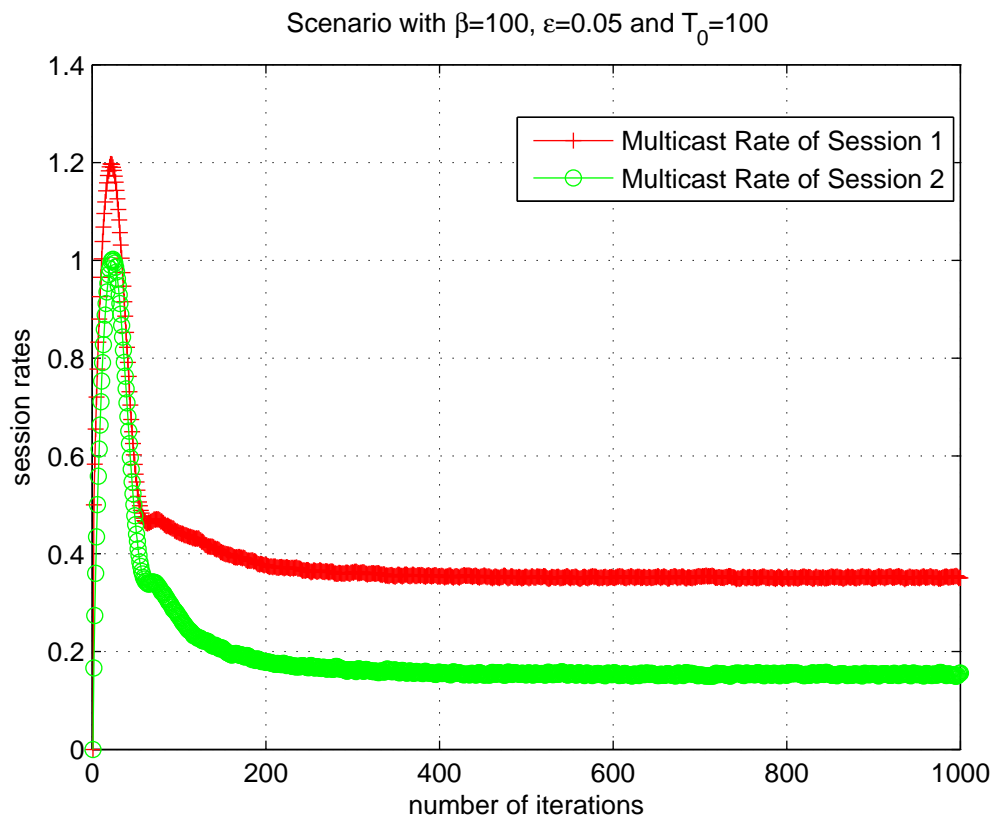


Figure 4.3: Performance of the primal-dual algorithm on multicast session rates with $\beta = 100$, $\epsilon = 0.05$, $T_0 = 100$. Initial values of all session rates are θ .

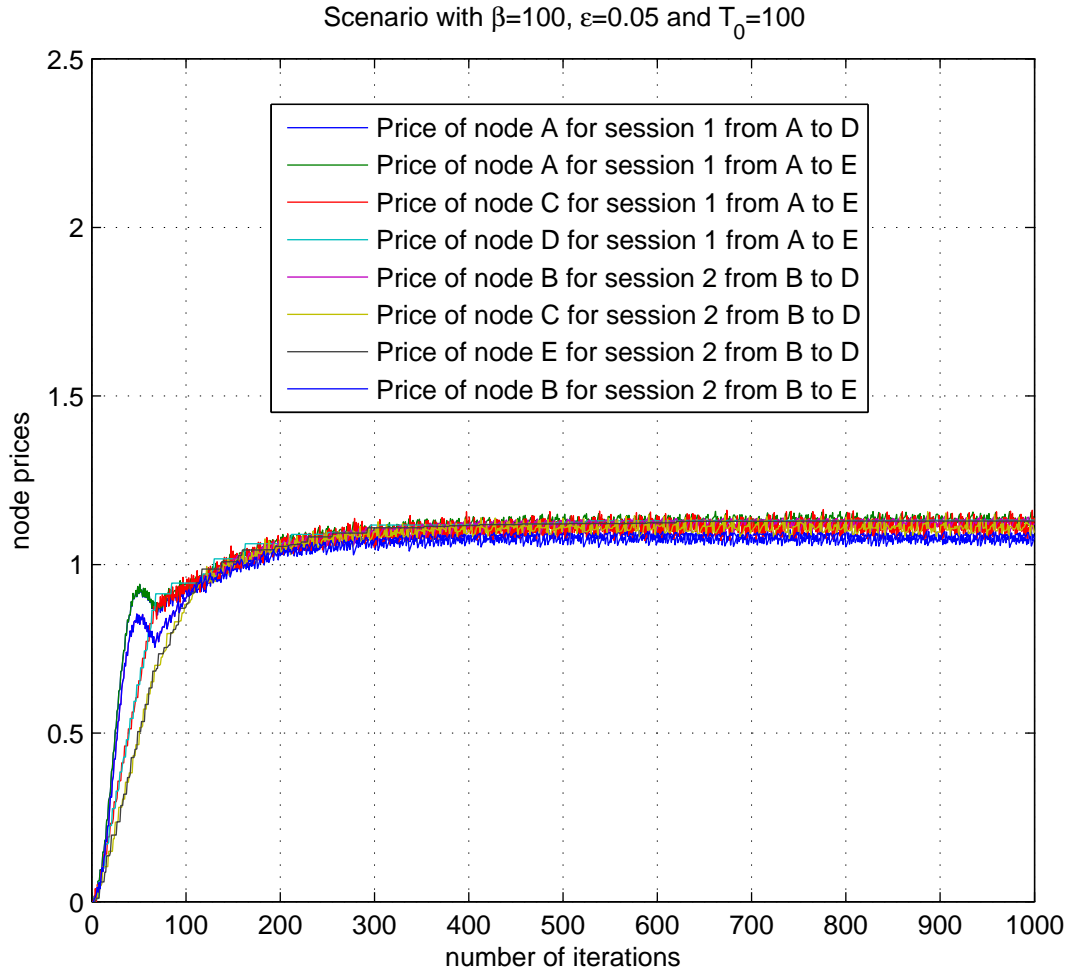


Figure 4.4: Performance of the primal-dual algorithm on node prices with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all node prices are θ . For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are θ all the time. Note that all node prices are stable despite some oscillations.

