Non-convex Power Control and Scheduling in Wireless Ad hoc Networks

QIAN, Liping

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

July 2010

UMI Number: 3446032

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI®
Dissertation Publishing

UMI 3446032
Copyright 2011 by ProQuest LLC.
All rights reserved. This edition of the work is protected against
unauthorized copying under Title 17, United States Code.

ProQuest®

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

To My Family

摘要

由于無線媒介的廣播效應，鏈路間的同時通信會互相幹擾（這個現象對鄰近鏈路尤其明顯），因此會嚴重影響網絡的數據率和服務質量。所以抗幹擾是下一代無線網絡中一個必須解決的問題。一個重要的抗幹擾技術是合理地控制各鏈路的傳輸功率。 由于寬帶無線數據服務的普及，系統效益通常被用來衡量功率控制的好壞。

由于鏈路間的互相幹擾，通過功率控制最優化系統效益一般是一個 NP 困難問題。因此，盡管功率控制對優化系統效益狠重要，但是狠難被解決。這個論文的第一個目的就是用單調性優化的技術爲一系列的系統效益最優化問題找到最優的功率分配。不同于直接處理非凸問題，我們可以通過開發功率控制問題的單調性來避開非凸問題。在單波及多波網絡中，我們建立一個單調優化構架去通過功率控制最優化系統效益。相應地，爲了有效地獲得最優的功率分配，我們爲單波網絡提出 MAPEL 算法以及爲多波網絡提出 M-MAPEL 算法。這兩種算法的主要好處是可用作衡量其他解決相同問題的啓發式算法的性能。基于這兩算法的幫助，我們評估了幾個存在的算法。另一方面，通過調整 MAPEL 和 M-MAPEL 算法中的參數，我們能夠實現一個性能與收斂時間之間的折中。

然而，在鄰近鏈路同時通信時嚴重互相幹擾的幹擾受限無線網絡，單獨的功率控制不足以減輕鄰近鏈路間的強烈幹擾。 在這種情況下，鄰近的鏈路輪流通信，即調度，將會對提高系統性能起到一個重要的角色。然而，通過聯合的功率控制

和調度最優化系統效益一直是一個富有挑戰性的問題。通信鏈路間的信噪比耦合以及不同時段通信功率變化的靈活度致使聯合的功率控制和調度是一個非凸優化問題，因此最優解狠難獲得。所以，這個論文的第二個目的是找到聯合的功率控制和調度的最優解。具體地，我們利用此問題的單調性而不是凸性提出 S-MAPEL 算法，從而有效地獲得最優的聯合的功率控制和調度策略。爲了進一步降低計算複雜度，我們在 S-MAPEL 的基礎上提出了一個加速算法，A-S-MAPEL。這個算法主要利用了聯合的功率控制和調度中的對稱性。這些算法獲得的最優的聯合的功率控制和調度策略可以作爲一個有用的標准去衡量其它存在的算法。基于提出的算法的幫助，我們發現對現階段的無線設備來說，開關調度就系統效益最優化方面狠有實際意義。

隨著無線分布式網路（如 Ad hoc 網絡，傳感器網絡）的普及， 狠需要設計一個分布式功率控制的算法。一般來說，分布式功率控制由于缺少中央控制器而變得更加複雜。因此，我們的第三個目的就是爲無線 Ad hoc 網絡設計一個分布性的功率控制算法，這個算法允許每個鏈路根據一定量的鏈路間交互分布且異步地更新自己的傳輸功率。盡管功率控制的非凸性，然而這個算法可以獲得最優化系統效益的功率分配。相對于其它存在的算法，我們這個算法對系統效益函數的定義沒有嚴格要求。

Abstract

Due to the broadcast nature of wireless medium, simultaneous transmissions interfere with each other (especially transmissions on nearby links), thus adversely affecting data rates and Quality of Service (QoS) in the system. Interference mitigation is therefore a fundamental issue that must be addressed in next generation wireless networks. An important technique for this is to control the links' transmission power. Driven by the wide spread of broadband wireless data services, a system-wide efficiency metric (i.e., system utility) is typically used to characterize the advantage of power control.

Maximizing a system-wide utility through power control is an NP-hard problem in general due to the complicated coupling interference between links. Thus, it is difficult to solve despite its paramount importance. The first goal of this thesis is to find global optimal power allocations to a variety of system utility maximization (SUM) problems based on the recent advances in monotonic optimization. Instead of tackling the non-convexity issue head on, we bypass non-convexity by exploiting the monotonic nature of the power control problem. In particular, we establish a monotonic optimization framework to maximize a system utility through power control in single-carrier or multi-carrier wireless networks. Furthermore, MAPEL and M-MAPEL are respectively proposed to obtain the global optimal power allocation efficiently in single-carrier or multi-carrier wireless networks. The main benefit of MAPEL and M-MAPEL

is to provide an important benchmark for performance evaluation of other heuristic algorithms targeting the same problem. With the help of MAPEL or M-MAPEL, we evaluate the performance of several existing algorithms through extensive simulations. On the other hand, by tuning the approximation factor in MAPEL and M-MAPEL, we could engineer a desirable tradeoff between optimality and convergence time.

In interference-limited wireless networks where simultaneous transmissions on nearby links heavily interfere with each other, however, power control alone is not sufficient to eliminate strong levels of interference between close-by links. In this case, scheduling, which allows close-by links to take turns to be active, plays a crucial role for achieving high system performance. Joint power control and scheduling that maximizes the system utility has long been a challenging problem. The complicated coupling between the signal-to-interference ratio of concurrently active links as well as the flexibility to vary power allocation over time gives rise to a series of non-convex optimization problems, for which the global optimal solution is hard to obtain. The second goal of this thesis is to solve the non-convex joint power control and scheduling problems efficiently in a global optimal manner. In particular, it is the monotonicity rather than the convexity of the problem that we exploit to devise an efficient algorithm, referred to as S-MAPEL, to obtain the global optimal solution. To further reduce the complexity, we propose an accelerated algorithm, referred to as A-S-MAPEL, based on the inherent symmetry of the optimal solution. The optimal joint-power-control-and-scheduling solution obtained by the proposed algorithms serves as a useful benchmark for evaluating other existing schemes. With the help of this benchmark, we find that on-off scheduling is of much practical value in terms of system utility maximization if "off-the-shelf" wireless devices are to be used.

With the proliferation of wireless infrastructureless networks such as ad hoc and sensor networks, it is increasingly crucial to devise an algorithm that solves the power control problem in a distributed fashion. In general, distributed power control is more complicated due to the lack of centralized infrastructure. As the third goal of this thesis, we consider a distributed power control algorithm for infrastructureless ad hoc wireless networks, where each link distributively and asynchronously updates its transmission power with limited message passing among links. This algorithm provably converges to the optimal strategy that picks global optimal solutions with probability 1 despite the non-convexity of the power control problem. In contrast with existing distributed power control algorithms, our algorithm makes no stringent assumptions on the system utility functions. In particular, the utility function is allowed to be *concave* or *non-concave*, *differentiable* or *non-differentiable*, *continuous* or *discontinuous*, and *monotonic* or *non-monotonic*.

Acknowledgements

My graduate study in CUHK is one of the most memorable periods in my life. I feel very fortunate to have met and worked with so many incredible individuals during the journey to my Ph. D. I have learned a lot from these individuals. First of all, I would like to express my sincerest gratitude to my supervisor, Professor Ying Jun (Angela) Zhang for her great guidance and support throughout the past four years. Her boundless energy, keen insights, constant encouragement and devotion to her students are highly beneficial to my academic growth. I also greatly appreciate her care and concern for my intellectual and personal growth. Prof. Zhang not only advises me how to do great academic research, but also teaches me how to become a self-restrained person.

I am also grateful to Prof. Mung Chiang. Prof. Chiang served as my advisor during my visiting in Princeton University and gave me many valuable comments on my research. I got a lot of insights from his research papers when I initiated this topic. I would also like to thank Prof. Jianwei Huang for his insightful comments which helped me to improve the quality of my first research paper. I also wish to thank my thesis committee members, whose broad perspective and suggestions have helped me in refining this thesis.

I am also deeply indebted to all my officemates who have contributed greatly to provide a supportive and collaborative research atmosphere. I am so proud

of being a part of the brilliant and fruitful group. Many thanks to Shengli Zhang, Caihong Kai, Liqun Fu, Wei Xie, Weiliang Li, Suzhi Bi and Leiyi Yao, with whom I have opportunities to attend numerous motivating discussions. Of course, I have to thank my dear friend Yuan Wu and all my friends at CUHK, for making the years so enjoyable.

Last but not least, I would like to sincerely thank my parents and my sister for their support, encouragement and love throughout my life. This dissertation is dedicated to them.

Contents

摘要 **ii**

Abstract **iv**

Acknowledgements **vii**

List of Tables **xiv**

List of Figures **xv**

List of Acronyms **xix**

List of Symbols **xxi**

1 Introduction **1**

1.1 Challenges and Motivations 2

1.2 Background and Related Work 6

1.2.1 Pure Power Control 7

1.2.2 Joint Power Control and Scheduling 10

1.2.3 Distributed Power Control 10

1.3 Overview of Contributions 11

2 MAPEL: Achieving Global Optimality for Non-convex Wireless Power Control Problems 17

2.1 System Model and Problem Formulation 18

2.2 Power Control as Monotonic Optimization (MO) 22

2.3 The MAPEL Algorithm 26

2.3.1 Related Mathematical Preliminaries 26

2.3.2 The MAPEL Algorithm 28

2.3.3 Global Convergence 30

2.3.4 Trade-off between Performance and Convergence Time . . 32

2.3.5 Extension to Max-min SINR Power Control 33

2.4 Performance Evaluation of MAPEL 33

2.5 Providing Benchmark For Existing Power Control Algorithms . . 35

2.5.1 Review of Existing Power Control Algorithms 37

2.5.2 Performance Study of SPC Algorithm and ADP Algorithm 38

2.6 Summary 44

2.7 Appendix 45

2.7.1 Proof of Theorem 2.2 45

2.7.2 Proof of Theorem 2.3 46

3 M-MAPEL: Monotonic Optimization for Non-concave Power Control in Multi-carrier Wireless Networks 48

3.1 System Model and Problem Formulation 49

3.2 Power Control as Monotonic Optimization 52

3.3 The M-MAPEL Algorithm 54

3.3.1 The M-MAPEL Algorithm 55

3.3.2 Global Convergence 57

3.3.3 Trade-off between Performance and Convergence Time . . 57

3.4 Performance Evaluation 59

3.4.1 Performance Evaluation of M-MAPEL 59

3.4.2 Providing Benchmark for Existing Algorithms 62

3.5 Summary 65

3.6 Appendix 65

3.6.1 Proof of Theorem 3.2 65

4 S-MAPEL: Monotonic Optimization for Non-convex Joint Power Control and Scheduling Problems 69

4.1 System Model and Problem Formulation 70

4.2 Joint Power Control and Scheduling as Monotonic Optimization . 73

4.2.1 Discretization 73

4.2.2 Monotonic Optimization 75

4.3 THE S-MAPEL ALGORITHM 77

4.3.1 General MO Algorithm 78

4.3.2 The S-MAPEL Algorithm 79

4.3.3 Global Convergence 82

4.4 An Accelerated Algorithm for Joint Power Control and Scheduling 84

4.5 Performance Evaluation 86

4.5.1 Near Optimality of A-S-MAPEL 86

4.5.2 Optimal Joint Power Control and Scheduling vs. Node Density 88

4.5.3 Performance Study of On-off Power Control, Pure Power Control and On-off Scheduling 90

4.6 Discussion 94

4.7 Summary 97

4.8 Appendix 98

4.8.1 Proof of Theorem 4.2 98

5 SEER: Globally Optimal Distributed Power Control for Nonconcave Utility Maximization 101

5.1 System Model and Problem Formulation 102

5.2 The SEER Algorithm 103

5.2.1 Mathematical Preliminaries Related to Gibbs Sampling . . 104

5.2.2 Discrete-SEER 105

5.2.3 Continuous-SEER 111

5.3 Simulation Results 113

5.3.1 Effect of θ 113

5.3.2 Comparison with Existing Distributed Algorithms 115

5.4 Discussion 118

5.4.1 Selection of θ 118

5.4.2 Effect of Message Delay 122

5.4.3 Effect of Message Loss 126

5.5 Summary 127

6 Conclusions and Future Work **129**

6.1 Summary of Thesis Contributions 129

6.2 Future Work 131

References **133**

List of Tables

1.1 Comparison of pure power control versus joint power control and scheduling 5

2.1 Optimality of SPC Algorithm as well as ADP Algorithm 43

3.1 The optimal data rates and power allocations obtained by M-MAPEL for different objective functions 68

4.1 Comparison of pure power control, on-off scheduling versus joint power control and scheduling 92

List of Figures

1.1 A two-link network with $G_{11} = 0.1$, $G_{22} = 0.2$, $G_{12} = G_{21} = 0.05$, where G_{ij} denotes the channel gain from node T_i and node R_j. . . 4

1.2 The data rate regions obtained by (a) pure power control strategy and (b) joint power control and scheduling strategy, respectively. 4

2.1 An example of sigmoidal function. 21

2.2 Shapes of $\mathcal{G}$ and Θ for a two-link network 25

2.3 Illustration about related mathematical preliminaries for MAPEL algorithm 28

2.4 Obtained weighted sum throughput and number of iterations for different approximation factor δ 34

2.5 A network topology with four links 36

2.6 The relationship between optimal transmission power and distance d 36

2.7 Maximal weighted sum throughput achieved by MAPEL algorithm as well as SPC algorithm for 500 different initial feasible power allocations in G_1 network 39