	Optimal	Approximation	Gap	Relative Error
Sum of Utility	-1.5970	-1.5822	0.0148	0.93%
Rate of Session 1	0.3500	0.3506	0.0006	0.17%
Rate of Session 2	0.1500	0.1561	0.0061	4.07%

Table 4.1: Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$

	Optimal	$\beta = 100$	$\beta = 50$	$\beta = 10$
Sum of Utility	-1.5970	-1.5822	-1.5858	-1.5827
Rate of Session 1	0.3500	0.3506	0.3518	0.3452
Rate of Session 2	0.1500	0.1561	0.1533	0.1614

Table 4.2: Performance comparison with $T_0 = 100, \epsilon = 0.05$

	Optimal	$T_0 = 100$	$T_0 = 50$	$T_0 = 10$
Sum of Utility	-1.5970	-1.5822	-1.5719	-1.4800
Rate of Session 1	0.3500	0.3506	0.3559	0.3732
Rate of Session 2	0.1500	0.1561	0.1555	0.1772

Table 4.3: Performance comparison with $\beta = 100, \epsilon = 0.05$

4.6.2 Unicast Sessions With Inter-session Network Coding

We consider a wireless butterfly network shown in Fig. 4.5, where two unicast sessions exist: session 1 includes source A and destination E, and session 2 includes source B and destination D. We choose utility function of session 1 and 2 to be $U(\cdot) = \log(\cdot + 0.1)$ and $U(\cdot) = \log(\cdot + 0.3)$ respectively. By running the primal-dual algorithm (4.23) with constant step size $\epsilon = 0.05$, constant update interval $T_0 = 100$ and approximating factor $\beta = 100$, we have the corresponding session rates and node prices shown in Fig. 4.6 and Fig. 4.7 respectively.

Focusing on user rates, we compare the numerical results with theoretically optimal values in Table 4.5. The numerical results of all users rates converges to the optimal solutions, all within 2.43% of the optimal values. We omit studies on node prices since the observations are very similar.

Now for three parameters β , T_0 and ϵ , we fix two of them at a time and

	Optimal	$\epsilon = 0.05$	$\epsilon = 0.08$	$\epsilon = 0.1$
Sum of Utility	-1.5970	-1.5822	-1.5916	-1.6060
Rate of Session 1	0.3500	0.3506	0.3515	0.3397
Rate of Session 2	0.1500	0.1561	0.1510	0.1565

Table 4.4: Performance comparison with $\beta = 100, T_0 = 100$

study the impacts of the remaining one on user rates. We run the primal-dual algorithm (4.23) for 1000 iterations. The numerical results on β , T_0 and ϵ are shown in Table 4.6, Table 4.7 and Table 4.8 respectively. The observations are similar to the counterparts for intra-session network coding and we omit them for clarity.

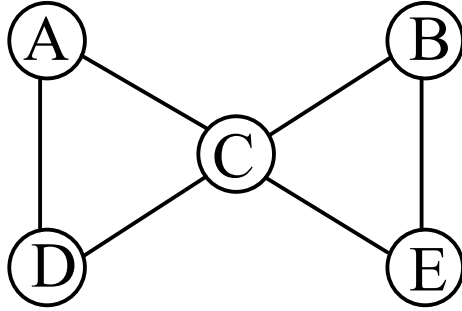


Figure 4.5: A wireless butterfly network with three hyperlinks, $(A, \{C, D\})$, $(B, \{C, E\})$, and $(C, \{D, E\})$. All hyperlinks have one unit capacity. There are two unicast sessions, session 1 includes source A unicasting to destination E, and session 2 includes source B unicasting to destination D.

	Optimal	Approximation	Gap	Relative Error
Sum of Utility	-1.2930	-1.2785	0.0161	1.12%
Rate of Session 1	0.3333	0.3381	0.0081	2.43%
Rate of Session 2	0.3333	0.3355	0.0055	1.65%

Table 4.5: Performance comparison with $\beta = 100, T_0 = 100, \epsilon = 0.05$

	Optimal	$\beta = 100$	$\beta = 50$	$\beta = 10$
Sum of Utility	-1.2930	-1.2785	-1.2765	-1.2799
Rate of Session 1	0.3333	0.3381	0.3384	0.3366
Rate of Session 2	0.3333	0.3355	0.3364	0.3369

Table 4.6: Performance comparison with $T_0 = 100, \epsilon = 0.05$

4.7 Conclusions

In this chapter, we develop a unified distributed scheme for wireless NUM problems with network coding and broadcast advantage, including scenarios

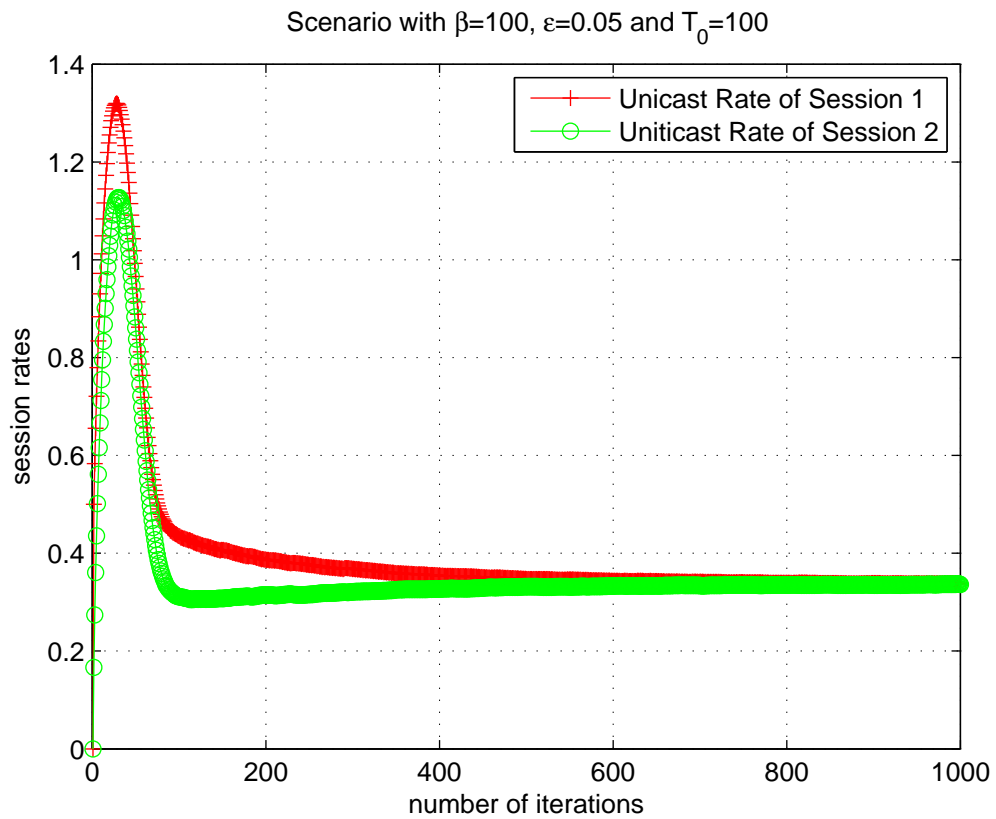


Figure 4.6: Performance of the primal-dual algorithm on multicast session rates with $\beta = 100$, $\epsilon = 0.05$, $T_0 = 100$. Initial values of all session rates are 0.

	Optimal	$T_0 = 100$	$T_0 = 50$	$T_0 = 10$
Sum of Utility	-1.2930	-1.2785	-1.2629	-1.1671
Rate of Session 1	0.3333	0.3381	0.3413	0.3678
Rate of Session 2	0.3333	0.3355	0.3409	0.3654

Table 4.7: Performance comparison with $\beta = 100$, $\epsilon = 0.05$

	Optimal	$\epsilon = 0.05$	$\epsilon = 0.08$	$\epsilon = 0.1$
Sum of Utility	-1.2930	-1.2785	-1.2772	-1.2749
Rate of Session 1	0.3333	0.3381	0.3368	0.3373
Rate of Session 2	0.3333	0.3355	0.3383	0.3391