2.8 Maximal weighted sum throughput achieved by MAPEL algorithm as well as ADP algorithm for 500 different initial feasible power allocations in G_1 network 40

2.9 Maximal weighted sum throughput achieved by MAPEL algorithm as well as SPC algorithm for 500 different initial feasible power allocations in G_2 network 40

2.10 Maximal weighted sum throughput achieved by MAPEL algorithm as well as ADP algorithm for 500 different initial feasible power allocations in G_2 network 41

2.11 Average sum throughput of different algorithms in n-link networks 42

2.12 Average sum throughput of different algorithms versus the data rate constraint in 4-link networks 44

3.1 Obtained total throughput and number of iterations for different error tolerance ε 60

3.2 Obtained summation of sigmoidal functions and number of iterations for different error tolerance ε 61

3.3 A network of four subcarriers with four links 62

3.4 Maximum total throughput achieved by M-MAPEL as well as SCALE for 100 channel gain realizations in D_1 network 64

3.5 Maximum total throughput achieved by M-MAPEL as well as SCALE for 100 channel gain realizations in D_2 network 64

4.1 The modified projection $\pi_o^{\mathcal{G}'}(z)$. 80

4.2 A network topology with four links. 87

4.3 Obtained total proportional fairness for different error tolerance ϵ. 87

4.4 The number of iterations needed for different error tolerance ϵ when obtaining total proportional fairness. 88

4.5 Obtained summation of sigmoidal functions for different error tolerance ϵ. 88

4.6 The number of iterations needed for different error tolerance ϵ when obtaining summation of sigmoidal functions. 89

4.7 The optimal allocation vs. node density (P.A.: Power Allocation). 90

4.8 Average system performance of pure power control vs. joint power control and scheduling in n-link networks. 91

4.9 Various stepsizes are used for the dual decomposition. 97

5.1 Obtained Proportional Fairness and number of iterations for different θ. 114

5.2 Obtained Total Throughput and number of iterations for different θ. 115

5.3 Maximal Total throughput achieved by Continuous-SEER algorithm as well as ADP algorithm for 100 different initial feasible power allocations. 117

5.4 The complexity comparison between Continuous-SEER and MAPEL. 117

5.5 Obtained Total Throughput and number of iterations for different adaptive θ. 122

5.6 The effects of message delay on the convergence and the performance of Continuous-SEER I 125

5.7 The effects of message delay on the convergence and the performance of Continuous-SEER II 126

5.8 The effects of message loss on the convergence and the performance of Continuous-SEER I 127

5.9 The effects of message loss on the convergence and the performance of Continuous-SEER II 128

List of Acronyms

QoS:	Quality of Service
SINR:	Single to Interference-plus-Noise Ratio
SUM:	System Utility Maximization
WTM:	Weighted Throughput Maximization
MO:	Monotonic Optimization
MAPEL:	MO-bAsed PowEr aLlocation
M-MAPEL:	Multi-carrier MAPEL
S-MAPEL:	Scheduling MAPEL
A-S-MAPEL:	Accelerated S-MAPEL
SEER:	aSynchronous distributEd powEr contRol
GP:	Geometric Programming
SP:	Signomial Programming
SPC:	SP Condensation
ADP:	Asynchronous Distributed Pricing
OFDMA:	Orthogonal Frequency Division Multiple Access
IWF:	Iterative Water Filling
SCALE:	Successive Convex Approximation for Low-complExity
KKT:	Karush-Kuhn-Tucker
OSB:	Optimal Spectrum Balancing
CDMA:	Code Division Multiple Access

CBS:	Current best solution
CBV:	Current best value
TTF:	Total throughput function
PFF:	Proportional fairness function
MFF:	Max-min fairness function
SSF:	Summation of sigmoidal functions
RSP:	Received signal power
PDF:	Probability density function

List of Symbols

M:	Number of links in a wireless network
$\mathcal{M}$:	Set of links in a wireless network
T_i:	Transmitter node of link i
R_i:	Receiver node of link i
G_{ij}:	Channel gain between node T_i and node R_j
$\boldsymbol{G}$:	Complete channel matrix $[G_{ij}]$
p_i:	Transmission power of link i
$\boldsymbol{p}$:	Transmission power vector
$P_i^{\max}$:	Maximum transmission power of link i
$\boldsymbol{P}_i^{\max}$:	Maximum transmission power vector
n_i:	Received noise of link i
$\gamma_i(\boldsymbol{p})$:	Received SINR of link i on the transmission power vector $\boldsymbol{p}$
$\gamma(\boldsymbol{p})$:	Achieved SINR vector of all links on the transmission power vector $\boldsymbol{p}$
$\log_2(1+\gamma_i(\boldsymbol{p}))$:	Data rate of link i on the transmission power vector $\boldsymbol{p}$
Γ_i:	SINR gap of link i
$\boldsymbol{p}^*$:	Optimal power allocation
$r_i^{\min}$:	Minimum data rate requirement of link i
$\boldsymbol{r}^{\min}$:	Vector of minimum data rate requirements
$\gamma_i^{\min}$:	Minimum SINR requirement of link i

w_i:	Priority weight of link i
$\boldsymbol{w}$:	Vector of priority weights
$\mathbb{R}$:	Real domain
$\mathbb{R}^N$:	N-dim real domain
$\mathbb{R}_+^M$:	M-dim nonnegative real domain
$\mathcal{P}$:	Set of feasible power allocation
$\mathcal{P}_i$:	Set of feasible power allocation of link i
$\mathcal{P}^*$:	Set of global optimal power allocation
$\boldsymbol{e}_j$:	Unit vector whose every element is equal to zero except the jth element being 1
$\mathcal{A} \setminus \mathcal{B}$:	Set $\{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{A} \text{ and } \boldsymbol{x} \notin \mathcal{B}\}$
$\mathcal{F}_{\boldsymbol{y}}$:	Set $\{\boldsymbol{y}' \in \mathbb{R}_+^M \mid \boldsymbol{y}' \succeq \boldsymbol{y} \text{ and } \boldsymbol{y}' \neq \boldsymbol{y}\}$
δ:	Approximation factor
ϵ:	Error tolerance
tol:	Error tolerance
d_{ij}:	Distance between node T_i and node R_j
L:	Number of subcarriers
$\mathcal{L}$:	Subcarrier index set
G_{ij}^l:	Channel gain between nodes T_i and R_j over subcarrier l
$\boldsymbol{G}^l$:	Channel submatrix over subcarrier l, i.e., $[G_{ij}^l]$
p_i^l:	Transmission power of link i over subcarrier l
$P_i^{\max,l}$:	Maximum transmission power of link i over subcarrier l
$P_i^{\max}$:	Maximum transmission power of link i
$\boldsymbol{p}^l$:	Transmission power vector over subcarrier l
n_i^l:	Received noise of link i over subcarrier l
$\log_2(1+\gamma_i^l(\boldsymbol{p}^l))$:	Data rate of link i over subcarrier l
r_i:	Data rate of link i

$\boldsymbol{r}$:	Vector of data rates
$U(\cdot)$:	System utility function
$U_i(\cdot)$:	Utility function of link i
z_i^l:	The $(M \times (l-1)+i)$th element of $\boldsymbol{z}$
$\boldsymbol{e}_i^l$:	Unit vector whose every element is equal to zero
	except the $(M \times (l-1)+i)$th element being 1
T:	Time interval under consideration
$p_i(t)$:	Transmission power of link i at time instant t
$\boldsymbol{p}(t)$:	Transmission power vector at time instant t
$\gamma_i(\boldsymbol{p}(t))$:	Received SINR of link i at time instant t
$r_i(t)$:	Instantaneous data rate of link i at time instant t
$\mathcal{R}(t)$:	Achievable instantaneous data rate set at time instant t
$\bar{\mathcal{R}}$:	Achievable average data rate set
$\mathcal{K}$:	Time interval index set
β^k:	Normalized time length of the kth time interval
	with respect to T
$\boldsymbol{\beta}$:	Vector of normalized time length
$(\boldsymbol{a}, \boldsymbol{b})$:	Concatenation of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$\log_2(1+\Gamma_i\gamma_i(\boldsymbol{p}(t)))$:	Instantaneous data rate of link i at time instant t
p_i^k:	Transmission power of link i in time interval k
$\boldsymbol{p}^k$:	Transmission power vector in time interval k
$\mathbf{0}$:	Vector with every element being 0
$\mathbf{1}$:	Vector with every element being 1
$\preceq$:	Component-wise smaller than or equal to
$\succeq$:	Component-wise larger than or equal to
$\prec$:	Component-wise smaller than
$\succ$:	Component-wise larger than

ρ: Probability of message loss

Chapter 1

Introduction

Driven by the increasing popularity of wireless broadband services, future wireless networks are expected to provide high-data-rate services to densely populated user environments. Due to the broadcast nature of wireless medium, simultaneous transmissions interfere with each other (especially transmissions on nearby links), thus adversely affecting data rates and Quality of Service (QoS) in the system. Interference mitigation is therefore a fundamental issue that must be addressed in next generation wireless networks.

Transmission power control in wireless networks has been extensively studied over the last two decades as an important mechanism to mitigate the adverse effect of interference (see a recent survey [1]). The research in this area can be divided into two main threads. The first thread is concerned with achieving fixed SINR (SINR: signal to interference-plus-noise ratio) targets with minimum transmission power. This formulation is motivated by traditional voice communications, where an SINR higher than a prescribed value is not helpful in terms of further improving user-perceived QoS. The second thread is concerned with achieving maximum system efficiency with power control. This formulation is motivated by data communication applications, where higher

SINR means higher data rate and better QoS. Such an optimization becomes more important as data applications will be dominant in next generation wireless networks (e.g., 4G and all IP-based communication systems).

In this chapter, we first discuss the technical challenges for implementing efficient power control in Section 1.1. We then give an introduction to related work in the area of power control in Section 1.2. Finally, in Section 1.3, we give an overview of the main contributions of this thesis.

1.1 Challenges and Motivations

Due to the convex nature, the fixed SINR target case has been well studied and implemented in the context of wireless cellular communications (e.g., [2–10]). Thanks to the explosive growth of wireless data services, there has been tremendous recent interest in finding an optimal transmission power allocation that maximizes a system-wide efficiency metric (i.e., utility) while satisfying the individual user's QoS requirements (e.g., [10–14]). However, due to the complicated interference coupling between links, optimal power control in a general SINR region is known to be a non-convex optimization problem, and hence is more difficult to solve despite its paramount importance.

A majority of efforts in solving the system utility maximization (SUM) problem aim to convexify it through transformation [11], reparameterization [14], relaxation[12], and approximation [11], oftentimes compromising the global optimality. To address the issue, we first propose the MAPEL algorithm (MObAsed PowEr aLlocation) algorithm, which is the first algorithm in the literature that can achieve the global optimal solution of the SUM problem in the general SINR regime [15]. Then, to solve the SUM problem in multi-carrier wireless networks, we propose the M-MAPEL (where the prefix "M" stands for

Multi-carrier) algorithm [16]. The key idea of these two proposed algorithms largely come from the latest development of monotonic optimization (MO) [18].

In dense networks, power control alone is not sufficient to eliminate strong levels of interference between close-by links. In this case, scheduling, which allows close-by links to take turns to be active, is indispensable. This is illustrated in the following motivating example.

Motivating Example: Consider a two-link network as shown in Fig. 1.1, where links are close to each other. Assume that the maximum allowable transmission power of each link is 1.0W. Meanwhile, the noise power at each receiver is 10^{-4}W. In this case, the achievable data rate region by using power control only is given in Fig. 1.2(a), where each point in the data rate region corresponds to a feasible power allocation strategy. On the other hand, if we integrate power control with scheduling, the achievable data rate region is considerably enlarged and becomes convex, as shown in Fig. 1.2(b). For further illustration, let us take a specific utility, namely proportional fair utility $U(r_1, r_2) = \log(r_1) + \log(r_2)$, as an example, where r_1 and r_2 denote the data rates of link 1 and link 2, respectively. The optimal power-control and joint-power-control-and-scheduling strategies that maximize the utility function are shown in Table 1.1. It can be seen that without scheduling, the data rates as well as the system utility are low due to severe co-channel interference. In contrast, the joint-power-control-and-scheduling strategy leads to much higher utility and data rates. This motivating example suggests that it can be spectrally inefficient to operate with power control alone in dense wireless networks. Joint power control and scheduling is a natural remedy to alleviate interference in this case.

Joint power control and scheduling is by nature much more challenging than pure power control, for the problem involves not only complicated coupling

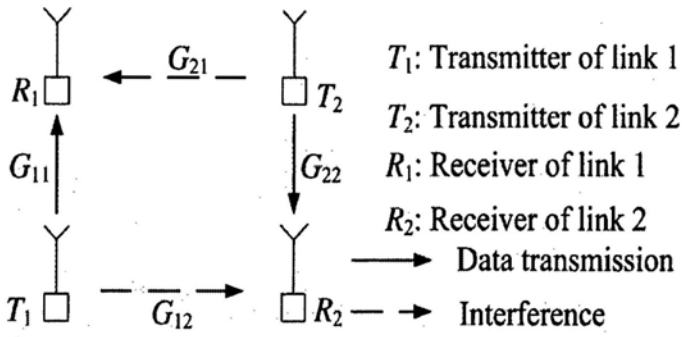
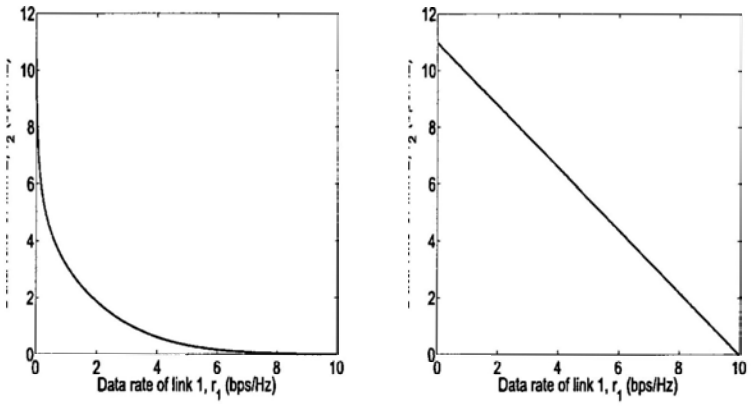


Figure 1.1: A two-link network with $G_{11} = 0.1$, $G_{22} = 0.2$, $G_{12} = G_{21} = 0.05$, where G_{ij} denotes the channel gain from node T_i and node R_j.



(a) (b)

Figure 1.2: The data rate regions obtained by (a) pure power control strategy and (b) joint power control and scheduling strategy, respectively.

between mutual interference of concurrently active links, but also that between the power allocation during different time periods. Notably, optimal joint power control and scheduling that maximizes a system utility in a general SINR region has remained an open problem to this day, despite the tremendous efforts that have been devoted to this subject (e.g., [19–22]). In this thesis, we address this problem by proposing an efficient algorithm, referred to as S-MAPEL (where the prefix "S" stands for Scheduling), to obtain the global optimal solution of the joint power control and scheduling problem [17]. The proposed algorithm

Table 1.1: Comparison of pure power control versus joint power control and scheduling

	Power control without scheduling	
	Optimal transmission power	
	$p_1^* = 1.0\text{W}$	$p_2^* = 0.71\text{W}$
Data rate	$r_1^* = 1.9294\text{bps/Hz}$	$r_2^* = 1.9390\text{bps/Hz}$
Overall utility	1.3194	

	Joint power control and scheduling	
	Optimal transmission power and scheduling	
	$p_1^*(t) = 1.0\text{W}, \forall t \in [0, 0.5T)$ $p_1^*(t) = 0\text{W}, \forall t \in [0.5T, T]$	$p_2^*(t) = 0\text{W}, \forall t \in [0, 0.5T)$ $p_2^*(t) = 1.0\text{W}, \forall t \in [0.5T, T]$
Data rate	$r_1^* = 4.9836\text{bps/Hz}$	$r_2^* = 5.4833\text{bps/Hz}$
Overall utility	3.3079	

* T is a given period of time, in which the channel gains keep constant.

complements our work MAPEL [15] that has only considered power control without scheduling. One key distinction of the problem formulation is that the system utility function is allowed to be *either concave or non-concave*, as long as it is monotonic. The results of this work address the following important questions: (i) given a wireless network, which links should transmit together and which should not; (ii) how much power and for how long should each link transmit to maximize the overall system utility while satisfying individual user's QoS requirements?

However, MAPEL, M-MAPEL and S-MAPEL can not be applied to infras-

tructureless ad hoc wireless networks, because they are largely centralized algorithms. With the proliferation of wireless infrastructureless networks such as ad hoc and sensor networks, it is increasingly crucial to devise an algorithm that solves the power control problem in a distributed fashion. In general, distributed power control is more complicated due to the lack of centralized infrastructure. Recently, distributed power control has attracted a lot of research interests (e.g., [13, 14, 23–25]). However, due to the non-convex nature of power control problems, it is hard for existing work to find the optimal power control solution in a distributed manner. Therefore, it is significant for us to propose a SEER (aSynchronous distributEd powEr contRol) algorithm to achieve the global optimal power allocation in a distributed fashion. The key idea of SEER largely comes from Gibbs Sampling, which is a well-studied optimization algorithm in fields such as statistical physics and image processing [50, 51].

To the best of our knowledge, power control in wireless networks has been a well-known topic for a long time. In the literature, there have been significant amount of efforts spent on it. In what follows, some related work will be introduced, from which we get a lot of insights and inspiration to conduct our research.

1.2 Background and Related Work

In this section, we review state-of-art work with respect to wireless power control, including pure power control, joint power control and scheduling, and distributed power control.

1.2.1 Pure Power Control

Due to the wide spread of broadband wireless data services, pure power control without scheduling generally aims to maximize a system-wide utility through allocating constant optimal power levels to links for the entire period of interest T. The existing work is broadly classified into the following categories in single-carrier wireless networks.

- ***Optimal Solutions: Convexification through Logarithmic Transformation***

The celebrated theorems in [8, 10, 28] show that the feasible SINR region, although not convex, is log-concave. This implies that the seemingly non-convex power control problem can be turned into a convex one by logarithmic transformation of SINR if the system-wide utility is log-concave in the feasible SINR region. This important finding greatly simplifies the task of handling the problem. It is based on this convexification that many distributed and convergent algorithms have been developed in recent years [13, 14]. However, such convexification only applies to special utility functions that satisfy the log-concave requirement. In particular, utility functions that fall into this category only include max-min fair utility, proportional fair utility, and α-fairness utility functions with $\alpha > 1$. Many utility functions, including simple ones such as throughput maximization, do not satisfy these properties.

- ***Suboptimal Solutions: Convexification through Approximation***

When the system-wide utility is not log-concave in SINR, maximizing a system-wide utility through power control is difficult to solve despite its paramount importance. Most previous work solves it through convexifying the power control problem with approximation [11] and iterative approximation [12]. Due

to the non-convex nature of power control problems, however, the algorithms proposed in [11, 12] are likely to converge to a suboptimal solution.

In particular, ref. [11] tackles the weighted throughput maximization (WTM) problem by making the high SINR assumption where the SINR of each link is much larger than 0dB. Based on the high-SINR assumption, the individual data rate $\log_2(1 + \text{SINR}_i)$ can be approximated by $\log_2(\text{SINR}_i)$, where SINR_i corresponds to the SINR of link i. Under such approximation, the WTM problem can be transformed into a convex one in the form of geometric programming (GP) by proper change of variables, and thus can be solved efficiently in a centralized fashion. Unfortunately, the high-SINR assumption is not valid in general for practical wireless adhoc networks when nearby links heavily interfere with each other. As a result, standard GP often yields a solution that is far from optimum due to possible strong interferences between links nearby. Compared with GP, the work in [12] does not require the high-SINR assumption. In particular, the authors in [12] first transform the WTM problem into an equivalent signomial programming (SP), which is provably NP-hard to solve. Then the authors adopt a successive convex programming method, SP Condensation (SPC) algorithm to solve SP. Similar to many algorithms used to solve non-convex optimization, the SPC algorithm only guarantees local optimal solutions. An improper initialization may considerably degrade the system throughput. To date, achieving a global optimal solution of the WTM problem still is an open problem, not to mention maximizing a system-wide utility through power control.

Not only in single-carrier wireless networks but also in multi-carrier wireless networks, many efforts [29–38, 40, 42–44] have been spent on the power control problem. For example, most of the prior work (e.g., [29–38]) considers OFDMA (Orthogonal Frequency Division Multiple Access) to avoid co-channel

interference, in which case no more than one link can transmit non-zero power on the same subcarrier simultaneously. This overly conservative assumption may lead to significant underutilization of radio spectrum when links are sufficiently apart, implying that frequency can be reused without causing severe co-channel interference. Contrary to [29–38], work [40, 42–44] focuses on the power control problem where frequency spectrum is reused whenever it yields a better system performance. In particular, work [40, 42–44] aims to maximize the total throughput through power control in multi-carrier wireless networks. In [44], an iterative water filling (IWF) scheme is proposed, where each link repeatedly measures the aggregate interference received from all other links, and then greedily water-pours their own power allocation without regard for the impact that would be had on other links. In [42], a Successive Convex Approximation for Low-complExity (SCALE) algorithm is presented to allocate transmission power through successive approximations until a KKT (Karush-Kuhn-Tucker) point is reached. Although SCALE performs significantly better than IWF, both algorithms are only capable of finding a sub-optimal solution, due to non-convexity of the optimization problem. On the other hand, several global optimization algorithms are also proposed in [40, 43] at the cost of prohibitively high complexity. The Optimal Spectrum Balancing (OSB) algorithm [43] makes use of a grid-search (exhaustive search) to find the optimal power allocation. Another approach [40] is to find the global optimum by searching all the roots of first-order necessary conditions and boundary points. However, this approach cannot be generalized to networks with more than 2 links, and searching all roots of a set of nonlinear equations is a computationally challenging task as well. So far, efficiently achieving the global optimal power allocation to system utility maximization is still an open problem, even for a objective function as simple as throughput maximization.

1.2.2 Joint Power Control and Scheduling

Driven by the fact that joint power control and scheduling can alleviate interference more efficiently than pure power control, the tremendous recent efforts have already been devoted to this subject. In [19] and [20], the authors aimed at scheduling as many link transmissions as possible in a given period of time, subject to the constraint on the minimum SINR requirement of each link. Utility-maximizing power controlled scheduling was previously studied in [21] and [22] under different contexts. Ref. [21] studied a downlink code division multiple access (CDMA) system, where a dual problem was solved to obtain the solution. Due to the non-convexity of the problem, however, the duality gap is non-zero. Hence, the solution obtained in [21] is not guaranteed to be optimal. Cruz et al investigated joint routing, scheduling and power control in multihop wireless networks in [22]. This paper focuses on a low-SINR regime, so that the data rate is a linear function of SINR. In this case, the optimal solution is simply a linear combination of extremal points, and hence easy to obtain. However, the low-SINR assumption may not be valid in practice, as links tend to maintain a reasonably high SINR for signal reception quality. Notably, optimal joint power control and scheduling that maximizes a system utility in a general SINR region has remained an open problem to this day, despite significant amount of efforts on it.

1.2.3 Distributed Power Control

Recently, distributed power control has attracted a lot of research interests due to its paramount importance. The power control problem is generally formulated as a noncooperative pricing game [23–25] or as a KKT decomposition [13, 14]. Due to the non-convex nature of power control problems, the dis-

tributed power control algorithms are likely to converge to a suboptimal solution. For example, the algorithms proposed in [13, 14] can obtain a global optimal solution only when the utility functions are strictly increasing, twice differentiable and strictly log-concave in the feasible SINR region. In practice, however, many utility functions, including simple ones such as throughput maximization, do not satisfy these properties. In this case, it is hard to for existing algorithms to find the optimal power control solution in a distributed manner. Furthermore, existing distributed algorithms require all links to update their transmission power at the same time. However, synchronous update is generally difficult to achieve due to the lack of a central clock in the system. Asynchronous update is therefore more preferable.

Even though much work has been done on power control issues in wireless networks, we still lack a systematic framework for maximizing a system utility through power control alone (or joint power control and scheduling). In this thesis, we focus on not only devising algorithms that can find the global optimal power allocation to the pure power control problem either in a centralized fashion or in a distributed fashion, but also proposing algorithms that can obtain the global optimal solution to the joint power control and scheduling problem. The main contributions are listed in the next section.

1.3 Overview of Contributions

We address the following specific questions in this thesis:

1. Given a single-carrier wireless network, how much power should each link transmit to maximize a system utility while satisfying individual user's QoS requirements? How can we efficiently solve the SUM problem in a

centralized fashion?

2. Given a multi-carrier wireless network, how much power on each subcarrier should each link transmit to maximize a system-wide utility while satisfying individual user's QoS requirements? How can we efficiently obtain the global optimal power allocation in a centralize fashion?

3. Given a single-carrier wireless network, which links should transmit together and which should not, as well as how much power and for how long should each link transmit to maximize the overall system utility while satisfying individual user's QoS requirements? How can we efficiently obtain a optimal joint power control and scheduling strategy?

4. Given a single-carrier wireless network, how can we maximize a system-wide utility through power control in a distributed fashion?

This thesis is structured as follows.

- We propose the MAPEL algorithm in Chapter 2, which is the first algorithm in the literature that can achieve the global optimal solution of the SUM problem in the general SINR regime. There are three key observations that enable MAPEL to efficiently solve the non-convex optimization problem. First, the objective function of SUM is monotonically increasing in (1+SINR). Second, the feasible set of the corresponding equivalent transformed problem, although may be not convex, is always *normal*. These two observations imply that the SUM problem can be transformed into a monotonic optimization (MO) problem [18]. Third, the monotonicity of the problem implies that the optimal solution always occurs at the upper boundary of the feasible (1+SINR) region. This, together with the normality of the feasible region, allows us to construct a sequence of polyblocks to approximate the (1+SINR) region boundary with an increasing

level of accuracy. Given an arbitrary small and finite error tolerance level, MAPEL is guaranteed to find one global $\varepsilon -$ optimal solution of the SUM problem within finite amount of time. On the other hand, a global optimal solution is guaranteed if we do not enforce a finite running time. A flexible tradeoff between performance and convergence time can be achieved by tuning the approximation factor.