Table 4.8: Performance comparison with $\beta = 100$, $T_0 = 100$

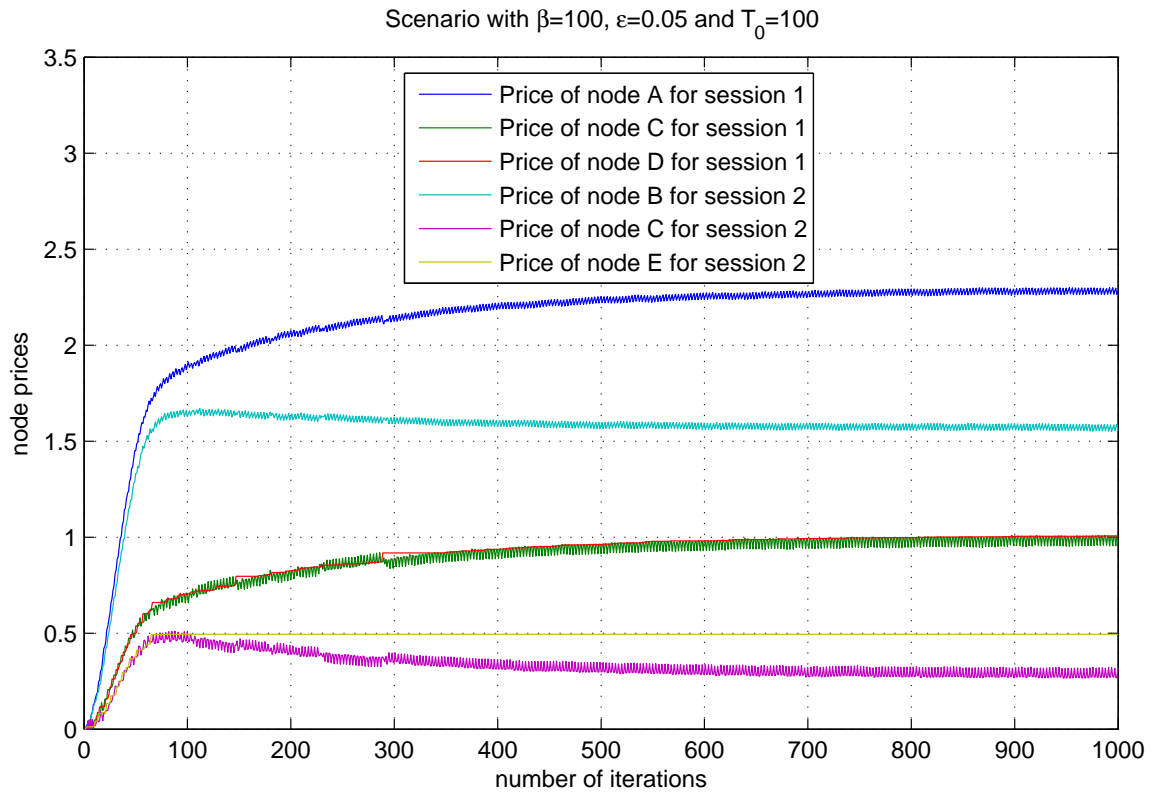


Figure 4.7: Performance of the primal-dual algorithm on node prices with $\beta = 100, \epsilon = 0.05, T_0 = 100$. Initial values of all node prices are θ . For convenience, we only show non-zero node prices. So all other node prices not shown in this picture are θ all the time. Note that all node prices are stable despite some oscillations.

of multicast sessions with intra-session network coding and unicast sessions with suboptimal inter-session network coding. For intra-session network coding scenario, we adopt the random linear network coding algorithm; while for inter-session network coding algorithm, we adopt some sub-optimal coding scheme. We approximately solve the wireless NUM problem in a distributed manner by Markov approximation framework, including the scheduling algorithm, primal-dual flow-control algorithm, and network coding algorithm. The primal-dual algorithm is shown to converge to the optimal solutions of the NUM problem with or without time-scale separation assumption. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant step size and constant update interval. Numerical results validates the analytical results. Actually the proposed distributed scheme can be applied to other inter-session network coding scenarios with variants of sub-optimal coding schemes.

4.8 Appendix of Chapter 4

4.8.1 Proof of Proposition 4.1

First, let state 0 denote the state where no hyperlink is transmitting. Since direct transitions are allowed only between two “adjacent” states that differ by one and only one hyperlink, it is not hard to verify that state 0 can reach any other state in a finite number of transitions and vice versa. Therefore, all two states h, h' can reach each other within a finite number of transitions, either by direct transitions (if h and h' are adjacent states), or $h(h')$ reaches 0 first, then starts from 0 to reach $h'(h)$. Therefore, the constructed Markov chain is irreducible. Further, it is a finite state ergodic Markov chain with a unique stationary distribution. We now show that the stationary distribution is indeed (4.13).

According to the Algorithm 2, only one hyperlink can be activated at a time. Thus direct transitions occur only between two “adjacent” states. Denote this link as (i, J) and its capacity as λ_{iJ} . Then (i, J) counts down with rate $\lambda_{iJ} \exp(\beta \lambda_{iJ} w_{iJ})$. So for any two adjacent states h and h' where $h = h' \cup \{(i, J)\}$, we have

$$q_{h',h} = \lambda_{iJ} \exp(\beta \lambda_{iJ} w_{iJ}).$$

On the other hand, transmission time of (i, J) follows an exponential distribution with rate λ_{iJ} , thus $q_{h,h'} = \lambda_{iJ}$.

With (4.13), we have $p_h \cdot q_{h,h'} = p_{h'} \cdot q_{h',h}$, i.e., the detailed balance equations hold. For any two non-adjacent states, the detailed balance equations hold trivially. Thus the constructed Markov chain is time-reversible and its stationary distribution is indeed (4.13) according to Theorem 1.3 and Theorem 1.14 in [34].

4.8.2 Proof of Theorem 4.4

This proof follows by applying Theorem 2.5 in chapter 2.3.1. To apply Theorem 2.5 in chapter 2.3.1, we need to check whether condition (2.27) is satisfied, i.e.,

$$|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \leq \frac{C_3}{T_0}, \quad \forall m$$

Actually we will show it is indeed true.

In the following, we consider the period m , i.e., from t_m to t_{m+1} . At time t_m with the transmission aggressiveness vector $\mathbf{r}(m)$, denote the corresponding Markov chain by $Y(t)$. $Y(t)$ is a continuous time Markov chain.

By (4.13), the stationary distribution of state $\mathbf{y} \in \mathcal{Y}$ is

$$\pi_{\mathbf{y}}(\mathbf{r}(m)) = \frac{\exp(\beta \sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^{\mathbf{y}} w_{iJ})}{C(\mathbf{r}(m))}, \quad \forall \mathbf{y}. \quad (4.52)$$

Where $C(\mathbf{r}(m)) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(\sum_{(i,J) \in \mathcal{L}} \lambda_{iJ}^{\mathbf{y}'} w_{iJ})$.

Since $r_i^{st} \leq r_{\max}, \forall i, s, t$, then $\forall (i, J) \in \mathcal{L}$,

$$\begin{aligned} 0 \leq w_{iJ} &= \max_{s \in S} \sum_{t \in \mathcal{T}_s} \max_{j \in J} [r_i^{st} - r_j^{st}]_+ \leq \max_{s \in S} \sum_{t \in \mathcal{T}_s} r_i^{st} \\ &\leq \sum_{t \in \mathcal{T}_s} r_{\max} \leq K \cdot r_{\max} \end{aligned} \quad (4.53)$$

where $K \triangleq \max_{s \in S} |\mathcal{T}_s|$, the maximum size of sets of destination nodes. Since the number of states $|\mathcal{Y}| \leq 2^{|\mathcal{L}|}$, and let $\lambda_{i,J}^y, \forall (i, J) \in \mathcal{L}, y \in \mathcal{Y}$ be upper bounded by λ_{\max} , then we have

$$\begin{aligned} C(\mathbf{r}(m)) &\leq \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(\beta |\mathcal{L}| \lambda_{\max} K r_{\max}) \\ &\leq 2^{|\mathcal{L}|} \exp(\beta |\mathcal{L}| \lambda_{\max} K r_{\max}) \end{aligned} \quad (4.54)$$