MAPEL provides an important benchmark for all algorithms that are designed to tackle the SUM problem in single-carrier networks, whether it is existing or to be proposed, centralized or distributed, optimal or heuristic. In this thesis, we show how such benchmark is useful in evaluating the performance of two state-of-art centralized and distributed algorithms ([12, 13]) in this area.

Finally, we note that some work has been done (e.g., [4, 9]) on the problem of maximizing the minimum achievable SINR of each link in wireless networks. This is motivated partially by fair allocation among various users in the network. All existing algorithms for solving this problem are centralized. Interestingly, our MAPEL algorithm can be easily adapted to solve the same max-min optimization problem in a different and also centralized manner. We will briefly discuss this extension as well.

- We then endeavor to maximize a system utility through power control in multi-carrier wireless networks. Based on the ideas behind MAPEL, we propose M-MAPEL algorithm in Chapter 3 to achieve the global optimal power allocation of the SUM problem in multi-carrier networks. Similar to MAPEL, M-MAPEL can strike a tradeoff between performance and convergence time by flexibly tuning the approximation factor. The main benefit of M-MAPEL is to provide a benchmark for other (mostly heuristic) algorithms that are designed to tackle the power control in multi-

carrier networks, whether it is existing or to be proposed, centralized or distributed. In Chapter 3, we show how such benchmark is useful in evaluating the performance of the state-of-art algorithm ([42]) in this area. Besides, we show through M-MAPEL that the power control that allows frequency reuse can significantly improve the network performance over the traditional power control with exclusive subcarrier allocation in multi-carrier networks.

- In Chapter 4, we start with formulating the non-convex joint power control and scheduling problem into a MO problem that is amenable to an efficient global optimization algorithm. There are three key observations that lead to such a formulation. First, by a standard convexity argument, the joint power control and scheduling problem is equivalent to seeking a piecewise constant power allocation that has $M + 1$ (M: the number of links) degrees of freedom in the time domain. Second, the objective function of the optimization problem is monotonically increasing in both SINR and scheduling period. Last, the feasible set of the equivalent transformed problem, although may be not convex, is always normal. These properties allow us to bypass the non-convexity of the problem and exploit monotonicity to devise efficient solution algorithms.

  Next, we propose an algorithm, referred to as S-MAPEL, to obtain the global optimal solution of the joint power control and scheduling problem. The MAPEL algorithm proposed for solving pure power control problems cannot be directly applied to the joint power control and scheduling problem due to convergence issues that will be discussed later. Being a non-trivial extension of MAPEL, S-MAPEL guarantees to converge to the global optimal solution even for non-concave system utilities. By tuning a small error tolerance, we can engineer a flexible tradeoff between

system performance and convergence time.

To further reduce the computational complexity, an accelerated algorithm, referred to as A-S-MAPEL, is proposed based on the inherent symmetry of the optimal solutions. Our results show that A-S-MAPEL greatly reduces the convergence time, while yielding a negligible performance degradation compared with S-MAPEL.

Finally, we use the optimal and near optimal solutions obtained by the proposed algorithms as a benchmark to evaluate the performance of existing heuristics. An interesting conclusion is that on-off power control achieves close-to-optimal performance when it is integrated with scheduling. On the other hand, without scheduling, on-off power control can be far from optimal. This implies that scheduling is an indispensable component in wireless system designs, if off-the-shelf wireless devices that do not offer the freedom to adjust transmission power are to be used.

- In Chapter 5, we propose a SEER (*aSynchronous distributEd powEr contRol*) algorithm to achieve the global optimal power allocation in a distributed fashion. The key idea of SEER largely comes from *Gibbs Sampling*, which is a well-studied optimization algorithm in fields such as statistical physics and image processing [50, 51]. SEER has four distinctions from previous work. First, SEER maximizes the system utility function by exploring the function's entire surface, and thus it has a provable convergence to the optimal strategy that picks global optimal solutions with probability 1 for any system utility function. In particular, the system utility function is allowed to be *non-concave, discontinuous*, and *non-monotonic*. Second, SEER achieves the optimal power control noticeably faster than the centralized algorithm MAPEL we proposed earlier. As such, the algorithm can efficiently handle large-scale wireless networks, as shown in our simulations.

Third, SEER requires only limited message passing among links and small memory storage at each link. Last, SEER allows asynchronous power update and message passing.

- In Chapter 6, we draw conclusions and discuss possible extensions of the work covered in this thesis

Chapter 2

MAPEL: Achieving Global Optimality for Non-convex Wireless Power Control Problems

In this chapter, we start with a single-hop single-carrier wireless ad hoc network where simultaneous transmissions interfere with each other and limit the wireless network performance due to the broadcast nature of wireless communications. Transmission power control has attracted extensive research interests as one important interference mitigation technique. However, maximizing a system utility through power control has been a long standing open problem in interference-limited wireless networks. The complicated coupling between the mutual interferences of links gives rise to a non-convex optimization problem. Previous work has considered the SUM problem in the high SINR regime or for some special utility functions that are log-concave in the feasible SINR region, in which case the problem can be approximated and transformed into a convex optimization problem through proper change of variables. In the general SINR regime, however, the approximation and transformation approach

does not work for the utility functions that are not log-concave in the feasible SINR region. To address this issue, we propose the MAPEL algorithm that achieves the global optimal power allocation of the SUM problem in the general SINR regime, through exploiting the hidden monotonicity of the SUM problem. MAPEL provides an important benchmark for performance evaluation of other heuristic algorithms targeting the same problem. With the help of MAPEL, we evaluate the performance of several existing algorithms through extensive simulations.

The remainder of this chapter is organized as follows. System model is discussed in Section 2.1. In Section 2.2, we transform the SUM problem into a MO problem. Some properties of the feasible region in the MO problem are also discussed. The MAPEL algorithm is proposed and analyzed in Section 2.3. A brief discussion on the extension to the max-min SINR problem is also provided. In Section 2.4, we evaluate the performance of MAPEL through several simulations. With the benchmark established by MAPEL, we evaluate the performance of two existing algorithms in Section 2.5. Section 2.6 concludes this chapter. The proofs for the global convergence of MAPEL and that MAPEL obtains an ϵ-optimal solution are discussed in Section 2.7.

2.1 System Model and Problem Formulation

We consider a wireless ad hoc network with a set $\mathcal{M} = \{1, \cdots, M\}$ of *distinct* links[1]. Each link includes a transmitter node T_i and a receiver node R_i. The channel gain between node T_i and node R_j is denoted by G_{ij}, which is determined by various factors such as path loss, shadowing and fading effects. The

[1]For example, this could represent a network snapshot under a particular schedule of transmissions determined by an underlying routing and MAC protocol.

complete channel matrix is denoted by $\boldsymbol{G} = [G_{ij}]$. Let p_i denote the transmission power of link i (i.e., from node T_i), and n_i denote the received noise of link i (i.e., measured at node R_i). The received SINR of link i is

$$\gamma_i(\boldsymbol{p}) = \frac{G_{ii}p_i}{\sum\limits_{j\neq i} G_{ji}p_j + n_i}, \tag{2.1.1}$$

and the data rate r_i calculated based on the Shannon capacity formula is $\log_2(1+\gamma_i(\boldsymbol{p}))$ [2]. To simplify notations, we use $\boldsymbol{p} = (p_i, \forall i \in \mathcal{M})$, $\boldsymbol{P}^{\max} = (P_i^{\max}, \forall i \in \mathcal{M})$ and $\gamma(\boldsymbol{p}) = (\gamma_i(\boldsymbol{p}), \forall i \in \mathcal{M})$ to respectively represent the transmission power vector, the maximum transmission power vector and achieved SINR vector of all links.

We want to find the optimal power allocation $\boldsymbol{p}^*$ that maximizes a system utility subject to individual data rate constraints. Mathematically, we want to solve the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & U(\boldsymbol{r}) \\ \text{subject to} \quad & r_i \geq r_i^{\min},\ \forall i \in \mathcal{M}, \\ \text{variables} \quad & 0 \leq p_i \leq P_i^{\max},\ \forall i \in \mathcal{M}, \end{aligned} \tag{P2.1}$$

where $r_i^{\min} \geq 0$ is the minimum data rate requirement of link i (including the special case of $r_i^{\min} = 0$, i.e., no rate constraint), and $U(\boldsymbol{r})$ represents the system-wide utility, with $\boldsymbol{r}$ being the vector of r_i. In general, $U(\boldsymbol{r})$ is assumed to be additive across links, i.e., $U(\boldsymbol{r}) = \sum\limits_{i=1}^{M} U_i(r_i)$, with $U_i(r_i)$ being the utility of link i. Notably, most previous work assumes that $U_i(r_i)$ is a concave and increasing function [11–14]. In practice, however, $U_i(r_i)$ is not necessarily concave. Typically, $U_i(r_i)$ is either a concave non-decreasing function for elastic traffic or a

[2]To better model the achievable rates in a practical system, we can re-normalize $\gamma_i(\boldsymbol{p})$ by $\Gamma_i\gamma_i(\boldsymbol{p})$, where $\Gamma_i \in [0,1]$ represents the SINR gap indicating the difference between the SINR needed to achieve a certain data rate for a practical modulation and coding scheme and the theoretical limit. Such modification, however, does not change the analysis in this chapter.

non-concave non-decreasing function for delay-sensitive traffic. In our work, we do not impose the concavity assumption on $U_i(r_i)$. The only assumption is $U_i(r_i)$ being increasing, which is a valid assumption for most applications. By appropriately choosing the utility function $U_i(r_i)$, we can strike different balances between spectrum efficiency and fairness. For elastic traffics, a general form of the utility function $U_i(r_i)$ is the generalized α-fairness utility function, i.e.,

$$U_i(r_i) = \begin{cases} \log(r_i) & \text{if } \alpha = 1, \\ \frac{(r_i)^{1-\alpha}}{1-\alpha} & \text{if } \alpha \geq 0 \text{ and } \alpha \neq 1. \end{cases} \tag{2.1.2}$$

Maximizing the objective function as given in (2.1.2) will correspond to *maximizing throughput* as $\alpha \to 0$, *proportional fairness* as $\alpha \to 1$, and *max-min fairness* as $\alpha \to \infty$. This implies that increasing α leads to fairer allocation. For delay-sensitive traffics, sigmoidal utility functions $U_i(r_i)$ as given in Fig. 2.1 capture the "happiness" of links. In general, such utility function in the standard form is expressed as:

$$U_i(r_i) = \frac{1}{1 + e^{-a_i(r_i - b_i)}}, \tag{2.1.3}$$

where $\{a_i, b_i\}$ are constant positive integers. From Fig. 2.1, it can be seen that b_i can be regarded as a kind of threshold such that $U_i(r_i)$ is concave when r_i is beyond b_i, and convex otherwise. On the other hand, a_i can be regarded as an indication of slope such that the larger a_i, the steeper the curve of $U_i(r_i)$.

Notice that if $r_i^{\min}$'s are too large, there may not exist a feasible solution to Problem (P2.1). Therefore, we have to check the feasibility of Problem (P2.1) before proceeding further. For a user i, its received SINR value needs to be at least $\gamma_i^{\min} = 2^{r_i^{\min}} - 1$ in order to satisfy its minimum rate requirement. Con-

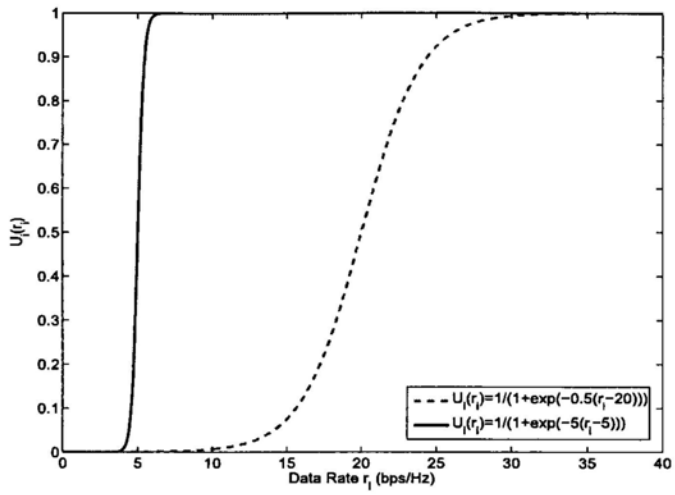


Figure 2.1: An example of sigmoidal function.

sider the following matrix $\boldsymbol{B}$ with

$$B_{ij} = \begin{cases} 0, & i = j \\ \frac{\gamma_i^{\min} G_{ji}}{G_{ii}}, & i \neq j. \end{cases}$$

According to Theorem 2.1 in [1], if the maximum eigenvalue of $\boldsymbol{B}$ is larger than 1, then there is no feasible solution to Problem (P2.1). Otherwise, we can find a power allocation $\hat{\boldsymbol{p}}$ as follows:

$$\hat{\boldsymbol{p}} = (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{u}, \tag{2.1.4}$$

where $\boldsymbol{I}$ is the $M \times M$ identity matrix and $\boldsymbol{u}$ is a $M \times 1$ vector with elements

$$u_i = \frac{\gamma_i^{\min} n_i}{G_{ii}}.$$

By Theorem 2.2 in [1], Problem (P2.1) is feasible if and only if the components of $\hat{\boldsymbol{p}}$ satisfies $0 \leq \hat{p}_i \leq P_i^{\max}$ for all i. Therefore, the procedure of checking the feasibility of Problem (P2.1) is as follows:

It has been shown that (e.g., [11–14]) Problem (P2.1) is a non-convex optimization problem in terms of the transmission power $\boldsymbol{p}$. Thus, it is difficult to

Procedure 1 Feasibility check of rate constraints $r_i^{\min}$'s in Problem (P2.1)

1: Transform minimum data constraints into minimum SINR constraints through $\gamma_i^{\min} = 2^{r_i^{\min}} - 1$ for all i.

2: Compute the maximum eigenvalue of matrix $\boldsymbol{B}$ and check if it is smaller than 1. If not, $r_i^{\min}$'s are infeasible. Otherwise, go to step 3.

3: Compute the power allocation $\hat{\boldsymbol{p}}$ according to (2.1.4) and check if it satisfies $0 \leq \hat{p}_i \leq P_i^{\max}$ for all i. If so, $r_i^{\min}$'s are feasible. Otherwise, $r_i^{\min}$'s are infeasible.

find a global optimal solution efficiently even in a centralized fashion. In Section 2.2, we will show Problem (P2.1) can be transformed to a Monotonic Optimization (MO) problem, which can then be solved efficiently by the MAPEL algorithm presented in Section 2.3.

2.2 Power Control as Monotonic Optimization (MO)

Due to the non-convex nature of Problem (P2.1), it is impossible to find the optimal solution based on the theory of convex optimization [47]. Fortunately, it is recently found in operations research that monotonicity is another important property besides convexity that can be exploited to efficiently solve an optimization problem. By exploiting the hidden monotonicity of optimization problems, we can bypass the non-convexity issue and obtain the global optimal solution efficiently. In this section, we first introduce the definition of MO, and then show how Problem (P2.1) can be transformed into a MO problem. We further discuss several key properties of the reformulation that are critical for the design of the MAPEL algorithm that solves the optimization problem.

Definition 2.1 (Box). *Given any vector $\boldsymbol{v} \in \mathbb{R}_+^M$, the hyper rectangle $[\boldsymbol{0}, \boldsymbol{v}] = \{\boldsymbol{x} | \boldsymbol{0} \preceq$*

$\boldsymbol{x} \preceq \boldsymbol{v}\}$ is referred to as a box with vertex $\boldsymbol{v}$.

Definition 2.2 (Normal). *An infinite set $\mathcal{F} \subset \mathbb{R}_+^M$ is said to be normal if for any element $\boldsymbol{v} \in \mathcal{F}$, the box $[\mathbf{0}, \boldsymbol{v}] \subset \mathcal{F}$.*

Proposition 2.1. *The intersection and the union of normal sets are still normal sets.*

Remark 2.1. *A box is normal.*

Definition 2.3 (MO). *An optimization problem belongs to the class of MO if it can be represented by the following formulation:*

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & \Phi(\boldsymbol{x}) \\ \text{variables} \quad & \boldsymbol{x} \in \mathcal{H}, \end{aligned} \tag{P_M}$$

where set $\mathcal{H} \subset \mathbb{R}_+^N$ is a nonempty normal closed set and function $\Phi(\boldsymbol{x})$ is an increasing function on $\mathbb{R}_+^N$ [18].

Definition 2.4 (Upper boundary). *A point $\boldsymbol{y} \in \mathbb{R}_+^M$ is an upper boundary point of a bounded normal set $\mathcal{F}$ if $\boldsymbol{y} \in \mathcal{F}$ while $\mathcal{F}_{\boldsymbol{y}} \subset \mathbb{R}_+^M \setminus \mathcal{F}$. The set of upper boundary points of $\mathcal{F}$ is the upper boundary of $\mathcal{F}$.*

Proposition 2.2. *The optimal solution of the MO Problem P_M (if it exists) is attained on the upper boundary of $\mathcal{H}$ [18].*

The proof of Proposition 2.2 is similar to that of Proposition 7 in [18], and thus is omitted here. Interested readers are referred to the Proposition 7 in [18].

Now we are ready to reformulate Problem (P2.1) into a MO problem.

Let $\boldsymbol{z}$ be vector $(z_1, \cdots, z_M)$. Since the function $U_i(r_i)$ is increasing, it is easy to see that the function $\Phi(\boldsymbol{z}) = \sum_{i=1}^{M} U_i(\log_2(z_i))$ is an increasing function on $\mathbb{R}_+^M$. That is, for any two vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ such that $\boldsymbol{z}_1$ is component-wise larger than or equal to $\boldsymbol{z}_2$ (denoted by $\boldsymbol{z}_1 \succeq \boldsymbol{z}_2$), we have $\Phi(\boldsymbol{z}_1) \geq \Phi(\boldsymbol{z}_2)$. We further note

that $1+\gamma_i(\boldsymbol{p})$ is positive for all i due to the existence of positive noise power n_i. Based on these observations, Problem (P2.1) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \Phi(\boldsymbol{z}) = \sum_{i=1}^{M} U_i(\log_2(z_i)) \\ \text{variables} \quad & \boldsymbol{z} \in \mathcal{G}, \end{aligned} \tag{P2.2}$$

where the feasible set

$$\mathcal{G} = \{\boldsymbol{z} | 0 \leq z_i \leq 1+\gamma_i(\boldsymbol{p}), \forall i \in \mathcal{M}, \boldsymbol{p} \in \mathcal{P}\}. \tag{2.2.1}$$

with

$$\mathcal{P} = \{\boldsymbol{p} | 0 \leq p_i \leq P_i^{\max}, \frac{f_i(\boldsymbol{p})}{g_i(\boldsymbol{p})} \geq 2^{r_i^{\min}}, \forall i \in \mathcal{M}\}, \tag{2.2.2}$$

Since $\Phi(\boldsymbol{z})$ is an increasing function in $\boldsymbol{z}$, the optimal solution to Problem (P2.2), denoted by $\boldsymbol{z}^*$, must occur at places where $z_i = 1+\gamma_i(\boldsymbol{p})$ for all i. If we can find a power allocation $\boldsymbol{p}^*$ corresponding to the optimal solution $\boldsymbol{z}^*$ such that $z_i^* = \gamma_i(\boldsymbol{p}^*)$ for all i, then such $\boldsymbol{p}^*$ is clearly the optimal solution to Problem (P2.1). Finding such $\boldsymbol{p}^*$ requires solving M linear equations $z_i^* g_i(\boldsymbol{p}^*) - f_i(\boldsymbol{p}^*) = 0$ with M variables $p_1^*, \cdots, p_M^*$, where $f_i(\boldsymbol{p}^*) = G_{ii}p_i^* + \sum_{j \neq i} G_{ji}p_j^* + n_i$ and $g_i(\boldsymbol{p}^*) = \sum_{j \neq i} G_{ji}p_j^* + n_i$. As the coefficients of $f_i(\boldsymbol{p}^*)$ and $g_i(\boldsymbol{p}^*)$ consist of random channel gains G_{ij}'s, we can show with probability 1 that the M equations are linearly independent, implying there is a unique solution $\boldsymbol{p}^*$. Hence, Problems (P2.1) and (P2.2) are both equivalent with each other. *We will focus on how to solve Problem (P2.2) efficiently in the rest of this chapter.*

Before attempting to solve Problem (P2.2), it is critical to understand several important properties of the feasible set $\mathcal{G}$ in (2.2.1).

According to the definition of box, the feasible set $\mathcal{G}$ can be characterized as a union of infinite number of boxes with vertices of all boxes belonging to the set $\{\boldsymbol{c} | c_i = \frac{f_i(\boldsymbol{p})}{g_i(\boldsymbol{p})}, \forall i \in \mathcal{M}, \boldsymbol{p} \in \mathcal{P}\}$. Each element in this set is determined by a power vector $\boldsymbol{p}$ that is feasible in Problem (P2.1). Since the feasible set $\mathcal{G}$ of

Problem (P2.2) is the union of infinite number of boxes, it is a normal set by Proposition 2.1 and Remark 2.1. This, together with $\Phi(z)$ being an increasing function in z, implies that Problem (P2.2) is a MO problem. Therefore, by Proposition 2.2, the optimal solution to Problem (P2.2), must occur at the upper boundary of set $\mathcal{G}$.

Fig. 2.2 illustrates one possible example of the shape of $\mathcal{G}$ in a 2-link network. Note that $\mathcal{G}$ is in general a non-convex set. However, this work shows that *convexity of the feasible set is not important in obtaining the global optimal solution.* It is the monotonicity of the objective function as well as the normality of the feasible set in the reformulated problem (P2.2) that facilitates efficient calculation of the global optimal solution.

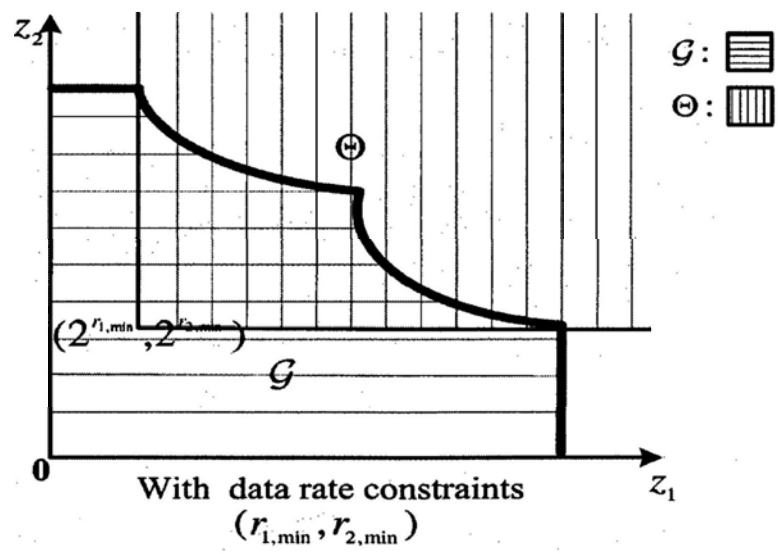


Figure 2.2: Shapes of $\mathcal{G}$ and Θ for a two-link network

Before leaving this section, note that $\frac{f_i(p)}{g_i(p)}$ is lower bounded by $2^{r_i^{\min}}$ for $p \in \mathcal{P}$. Consequently, the optimal solution z^* to Problem (P2.2), which occurs only at places where $z_i = \frac{f_i(p)}{g_i(p)}$ for all i, is also lower bounded by $(2^{r_i^{\min}}, \forall i \in \mathcal{M})$. In other words, the optimal solution z^* must reside in the set $\mathcal{G} \cap \Theta$, where $\Theta = \{z | z_i \geq 2^{r_i^{\min}}, \forall i \in \mathcal{M}\}$.

2.3 The MAPEL Algorithm

In this section, we propose a novel algorithm, MAPEL, to solve Problem (P2.2) based on the special characteristics of MO. The key idea of MAPEL largely comes from the recent advances in global optimization including monotonic optimization and fractional programming [45, 46]. Some mathematical preliminaries will be introduced first before we present the algorithm.

2.3.1 Related Mathematical Preliminaries

Definition 2.5 (Polyblock). *Given any finite set $\mathcal{T} \subset \mathbb{R}_+^M$ with elements $\boldsymbol{v}_i$, the union of all the boxes $[\mathbf{0}, \boldsymbol{v}_i]$ is a polyblock with vertex set $\mathcal{T}$.*

Definition 2.6 (Proper). *An element $\boldsymbol{v} \in \mathcal{T}$ is proper if there does not exist $\tilde{\boldsymbol{v}} \in \mathcal{T}$ such that $\boldsymbol{v} \neq \tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{v}} \succeq \boldsymbol{v}$. If every element $\boldsymbol{v} \in \mathcal{T}$ is proper, then the set $\mathcal{T}$ is a proper set.*

Proposition 2.3. *If $\Phi(\boldsymbol{v})$: $\mathbb{R}_+^N \rightarrow \mathbb{R}_+$ is an increasing function of $\boldsymbol{v}$, then the maximum of $\Phi(\boldsymbol{v})$ over a polyblock occurs at one proper vertex of this polyblock.*

Proof: Let $\boldsymbol{v}^*$ be a global optimal solution of $\Phi(\boldsymbol{v})$ over a polyblock $\mathcal{S}$. If $\boldsymbol{v}^*$ is not a proper vertex of $\mathcal{S}$, then $\tilde{\boldsymbol{v}} \succeq \boldsymbol{v}^*$ for some proper vertex $\tilde{\boldsymbol{v}} \neq \boldsymbol{v}^*$. Since $\Phi(\boldsymbol{v}^*) \leq \Phi(\tilde{\boldsymbol{v}})$ due to the increasing property of $\Phi(\boldsymbol{v})$, it follows that $\tilde{\boldsymbol{v}}$ is also a global optimal solution of $\Phi(\boldsymbol{v})$, which is a contradiction to $\boldsymbol{v}^*$ being a global optimal solution. Hence, Proposition 2.3 follows immediately. ■

Definition 2.7 (Projection). *Given any nonempty normal set $\mathcal{F} \subset \mathbb{R}_+^M$ and any $\boldsymbol{v} \in \mathbb{R}_+^M \setminus \{\mathbf{0}\}$, $\boldsymbol{\pi}^{\mathcal{F}}(\boldsymbol{v})$ is a projection of $\boldsymbol{v}$ on $\mathcal{F}$ if $\boldsymbol{\pi}^{\mathcal{F}}(\boldsymbol{v}) = \lambda \boldsymbol{v}$ with $\lambda = \max\{\alpha | \alpha \boldsymbol{v} \in \mathcal{F}\}$. In other words, $\boldsymbol{\pi}^{\mathcal{F}}(\boldsymbol{v})$ is the unique point where the halfline from $\mathbf{0}$ through $\boldsymbol{v}$ meets the upper boundary of $\mathcal{F}$.*