Thus the minimal probability in the stationary distribution

$$\begin{aligned} \pi_{\min}(\mathbf{r}(m)) &\triangleq \min_{\mathbf{y}} \pi_{\mathbf{y}}(\mathbf{r}(m)) \geq \frac{1}{C(\mathbf{r}(m))} \\ &\geq \exp(-|\mathcal{L}| \cdot (\log |2| + K \beta \lambda_{\max} r_{\max})). \end{aligned} \quad (4.55)$$

To utilize the existing bounds on convergence to the stationary distribution of discrete-time Markov chain, we uniformize the continuous-time Markov chain $Y(t)$. Uniformization [38] plays the role of bridging between discrete-time Markov chain and continuous-time Markov chain.

Let the transition rate matrix of $Y(t)$ is denoted by $Q = \{Q(\mathbf{y}, \mathbf{y}')\}$. Construct a discrete-time Markov chain $Z(n)$ with its probability transition matrix $P = I + Q/q_m$, where I is the identity matrix. Then consider a system that successive states visited form a Markov chain $Z(n)$ and the times at which the system changes its state form a Poisson process $N(t)$. Here $N(t)$ is an independent Poisson process with rate q_m . Then the state of this system at time t is denoted by $Z(N(t))$, which is called a *subordinated Markov chain* [38]. Let

$$q_m = |\mathcal{L}| \cdot \exp(K \beta \lambda_{\max} r_{\max}). \quad (4.56)$$

Since $\forall \mathbf{y}, \mathbf{y}', Q(\mathbf{y}, \mathbf{y}') \leq \exp(K\beta\lambda_{\max}r_{\max})$, and \mathbf{y} can at most transit to $|\mathcal{L}|$ other states, thus $\sum_{\mathbf{y} \neq \mathbf{y}'} Q(\mathbf{y}, \mathbf{y}') \leq |\mathcal{L}| \cdot \exp(K\beta\lambda_{\max}r_{\max}) = q_m$. Then by uniformization theorem [38], $Y(t)$ and $Z(N(t))$ has the same distribution, denoted by $Y(t) \stackrel{d}{=} Z(N(t))$.

Now let the vector $\boldsymbol{\omega}_m(t) = \{\omega_m(t, \mathbf{y})\}$ be the probabilities of all states at time $t_m + t$ ($0 \leq t \leq T_0$), given that the initial state at time t_m is $\mathbf{y}^0(m)$ and that the transmission aggressiveness during period m ($[t_m, t_{m+1})$) are $\mathbf{r}(m)$. Let $\mathbf{y}(t_m + t)$ be the state at time $t_m + t$, then

$$\begin{aligned}
& E[\bar{\lambda}_{iJ}(m) | \mathcal{F}_m] \\
&= \int_0^{T_0} E[\lambda_{iJ}(t_m + t) | \mathcal{F}_m] dt / T_0 \\
&= \int_0^{T_0} \sum_{\mathbf{y}'} \lambda_{iJ}^{\mathbf{y}'} \cdot \omega_m(t, \mathbf{y}') dt / T_0 \\
&= \sum_{\mathbf{y}'} \lambda_{iJ}^{\mathbf{y}'} \cdot \int_0^{T_0} \omega_m(t, \mathbf{y}') dt / T_0 \\
&= \sum_{\mathbf{y}'} \lambda_{iJ}^{\mathbf{y}'} \cdot \bar{\omega}_m(\mathbf{y}') \tag{4.57}
\end{aligned}$$

Where $\bar{\omega}_m(\mathbf{y}') = \int_0^{T_0} \omega_m(t, \mathbf{y}') dt / T_0$ is the time-averaged probability of state \mathbf{y}' in the interval.

Since the initial distribution is concentrated at a single definite starting state $\mathbf{y}^0(m)$, we denote this distribution by $\delta_{\mathbf{y}^0}$. We let $\pi_{\mathbf{y}^0}(\mathbf{r}(m))$ be the probability of $\mathbf{y}^0(m)$ in the stationary distribution of $Y(t)$. Let $\boldsymbol{\pi}(\mathbf{r}(m)) \triangleq \{\pi_{\mathbf{y}}(\mathbf{r}(m))\}$ be the stationary distribution of $Y(t)$, then by uniformization theorem [38], $\boldsymbol{\pi}(\mathbf{r}(m))$ is also the stationary distribution of $Z(n)$.

We use $\|\cdot\|_{TV}$ to denote the total variation distance between two distributions [17], which satisfies triangle inequality. We use ρ_2 to denote the second largest eigenvalue of transition matrix P . Thus for reversible discrete-time Markov chain $Z(n)$ with transition matrix P , and for any $n \geq 0$, we have

the following inequality [17]:

$$\|\delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \leq \frac{1}{2} \sqrt{\frac{1 - \pi_{\mathbf{y}^0}(\mathbf{r}(m))}{\pi_{\mathbf{y}^0}(\mathbf{r}(m))}} \cdot \rho_2^n \quad (4.58)$$

Therefore,

$$\begin{aligned} & \|\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \\ &= \left\| \sum_{n=0}^{\infty} \frac{(q_m t)^n}{n!} \exp(-q_m t) \delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(\mathbf{r}(m)) \right\|_{TV} \\ &\leq \sum_{n=0}^{\infty} \frac{(q_m t)^n}{n!} \exp(-q_m t) \|\delta_{\mathbf{y}^0} P^n - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \\ &\leq \frac{1}{2} \sqrt{\frac{1}{\pi_{\min}(\mathbf{r}(m))}} \cdot \exp(-q_m(1 - \rho_2)t) \end{aligned}$$