We illustrate the above concepts in Fig. 2.3(a). In Fig. 2.3(a), the rectangles $\boldsymbol{a0cv_1}$ [3] and $\boldsymbol{b0dv_2}$ represent boxes $[\mathbf{0}, \boldsymbol{v}_1]$ and $[\mathbf{0}, \boldsymbol{v}_2]$, respectively. $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the respective vertices of these two boxes. The area consisting of rectangles $\boldsymbol{a0cv_1}$ and $\boldsymbol{b0dv_2}$ represents polyblock $\mathcal{S} = [\mathbf{0}, \boldsymbol{v}_1] \cup [\mathbf{0}, \boldsymbol{v}_2]$ with proper vertex set $\mathcal{T} = \{\boldsymbol{v}_1, \boldsymbol{v}_2\}$. If we choose any point $\boldsymbol{v}_3 \in \mathcal{S}$, it is obvious that the rectangle $\boldsymbol{e0fv_3}$ belongs to polyblock $\mathcal{S}$, i.e., $[\mathbf{0}, \boldsymbol{v}_3] \subset \mathcal{S}$. Hence, polyblock $\mathcal{S}$ is said to be normal. Being the only intersection of the halfline from $\mathbf{0}$ through $\boldsymbol{v}_4$ and the upper boundary of $\mathcal{S}$, $\pi^{\mathcal{S}}(\boldsymbol{v}_4)$ is a projection of $\boldsymbol{v}_4$ on $\mathcal{S}$. Moreover, if $\Phi(\boldsymbol{v})$ is an increasing function on $\mathcal{S}$, then $\Phi(\boldsymbol{v}) \leq \max\{\Phi(\boldsymbol{v}_1), \Phi(\boldsymbol{v}_2)\}$ for all $\boldsymbol{v} \in \mathcal{S}$. In other words, the maximum of the increasing function $\Phi(\boldsymbol{v})$ occurs only at either $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$, a proper vertex of $\mathcal{S}$.

Now let us use the above concepts to illustrate how we can construct a series of polyblocks that approximate a set $\mathcal{F}$ with an increasing level of accuracy.

Proposition 2.4. *Let $\mathcal{S} \subset \mathbb{R}_+^M$ be a polyblock with a proper vertex set $\mathcal{T}$. Also let $\mathcal{F}$ be a nonempty normal closed set that is contained in $\mathcal{S}$, i.e., $\mathcal{F} \subset \mathcal{S} \subset \mathbb{R}_+^M$. For a given vertex $\boldsymbol{v}_i \in \mathcal{T}$, let $\mathcal{T}'$ be the set obtained from $\mathcal{T}$ by replacing the vertex $\boldsymbol{v}_i$ with M new vertices, $(\boldsymbol{v}_{i1}, \cdots, \boldsymbol{v}_{iM})$. Here the new vertex $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - (v_{i,j} - \pi_j^{\mathcal{F}}(\boldsymbol{v}_i))\boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is the jth unit vector of $\mathbb{R}_+^M$, $v_{i,j}$ is the jth element of the old vertex $\boldsymbol{v}_i$, and $\pi_j^{\mathcal{F}}(\boldsymbol{v}_i)$ is the jth element of the projection $\boldsymbol{\pi}^{\mathcal{F}}(\boldsymbol{v}_i)$. Note that some of the new vertices $(\boldsymbol{v}_{i1}, \cdots, \boldsymbol{v}_{iM})$ might not be proper. If we further remove all improper elements from set $\mathcal{T}'$ and obtain a new set $\mathcal{T}^*$, then the polyblock $\mathcal{S}^*$ with vertex set $\mathcal{T}^*$ satisfies $\mathcal{F} \subset \mathcal{S}^* \subset \mathcal{S}$. In this way, we have constructed a smaller polyblock $\mathcal{S}^*$ that still contains $\mathcal{F}$.*

The proof of Proposition 2.4 is similar to that of Proposition 3 in [45], and thus is omitted here. Interested readers are referred to Proposition 3 in [45].

[3] The rectangle is denoted using four letters representing its four vertices.

We use Fig. 2.3(b) to illustrate the above procedure. As shown in Fig. 2.3(b), given $\mathcal{F}$ and $\mathcal{S}$ such that $\mathcal{F} \subset \mathcal{S} \subset \mathbb{R}^2_+$, we can obtain a polyblock $\mathcal{S}^*$ with proper vertex set $\mathcal{T}^* = \{v_{11}, v_2\}$ satisfying $\mathcal{F} \subset \mathcal{S}^* \subset \mathcal{S}$. $\mathcal{T}^* = \{v_{11}, v_2\}$ is obtained by replacing v_1 in $\mathcal{T} = \{v_1, v_2\}$ with $v_{1i} = v_1 - (v_{1,i} - \pi_i^{\mathcal{F}}(v_1))e_i$, $i = 1, 2$, and then deleting the improper element v_{12} from $\mathcal{T}' = \{v_{11}, v_{12}, v_2\}$.

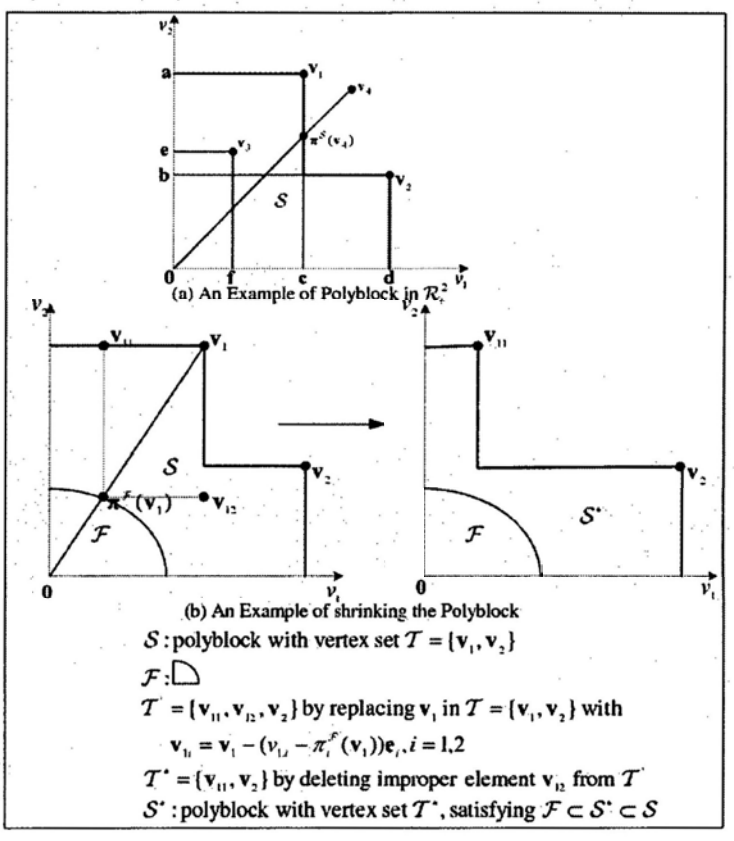


Figure 2.3: Illustration about related mathematical preliminaries for MAPEL algorithm

2.3.2 The MAPEL Algorithm

The MAPEL algorithm works as follows. We first check the feasibility of the minimum data rate requirements $r_i^{\min}$'s by Procedure 1. If the requirements $r_i^{\min}$'s are infeasible, there is no feasible power allocation and hence the algo-

rithm is terminated at once. Otherwise, we construct a polyblock $\mathcal{S}_1$ that contains the feasible set of Problem (P2.2), $\mathcal{G}$. Let $\mathcal{T}_1$ denote the proper vertex set of $\mathcal{S}_1$. By Proposition 2.3, the maximum of the objective function of Problem (P2.2) (i.e., $\Phi(z) = \sum_{i=1}^{M} U_i(\log_2(z_i))$ over set $\mathcal{S}_1$ occurs at some proper vertex z_1 of $\mathcal{S}_1$, i.e., $z_1 \in \mathcal{T}_1$. If z_1 happens to reside in $\mathcal{G}$ as well, then it solves Problem (P2.2) and $z^* = z_1$. Otherwise, based on Proposition 2.4 we can construct a smaller polyblock $\mathcal{S}_2 \subset \mathcal{S}_1$ that still contains $\mathcal{G}$ but excludes z_1. This is achieved by constructing the vertex set $\mathcal{T}_2$ by replacing z_1 in $\mathcal{T}_1$ with M new vertices $(z_{11}, \cdots, z_{1M})$, where $z_{1j} = z_1 - (z_{1,j} - \pi_j^{\mathcal{G}}(z_1))e_j$, and then removing improper vertices. We can repeat this procedure until an optimal solution is found. This leads to a sequence of polyblocks containing $\mathcal{G}$: $\mathcal{S}_1 \supset \mathcal{S}_2 \supset \cdots \supset \mathcal{G}$. Obviously, $\Phi(z_1) \geq \Phi(z_2) \geq \cdots \geq \Phi(z^*)$, where z_i is the optimal vertex that maximizes $\Phi(z)$ over set $\mathcal{S}_i$. The algorithm terminates at the kth iteration if $z_k \in \mathcal{G}$. For practical implementation, we say $z_k \in \mathcal{G}$ when $\max_i\{(z_{k,i} - \pi_i^{\mathcal{G}}(z_k))/z_{k,i}\} \leq \delta$ where $\delta > 0$ is a small positive number representing the error tolerance level.

We can further expedite the above process by selecting z_k from a smaller set $\mathcal{T}_k \cap \Theta$, where $\Theta = \{z | z_i \geq 2^{r_i^{\min}}, \forall i \in \mathcal{M}\}$. This will not affect the convergence or optimality of the algorithm since the optimal solution z^* is lower bounded by $(2^{r_i^{\min}}, \forall i \in \mathcal{M})$.

A critical step in constructing new polyblocks and checking the termination criterion is calculating the projection $\pi^{\mathcal{G}}(z_k)$. This is, however, by no means trivial, since the upper boundary of $\mathcal{G}$ is not explicitly known. In particular, $\pi^{\mathcal{G}}(z_k) = \lambda_k z_k$ is obtained by solving the following max-min problem for λ_k:

$$\begin{aligned}
\lambda_k &= \max\{\alpha | \alpha z_k \in \mathcal{G}\} \\
&= \max\{\alpha | \alpha \leq \min_{1 \leq i \leq M} \frac{f_i(\boldsymbol{p})}{z_{k,i} g_i(\boldsymbol{p})}, \ \boldsymbol{p} \in \mathcal{P}\} \\
&= \max_{\boldsymbol{p} \in \mathcal{P}} \min_{1 \leq i \leq M} \frac{f_i(\boldsymbol{p})}{z_{k,i} g_i(\boldsymbol{p})}.
\end{aligned} \tag{2.3.1}$$

This is a fractional programming problem by the definition in [46]. We solve this problem using the Dinkelbach-type algorithm in [46] with slight modifications. The details are shown in Algorithm 2.1[4]:

Algorithm 2.1 Max-Min Projection Algorithm (for finding $\pi^{\mathcal{G}}(z_k)$)

1: **Initialization**: Choose $p^{(0)} \in [\mathbf{0}, P^{\max}]$ and let $j = 0$.
2: **repeat**
3: Given $p^{(j)}$, solve $\lambda_k^{(j)} = \min_{1 \le i \le M} \frac{f_i(p^{(j)})}{z_{k,i} g_i(p^{(j)})}$.
4: Given $\lambda_k^{(j)}$, solve $p^{(j+1)} = \arg\max_{p \in \mathcal{P}} \min_{1 \le i \le M} (f_i(p) - \lambda_k^{(j)} z_{k,i} g_i(p))$.
5: $j = j + 1$.
6: **until** $\max_{p \in \mathcal{P}} \min_i (f_i(p) - \lambda_k^{(j-1)} z_{k,i} g_i(p)) \le 0$.
7: The projection is $\pi^{\mathcal{G}}(z_k) = \lambda_k^{(j-1)} z_k$.

Definition 2.8 (Q-super linear convergence [46]). *A sequence* $\{s_j, j = 1, 2, \cdots\} \in \mathbb{R}$ *with the limit* s_∞ *converges Q-super (quotient super) linearly if*

$$\lim_{j \to \infty} \left| \frac{s_{j+1} - s_\infty}{s_j - s_\infty} \right| = 0. \tag{2.3.2}$$

Theorem 2.1. *The sequence* $\{\lambda_k^{(j)}, j = 1, 2, \cdots\}$ *converges Q-super linearly to the optimal solution.*

Proof: Immediate from Theorem 8.7 in [46].

Having introduced the basic operations, we now formally present the MAPEL algorithm in Algorithm 2.2.

2.3.3 Global Convergence

The following theorem is proved in Section 2.7.1 and it shows that MAPEL is a global optimal algorithm of solving the SUM problem.

[4]In fact, each step 4 of Algorithm 2.1 is a linear programming in the convex power domain.

Algorithm 2.2 The MAPEL Algorithm

1: **Initialization**: Check the feasibility of minimum data rate requirements $r_i^{\min}$'s based on Procedure 1. If $r_i^{\min}$'s are infeasible, terminate the algorithm. Otherwise, choose the approximation factor $\delta > 0$, and let $k = 1$.

2: **repeat**

3: If $k = 1$, construct the initial polyblock $\mathcal{S}_1$ with vertex set $\mathcal{T}_1 = \{\boldsymbol{b}\}$, where the ith element of vector $\boldsymbol{b}$ is

$$b_i = \max_{\boldsymbol{p} \in [\boldsymbol{0},\, \boldsymbol{P}^{\max}]} \frac{f_i(\boldsymbol{p})}{g_i(\boldsymbol{p})} = 1 + \frac{G_{ii} P_i^{max}}{n_i}, \forall i \in \mathcal{M}.$$

It is clear that polyblock $\mathcal{S}_1$ is a box $[\boldsymbol{0}, \boldsymbol{b}]$ containing $\mathcal{G}$. If $k > 1$, construct a smaller polyblock $\mathcal{S}_k$ with vertex set $\mathcal{T}_k$ by replacing $\boldsymbol{z}_{k-1}$ in $\mathcal{T}_{k-1}$ with M new vertices $(\boldsymbol{z}_{k-1\,1}, \cdots, \boldsymbol{z}_{k-1\,M})$, where $\boldsymbol{z}_{k-1\,j} = \boldsymbol{z}_{k-1} - (z_{k-1,j} - \pi_j^{\mathcal{G}}(\boldsymbol{z}_{k-1}))\boldsymbol{e}_j$, and removing improper vertices.

4: Find $\boldsymbol{z}_k$ that maximizes the objective function of Problem (P2.2) over set $\mathcal{T}_k \cap \Theta$, i.e.,

$$\boldsymbol{z}_k = \operatorname{argmax}\{\Phi(\boldsymbol{z}) = \sum_{i=1}^{M} U_i(\log_2(z_i)) | \boldsymbol{z} \in \mathcal{T}_k \cap \Theta\}. \tag{2.3.3}$$

5: Find $\boldsymbol{\pi}^{\mathcal{G}}(\boldsymbol{z}_k)$ based on Algorithm 2.1.

6: $k = k + 1$.

7: **until** $\max_i \{(z_{k-1,i} - \pi_i^{\mathcal{G}}(\boldsymbol{z}_{k-1}))/z_{k-1,i}\} \leq \delta$.

8: Compute the optimal power allocation $\boldsymbol{p}^*$ (i.e., optimal solution to Problem (P2.1)) by solving $\pi_i^{\mathcal{G}}(\boldsymbol{z}_{k-1}) = \frac{f_i(\boldsymbol{p}^*)}{g_i(\boldsymbol{p}^*)}$ for all i.

Theorem 2.2. *The MAPEL algorithm globally converges to a global optimal solution of Problem (P2.2).*

Before leaving this subsection, note that although MAPEL is proved to converge to the global optimal solution, the convergence speed is still an open problem.

2.3.4 Trade-off between Performance and Convergence Time

The convergence time of MAPEL is infinite if the approximation factor $\delta = 0$. However, it can be easily shown that MAPEL always terminates with finite steps when $\delta > 0$ [45]. Next, we analyze the influence of the approximation factor δ on performance.

Definition 2.9 (ε-optimal solution). *Given an $\varepsilon \geq 0$, we say that a vector $\boldsymbol{y} \in \mathcal{G}$ is an ε-optimal solution of Problem (P2.2) if $\Phi(\boldsymbol{z}^*) \leq (1+\varepsilon)\Phi(\boldsymbol{y})$.*

Theorem 2.3. *For the weighted throughput maximization $U(\boldsymbol{r}) = \sum_{i=1}^{M} w_i r_i$ [5], the solution obtained by MAPEL is an ε-optimal solution with $\varepsilon \leq \frac{\delta}{1-\delta}$.*

The proof of the above theorem is relegated to Section 2.7.2.

Remark 2.2. *We note that $\frac{\delta}{1-\delta} \approx \delta$ when $\delta \ll 1$. Furthermore, $\frac{\delta}{1-\delta}$ is generally a conservative estimate of ε. In practice, the algorithm often yields an error that is much smaller than δ.*

An advantage of the MAPEL algorithm is that we can trade off performance for convergence time by tuning δ. The smaller δ, the longer the algorithm runs and the more accurate the solution is.

[5] $w_i > 0$ is the priority weight of link i. Without loss of generality, the weights w_i's are normalized so that $\sum_{i=1}^{M} w_i = 1$.

2.3.5 Extension to Max-min SINR Power Control

As discussed in the Introduction, some previous work on power control aimed at maximizing the minimum SINR of all links. Mathematically, they tried to solve the following problem

$$\max_{\boldsymbol{p}\in\mathcal{P}} \min_i \gamma_i(\boldsymbol{p}) = \max_{\boldsymbol{p}\in\mathcal{P}} \min_i \frac{G_{ii}p_i}{\sum_{j\neq i} G_{ji}p_j + n_i}. \tag{2.3.4}$$

Obviously, this is again a fractional programming problem by the definition in [46]. In fact, this formulation is similar to the one in described (2.3.1). Hence, the Dinkelbach-type algorithm (Algorithm 3.1) that is adopted to solve (2.3.1) can be easily extended to solve the max-min SINR problem in (2.3.4).

2.4 Performance Evaluation of MAPEL

To the best of our knowledge, maximizing the total weighted throughput through power control is a "representative" non-convex but monotonic power control problem. In this section, we therefore illustrate the effectiveness of the MAPEL algorithm applied to the WTM problem through several examples .

Example 1 (Performance and convergence time tradeoff through the approximation factor δ): We consider a four-link network where the links are randomly placed in a 10m-by-10m area. The resultant channel gain matrix is

$$\boldsymbol{G_1} = \begin{bmatrix} 0.4310 & 0.0002 & 0.2605 & 0.0039 \\ 0.0002 & 0.3018 & 0.0008 & 0.0054 \\ 0.0129 & 0.0005 & 0.4266 & 0.1007 \\ 0.0011 & 0.0031 & 0.0099 & 0.0634 \end{bmatrix}. \tag{2.4.1}$$

Assume that $\boldsymbol{P}^{\max}$=(0.7 0.8 0.9 1.0)mW, $n_i = 0.1\mu$W for all link i, and the priority weights $\boldsymbol{w}$=$(\frac{1}{6}\ \frac{1}{6}\ \frac{1}{3}\ \frac{1}{3})$. Also we do not consider minimum data rate constraints

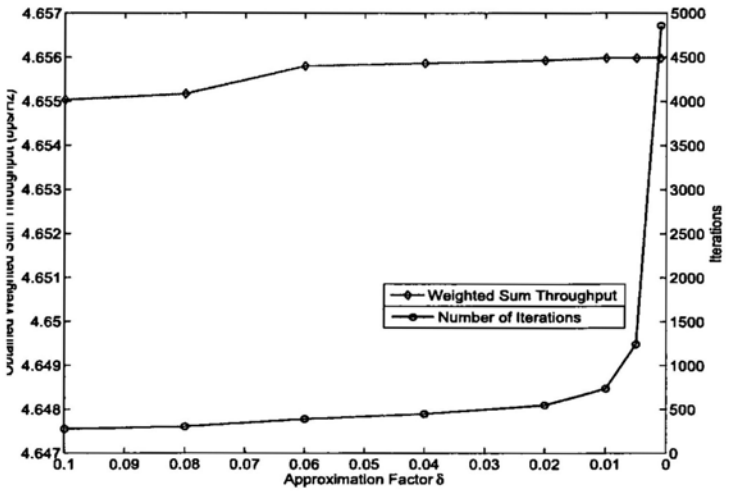


Figure 2.4: Obtained weighted sum throughput and number of iterations for different approximation factor δ

in this example.

In Fig. 2.4, we plot the optimal weighted-sum throughput obtained by MAPEL, together with the needed number of iterations versus δ. It is not surprising to see that the algorithm performance improves with a decreasing value δ, which has been predicted by Theorem 2.3. On the other hand, the total number of iterations increases when δ decreases, and the change is drastic when δ is close to 0. Moreover, the algorithm performance is not sensitive to the value of δ. For example, when $\delta = 0.1$, we achieve a weighted-sum throughput of 4.655bps/Hz that is only 0.025% away from the exact optimum. This illustrates that the performance bound obtained in Theorem 2.3 is quite loose, and the actual performance could be much better than the bound. It is also clear that parameter δ provides a tuning knob for achieving various trade-off between algorithm performance and convergence time.

Example 2 (Global optimal power allocation): MAPEL enables us to easily

characterize the global optimal solution[6] of the WTM problem for an arbitrary wireless network. This is not possible before without exhaustive search. We consider a different 4-link network in Fig. 2.5 as a simple illustrating example. The length of each link is 4m, while the distances between T_i and R_j for $i \neq j$, denoted by d_{ij}, are proportional to d. The four links have different channel gains: $G_{11} = 1$, $G_{22} = 0.75$, $G_{33} = 0.50$, $G_{44} = 0.25$. The priority weight of each link is equal. Meanwhile, $G_{ij} = d_{ij}^{-4}$, $\boldsymbol{P}^{\max}$=(0.7 0.8 0.9 1.0)mW, $n_i = 0.1\mu$W for all i. In Fig. 2.6, the optimal transmission power of each link is plotted against the topology parameter d. It can be seen that when the links are very close to each other, only the link with the largest channel gain (i.e., Link 1) is active with maximum transmission power $P_1^{\max}$, while all the other links keep silent. When d increases, a quantum jump in p_2 from 0 to $P_2^{\max}$ is observed[7]. As d further increases, Link 3 starts to transmit, followed by Link 4. In this particular example, it can be seen that when sum throughput is to be maximized, priority is always given to the link with a larger channel gain. Although the result is neither surprising nor general, this toy example illustrates the possibility of using MAPEL as a tool to investigate the characteristics of global optimal solutions to power control problems.

2.5 Providing Benchmark For Existing Power Control Algorithms

A key application of MAPEL is to provide performance benchmark for other centralized or distributed algorithms that have been (or to be proposed) to solve

[6]MAPEL will only find one of the possible many global optimal solutions, depending on the choice of initial conditions.

[7]In fact, if there are only two active links, they must both transmit at the maximum power.

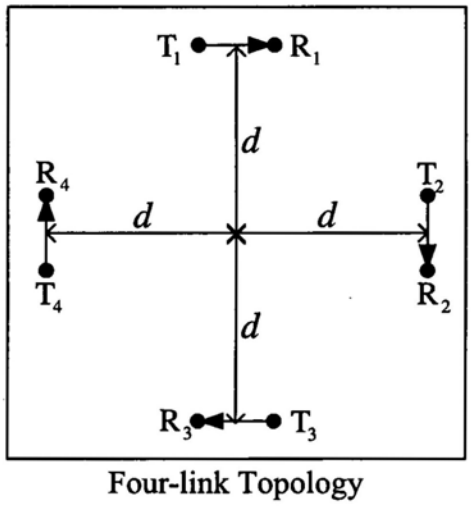


Figure 2.5: A network topology with four links

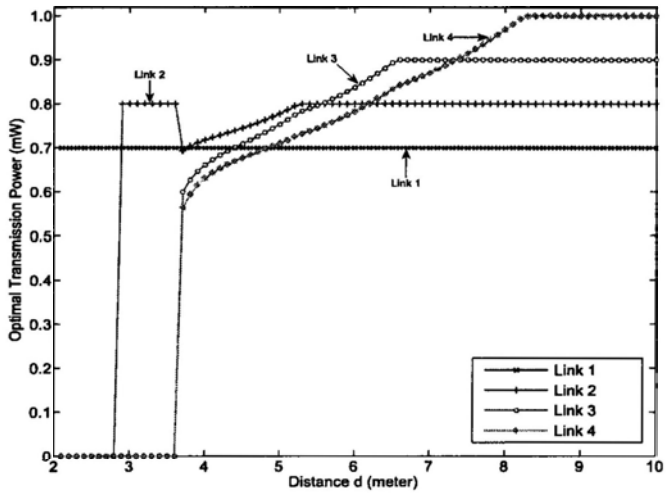


Figure 2.6: The relationship between optimal transmission power and distance d

the SUM problem. With MAPEL, we are able to give quantitative measurements of these algorithms' performance (e.g., the chances of achieving global optimal solution and the gap of sub-optimality) under a wide range of network scenarios (e.g, different network densities and topologies).

2.5.1 Review of Existing Power Control Algorithms

As we mentioned in Introduction, the current existing power control algorithms are essentially divided into two categories: centralized and distributed. Here we will review one "representative" algorithm from each category that represents the state-of-art in this area. Notice that the focus here is to show how MAPEL can be used to provide effective benchmark for the algorithms that tackle the same problem (i.e., Problem (P2.1)). Readers can choose your favorite algorithm to conduct the study. Our choice in this section may be biased, and the selected algorithms may not be "the best".

- **Centralized algorithm: Signomial Programming Condensation (SPC) Algorithm** [12]

SPC Algorithm is considered to be one of the best existing centralized algorithms for solving weighted throughput maximization (i.e., $U(\boldsymbol{r}) = \sum_{i=1}^{M} w_i r_i$ in Problem (P2.1)). It utilizes the fact that the WTM problem can be rewritten as minimizing a ratio between two posynomials (i.e., a SP):

$$\begin{aligned} \text{minimize} \quad & \prod_{i=1}^{M} \frac{g_i^{w_i}(\boldsymbol{p})}{f_i^{w_i}(\boldsymbol{p})} \\ \text{variables} \quad & \boldsymbol{p} \in \mathcal{P}. \end{aligned} \tag{2.5.1}$$

The key idea of SPC Algorithm is to improve the solution of Problem (2.5.1) through successive approximations until a KKT point is reached. During each step, the SP is approximated by a GP, which can be solved efficiently using a centralized interior point method.