Further,

$$\begin{aligned} & \|\bar{\boldsymbol{\omega}}_m - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \quad (4.59) \\ &= \left\| \int_0^{T_0} [\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\mathbf{r}(m))] dt / T_0 \right\|_{TV} \\ &\leq \int_0^{T_0} \|\boldsymbol{\omega}_m(t) - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} dt / T_0 \\ &\leq \frac{1}{2} \sqrt{\frac{1}{\pi_{\min}(\mathbf{r}(m))}} \frac{1}{q_m(1 - \rho_2)T_0} \quad (4.60) \end{aligned}$$

Now we bound ρ_2 by Cheeger's inequality [17]:

$$\rho_2 \leq 1 - \phi^2/2$$

Where ϕ is the ‘‘Conductance’’ of P , defined as

$$\phi \triangleq \min_{N \subset \Omega, \pi(N) \in (0, 1/2]} \frac{F(N, N^c)}{\pi_N(\mathbf{r}(m))}$$

Here Ω is the state space, $\pi_N(\mathbf{r}(m)) = \sum_{\mathbf{y} \in N} \pi_{\mathbf{y}}(\mathbf{r}(m))$ and $F(N, N^c) = \sum_{\mathbf{y} \in N, \mathbf{y}' \in N^c} \pi_{\mathbf{y}}(\mathbf{r}(m)) P(\mathbf{y}, \mathbf{y}')$. Thus

$$\begin{aligned} \phi &\geq \min_{N \subset \Omega, \pi(N) \in (0, 1/2]} F(N, N^c) \\ &\geq \min_{\mathbf{y} \neq \mathbf{y}', P(\mathbf{y}, \mathbf{y}') > 0} F(\mathbf{y}, \mathbf{y}') \\ &\geq \pi_{\min}(\mathbf{r}(m)) / q_m \end{aligned}$$

then

$$\frac{1}{1 - \rho_2} \leq \frac{2}{\phi^2} = 2 \cdot q_m^2 [\pi_{\min}(\mathbf{r}(m))]^{-2}. \quad (4.61)$$

Combing (4.61), (4.55) , (4.56) with (4.60), it follows that

$$\|\bar{\omega}_m - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \quad (4.62)$$

$$\leq \frac{q_m}{T_0} [\pi_{\min}(\mathbf{r}(m))]^{-5/2} \quad (4.63)$$

$$\begin{aligned} &= (|\mathcal{L}|/T_0) \cdot \exp[(5/2|\mathcal{L}| + 1)K\beta\lambda_{\max}r_{\max} + 5/2|\mathcal{L}| \log 2] \\ &\triangleq C_4/T_0 \end{aligned} \quad (4.64)$$

So by (4.57), we have $\forall (i, J) \in \mathcal{L}$,

$$\begin{aligned} |E[\bar{\lambda}_{iJ}(m)|\mathcal{F}_m] - \lambda_{iJ}(m)| &= \left| \sum_{\mathbf{y}'} \lambda_{iJ}^{\mathbf{y}'} \cdot (\bar{\omega}_m(\mathbf{y}') - \pi_{\mathbf{y}'}(\mathbf{r}(m))) \right| \\ &\leq \lambda_{\max} \cdot 2 \|\bar{\omega}_m - \boldsymbol{\pi}(\mathbf{r}(m))\|_{TV} \leq 2C_4\lambda_{\max}/T_0 \end{aligned}$$

Let $J_{\max} \triangleq \max_{J:(i,J) \in \mathcal{L}, i \in \mathcal{N}} |J|$. Then by (4.20) we have

$$\begin{aligned} |B_i^{st}(m)| &\leq 2|\mathcal{L}|J_{\max}|E[\bar{\lambda}_{iJ}(m)|\mathcal{F}_m] - \lambda_{iJ}(m)| \\ &\leq 4C_4|\mathcal{L}|J_{\max}\lambda_{\max}/T_0, \quad \forall i \in \mathcal{N}, s \in S, t \in \mathcal{T}_s \end{aligned}$$

Since $\forall i \in \mathcal{N}, s \in S, t \in \mathcal{T}_s$, \hat{r}_i^{st} is bounded by some $\bar{r} > 0$, then $|\hat{r}_i^{st} - r_i^{st}(m)| \leq \bar{r} + r_{\max}$. Therefore

$$\begin{aligned} &|[\hat{\mathbf{r}} - \mathbf{r}(m)]^T \mathbf{B}(m)| \\ &\leq K|\mathcal{N}||S| \cdot (\bar{r} + r_{\max}) \cdot 4C_4|\mathcal{L}|J_{\max}\lambda_{\max}/T_0 \end{aligned} \quad (4.65)$$

$$= C_3/T_0 \quad (4.66)$$

Where $C_3 = K|\mathcal{N}||S| \cdot (\bar{r} + r_{\max}) \cdot 4C_4|\mathcal{L}|J_{\max}\lambda_{\max}$ is a positive constant.

This concludes the proof.

Chapter 5

Conclusions

A journey of a thousand miles must
begin with a single step.

Chinese Proverb

Markov approximation framework can be applied to design generally applicable, flexible, robust and easy-to-implement approximation algorithms, which are able to obtain near-optimal solutions for a wide range of combinatorial network optimization problems. However, in practice, the system-wide algorithms designed within Markov approximation framework can be diverge and unstable. There are mainly two reasons: noisy (or imprecise) measurements of network parameters and no assumptions of time-scale separation.

The central motivation of this thesis is to study the convergence issues of the system-wide algorithms. A major contribution of this thesis is to show the system-wide algorithms converge under mild conditions for two general scenarios: Markov chain over resource allocation algorithms and resource allocation algorithms over Markov chain. Note that the proof for the former scenario is inspired by the one used in Jiang-Walrand's paper [30]. Assured by these convergence results, we can apply Markov approximation framework to design distributed stochastic algorithms for combinatorial network optimization problems, which are able to converge to near-optimal solutions.

In this thesis, we first review the basics of Markov approximation framework. There are two key steps of Markov approximation framework: log-sum-exp approximation and distributed Markov chain Monte Carlo (MCMC). The later step gives us two degrees of freedom in designing algorithms: neighboring structures of configuration space and a generation mechanism of Markov chain for configuration transitions.

Then we show many examples of combinatorial network optimization. For scenario of Markov chain over resource allocation algorithms, we find some instances where the corresponding convergence results can be applied, including optimal selection of paths, trees, neighbors, channels and power levels for wireless, wireline and peer-to-peer networks. For scenario of resource allocation algorithms over Markov chain, we mainly focus on two instances: cross-layer design of wireless networks with deterministic channel models and wireless networks with network coding.

For cross-layer design of wireless networks with deterministic channel, we extend the well-studied conflict graph model to capture the flow interactions over the deterministic channels and characterize the feasible rate region. In the important cases such as single-hop multiple access wireless networks, we show that the feasible rate region is equal to the information theoretic capacity region. Then we consider general multi-hop wireless networks with both link-centric formulation and node-centric formulation. We approximately solve the optimization problem in a distributed manner by Markov approximation framework, including the scheduling algorithm and primal-dual flow-control algorithm. The primal-dual algorithm is shown to converge to the optimal solutions with or without time-scale separation assumption. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant step size and constant update interval. Numerical results illustrate not only the advantage of deterministic channel model over treating interference as noise, but also the convergence and optimality of our

joint scheduling and primal-dual flow control algorithms.

For cross-layer design of wireless networks with network coding, we consider both scenarios of multicast sessions with intra-session network coding and unicast sessions with suboptimal inter-session network coding. For intra-session network coding scenario, we adopt the random linear network coding algorithm; while for inter-session network coding algorithm, we adopt local butterfly structure based session decomposition scheme, i.e., a sub-optimal coding scheme. Then we exploit the wireless broadcast advantage and develop a unified distributed scheme for these scenarios guided by Markov approximation framework. The designed distributed stochastic algorithm is shown to converge to the optimal solutions with or without time-scale separation assumption. Further, we show the convergence to the bounded neighborhood of optimal solutions with probability one under constant step sizes and constant update intervals. Numerical results validate the analytical results as well. Actually the proposed distributed scheme can be applied to other inter-session network coding scenarios with variants of sub-optimal coding schemes.

In summary, Markov approximation framework assumes and utilizes only minimal structure of targeted combinatorial optimization problems. This keeps it simple and general in purpose to design distributed stochastic algorithms and is likely to ensure Markov approximation framework a niche in the arsenal of distributed optimization algorithms for communication networks.

It is always exciting to look ahead. We now conclude by discussing possible future directions. There are three natural extensions we need to address. First, for five examples listed in session 2.3.2., we want to design distributed algorithms without message passing. Second, we are concerned with the convergence time of the designed Markov chain. We need to ensure that the Markov chain mixes fast, i.e., it converges to the desired stationary distribution quickly. Third, we want to find more applications of Markov approximation framework. In dosing so, we need to exploit the structure information of problem instances

as much as possible. Then with the help of Markov approximation framework, we can give tailored solutions to solve the problem instances efficiently.

Bibliography

- [1] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.
- [2] A. Avestimehr, S. Diggavi, and D. Tse. A deterministic approach to wireless relay networks. In *Proceedings of 45th Allerton Conference on Communication, Control, and Computing*, 2007.
- [3] A. Avestimehr, S. Diggavi, and D. Tse. Wireless network information flow: a deterministic approach. *submitted to IEEE Transactions on Information Theory*, 2009. Available at <http://arxiv.org/abs/0906.5394>.
- [4] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [6] G. Bresler and D. Tse. The two-user Gaussian interference channel: a deterministic view. *European Transactions on Telecommunications*, 19(4):333–354, 2008.
- [7] V. Cadambe and S. Jafar. Interference Alignment and Degrees of Freedom of the K -User Interference Channel. *IEEE Transactions on Information Theory*, 54(8):3425–3441, 2008.

- [8] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Transactions on Information Theory*, 54(2):572, 2008.
- [9] P. Chaporkar and A. Proutiere. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *Proceedings of ACM MobiCom*, page 146, 2007.
- [10] M. Chen, S. Liew, Z. Shao, and C. Kai. Markov Approximation for Combinatorial Network Optimization. In *Proceedings of IEEE INFOCOM 2010*.
- [11] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. Chou. Utility maximization in peer-to-peer systems. In *Proc. ACM Sigmetrics*, 2008.
- [12] M. Chiang, S. Low, A. Calderbank, and J. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255, 2007.
- [13] T. Cover and J. Thomas. *Elements of information theory*. New York: Wiley, 1991.
- [14] T. Cover and J. Thomas. *Elements of information theory*. New York: Wiley, 2006.
- [15] T. Cui, L. Chen, and T. Ho. Distributed optimization in wireless networks using broadcast advantage. In *Proceedings of IEEE CDC*, 2007.
- [16] T. Cui, L. Chen, and T. Ho. Energy efficient opportunistic network coding for wireless networks. In *Proceedings of IEEE INFOCOM*, 2008.
- [17] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, pages 36–61, 1991.

- [18] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, 51(8):2745–2759, 2005.
- [19] A. Eryilmaz and D. Lun. Control for inter-session network coding. In *Proceedings of Workshop on Network Coding, Theory and Applications*, 2007.
- [20] A. Eryilmaz, A. Ozdaglar, and E. Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, 2007.
- [21] R. Glauber. Time-dependent statistics of the Ising model. *Journal of mathematical physics*, 4(2):294, 1963.
- [22] T. Ho, Y. Chang, and K. Han. On constructive network coding for multiple unicasts. In *Proceedings of 44th Allerton Conference*, 2006.
- [23] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.
- [24] T. Ho and H. Viswanathan. Dynamic algorithms for multicast with intra-session network coding. In *Proceedings of 43rd Annual Allerton Conference*, 2005.
- [25] H. Hoos and T. Stutzle. *Stochastic local search: Foundations and applications*. Morgan Kaufmann, 2005.
- [26] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 25(1):187, 1995.

- [27] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.
- [28] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wireless networks*, 11(4):471–487, 2005.
- [29] L. Jiang, D. Shah, J. Shin, and J. Walrand. Distributed Random Access Algorithm:Scheduling and Congestion Control. *Technical Report*, 2009. Available at <http://arxiv.org/abs/0907.1266>.
- [30] L. Jiang and J. Walrand. Convergence and Stability of a Distributed CSMA Algorithm for Maximal Network Throughput. Technical report, Technical Report, UC Berkeley, Mar. 2009. URL: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-43.html>.
- [31] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. In *Proceedings of 46th Allerton Conference on Communication, Control, and Computing, Urbana-Champaign, IL*, 2008.
- [32] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the Greedy maximal scheduling algorithm in multihop wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1132–1145, 2009.
- [33] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. XORs in the air: practical wireless network coding. *IEEE/ACM Transactions on Networking (TON)*, 16(3):497–510, 2008.
- [34] F. Kelly. *Reversibility and stochastic networks*. Wiley,Chichester, 1979.

- [35] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, pages 237–252, 1998.
- [36] H. Khalil. *Nonlinear systems*. Prentice Hall, Englewood Cliffs, NJ, 2001.
- [37] A. Khreishah, C. Wang, and N. Shroff. Optimization based rate control for communication networks with inter-session network coding. In *Proceedings of IEEE INFOCOM*, 2008.
- [38] M. Kijima. *Markov processes for stochastic modeling*. CRC Press, 1997.
- [39] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM transactions on networking*, 11(5):782–795, 2003.
- [40] B. Korte and J. Vygen. *Combinatorial optimization: theory and algorithms*. Springer Verlag, 2008.
- [41] H. Kushner and G. Yin. *Stochastic approximation and recursive algorithms and applications*. Springer Verlag, 2003.
- [42] P. Laarhoven and E. Aarts. *Simulated annealing: theory and applications*. Springer, 1987.
- [43] E. Lawler. *Combinatorial optimization: networks and matroids*. Dover Pubns, 2001.
- [44] M. Leconte, J. Ni, and R. Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. In *Proceedings of ACM MobiHoc*, pages 165–174, 2009.
- [45] A. Lehman and E. Lehman. Complexity classification of network information flow problems. In *Proceedings of SODA*, pages 142–150, 2004.

- [46] D. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times*. Amer Mathematical Society, 2009.
- [47] S. Li, R. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.
- [48] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 14(2):315, 2006.
- [49] X. Lin, N. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, 2006.
- [50] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. Poor. Convergence and Tradeoff of Utility-Optimal CSMA. *submitted for publication*, <http://arxiv.org/abs/0902.1996>, 2009.
- [51] S. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions On Networking*, 11(4):525–536, 2003.
- [52] S. Low and D. Lapsley. Optimization flow control I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking (TON)*, 7(6):861–874, 1999.
- [53] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking (ToN)*, 8(5):556–567, 2000.
- [54] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proceedings of ACM SIGMETRICS /IFIP Performance*, 2006.

- [55] J. Ni, B. Tan, and R. Srikant. Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks. In *Proceedings of IEEE INFOCOM Mini-Conference*, 2010.
- [56] C. Papadimitriou and K. Steiglitz. *Combinatorial optimization*. Prentice-Hall Englewood Cliffs, NJ, 1982.
- [57] B. Polyak. *Introduction to optimization*. Optimization Software Inc., 1987.
- [58] S. Rajagopalan and D. Shah. Distributed algorithm and reversible network. In *Proceedings of CISS*, 2008.
- [59] S. Rajagopalan, D. Shah, and J. Shin. Network adiabatic theorem: an efficient randomized protocol for contention resolution. In *Proceedings of SIGMETRICS/Performance 2009*, pages 133–144, 2009.
- [60] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *Proceedings of IEEE INFOCOM*, pages 1028–1036, 2007.
- [61] R. Srikant. *The mathematics of Internet congestion control*. Birkhauser, 2004.
- [62] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 1992.
- [63] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard. Network coding for multiple unicasts: An approach based on linear optimization. In *Proceedings of IEEE ISIT*, pages 1758–1762, 2006.
- [64] V. Vazirani. *Approximation algorithms*. Springer, 2001.

- [65] T. Voice. Stability of congestion control algorithms with multi-path routing and linear stochastic modelling of congestion control. *Ph.D. dissertation, University of Cambridge, Cambridge, UK*, May 2006.
- [66] J. Walrand. Entropy in communication and chemical systems. In *First International Symposium on Applied Sciences on Biomedical and Communication Technologies, (ISABEL'08)*, pages 1–5, 2008.
- [67] X. Wu, R. Srikant, and J. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE transactions on mobile computing*, 6(6):595, 2007.
- [68] R. Yeung, S. Li, N. Cai, and Z. Zhang. *Network coding theory*. Foundations and Trends in Communications and Information Theory, Now Publisher, 2006.