- **Asynchronous Distributed Pricing (ADP) Algorithm** [13]

ADP Algorithm is a distributed algorithm that can be used to maximize the weighted total throughput without minimum data rate constraints. In ADP,

each link announces a price that reflects its sensitivity to the received interference, and updates its own transmission power based on the prices announced by other links. The price and power values need to be updated iteratively and asynchronously until a convergent point is found. To implement the updates, each link only needs to acquire limited information from the network. We observe that ADP algorithm converges very fast in our numerical experiments, mainly because no stepsize is used in the updates. Its theoretical convergence to the global optimal point, however, is difficult to prove in general.

2.5.2 Performance Study of SPC Algorithm and ADP Algorithm

In this subsection, we evaluate the performance of both algorithms through serval examples by utilizing the benchmark provided by MAPEL.

Example 3 (Probability of achieving global optimal solution): MAPEL always guarantees the global optimality of the WTM problem, while the SPC algorithm and the ADP algorithm fail to do so. Using the same 4-link network given in Example 1 (topology G_1), we simulate both algorithms from 500 random initializations and show the results in Fig. 2.7 and Fig. 2.8, respectively. Then we change the topology to G_2 with channel matrix illustrated in (2.5.2), and simulate both algorithms again in Fig. 2.9 and Fig. 2.10, respectively.

$$G_2 = \begin{bmatrix} 0.1476 & 0.0105 & 0.0018 & 0.0402 \\ 0.0034 & 0.1784 & 0.0013 & 0.2472 \\ 0.0014 & 0.0017 & 0.3164 & 0.0046 \\ 0.0048 & 0.4526 & 0.0012 & 0.6290 \end{bmatrix}. \tag{2.5.2}$$

Other system parameters are the same as in Example 1. The figures show that MAPEL always converges to the global optimal solution, regardless of the initial power allocation. On the other hand, the SPC algorithm and the ADP al-

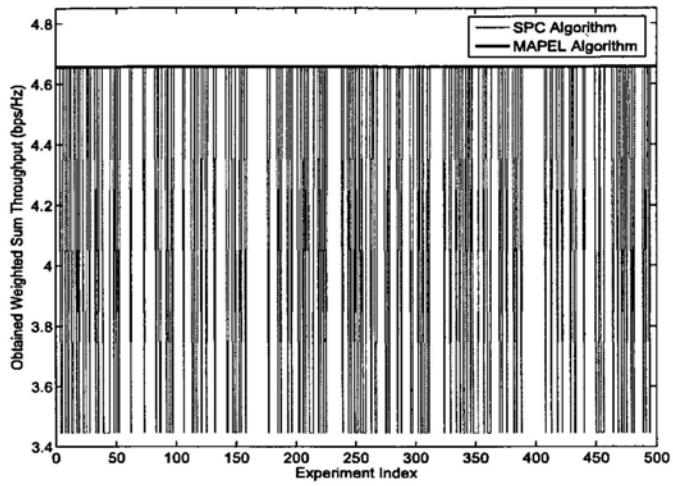


Figure 2.7: Maximal weighted sum throughput achieved by MAPEL algorithm as well as SPC algorithm for 500 different initial feasible power allocations in G_1 network

gorithm are trapped in local optimal solutions from time to time. For example, Fig. 2.7 and Fig. 2.8 show that SPC and ADP algorithms achieve the global optimal solution 70.8% and 62.6% of the time, respectively. However, Fig. 2.9 and Fig. 2.10 show that in a different topology SPC and ADP algorithms achieve the global optimal solution 96% and 93.6% of the time, respectively. In these four figures, we can find that the probability of achieving global optimal solution for SCP (or ADP) is sensitive to the network topology.

Example 4 (Average algorithm performance without minimum data rate constraints): In Fig. 2.11, we compare the average performance of the SPC algorithm, the ADP algorithm and the GP algorithm, with MAPEL under different network densities. Compared with SPC and ADP, GP [11] approximates and solves the WTM problem based on high-SINR assumption. For each fixed total number of links n, we place the links randomly in a 10m-by-10m area. The length of each link is uniformly distributed within [1m, 2m]. The priority weight of each link is equal. Meanwhile, we have $P_i^{\max}$=1mW, $n_i = 0.1\mu$W,

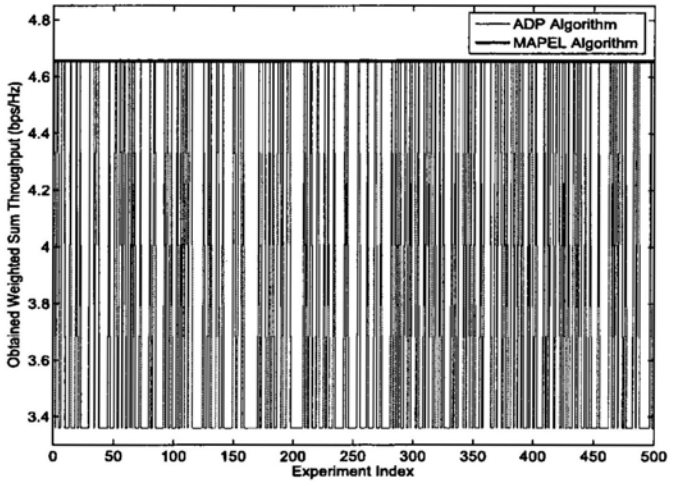


Figure 2.8: Maximal weighted sum throughput achieved by MAPEL algorithm as well as ADP algorithm for 500 different initial feasible power allocations in G_1 network

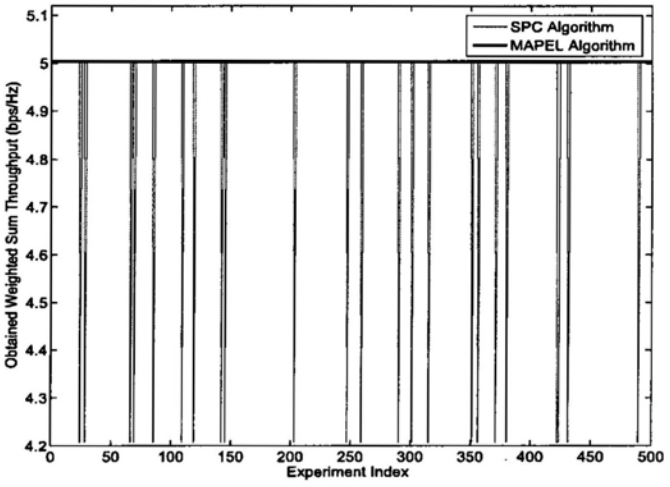


Figure 2.9: Maximal weighted sum throughput achieved by MAPEL algorithm as well as SPC algorithm for 500 different initial feasible power allocations in G_2 network

and initial power allocation is fixed at $P^{\max}/2$. We vary the total number of links n from 1 to 10. Each point is obtained by averaging over 500 different topologies of the same link density. On average, the performance loss of SPC

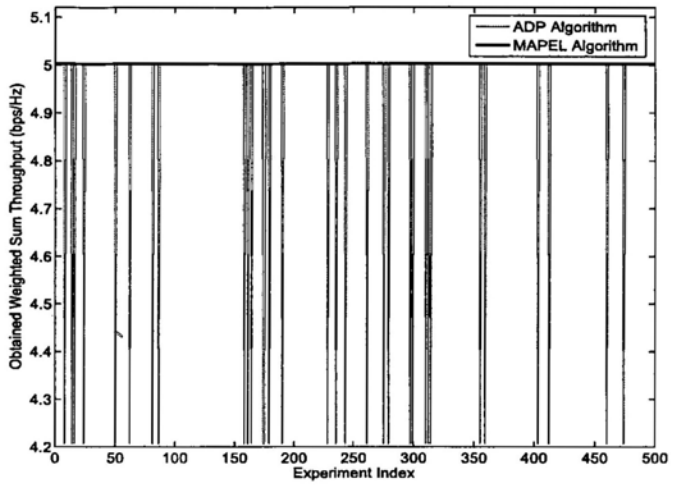


Figure 2.10: Maximal weighted sum throughput achieved by MAPEL algorithm as well as ADP algorithm for 500 different initial feasible power allocations in G_2 network

with respect to the global optimality is about 2%, thus is quite small. Notice that the performance loss of each particular realization might be smaller (e.g., 0% when reaching the global optimality) or much larger (when trapped in a local optimal). The average performance degradation of the ADP algorithm is about 10%, which implies that ADP is trapped in local optimum more often than SPC. Noticeably, the gap between SPC (or ADP) and the global optimum is not known before this work, as there is no previous algorithm that can guarantee the global optimal solution. This is in fact one of the key contributions of MAPEL. In addition, Fig. 2.11 shows that GP works reasonably well when the network density is low, where all (or most) links are active and some of them are indeed in the high SINR regime. However, the gap from the global optimum is much bigger when the network density becomes higher, where many links need to be silent in order to avoid heavy interferences to their neighbors.

Table 2.1 gives more detailed statistics about the performance of the two algorithms. As shown in Table I, SPC achieves the global optimality with a prob-

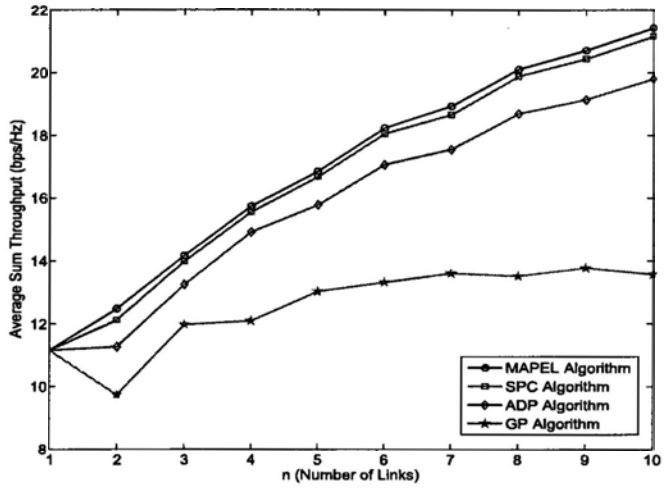


Figure 2.11: Average sum throughput of different algorithms in n-link networks

ability that is always larger than 65% with the number of links up to 10. In contrast, the probability of ADP achieving the global optimality can be very low, e.g., only 0.6% in 10-link networks. It suggests that the initial power allocation of $\boldsymbol{P}^{\max}/2$ is a good initial point for SPC, but may not for ADP. On the other hand, we find that SPC has a high-mean and low-variance average performance compared to the global optimality, which implies that SPC can achieve close-to-optimal performance with the initial power allocation of $\boldsymbol{P}^{\max}/2$ for most topologies. However, ADP has a low-mean and high-variance average performance, which implies that ADP maintains a large degradation for some topologies.

Example 5 (Average algorithm performance with minimum data rate constraints): We consider a series of 4-link networks with minimum data rate constraints on each link. The four links are randomly placed within a 10m-by-10m area, and the length of each link is uniformly distributed within the interval [1m, 2m]. $\boldsymbol{P}^{\max}$=(0.7 0.8 0.9 1.0)mW, $n_i = 0.1\mu$W for all i. Meanwhile, the priority weight of each link is equal. In Fig. 2.12, the performance of MAPEL, GP,

Table 2.1: Optimality of SPC Algorithm as well as ADP Algorithm

Number of Links	SPC Algorithm		
	Probability of achieving global optimality	Average performance	Coefficient of variation
2	69.8%	96.9%	7.22%
4	80.4%	98.7%	3.91%
6	77.2%	98.9%	3.13%
8	69.4%	98.8%	2.58%
10	65.6%	98.7%	2.81%
Number of Links	**ADP Algorithm**		
	Probability of achieving global optimality	Average performance	Coefficient of variation
2	50.6%	89.6%	17.8%
4	25.0%	94.3%	8.79%
6	6.0%	93.4%	7.89%
8	1.4%	92.7%	7.42%
10	0.6%	92.1%	8.18%

and SPC is plotted against the data rate constraint of each link. Each point for sum throughput on the curves is an average over 500 different topologies. We eliminate the topologies that are not feasible. Since ADP algorithm performs poorly in this case, we do not show its performance here.

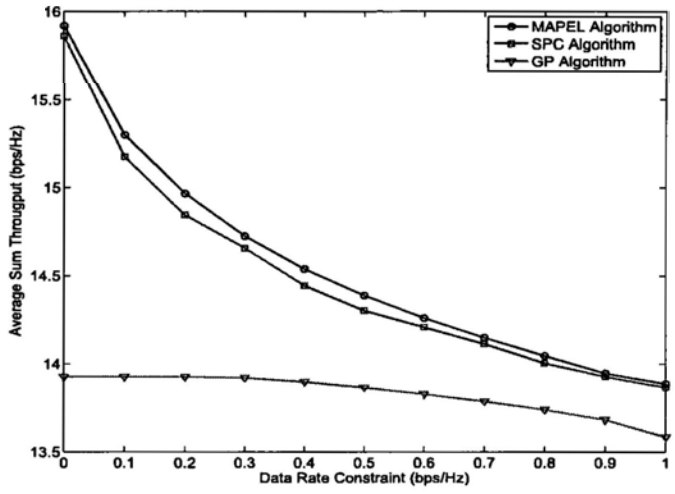


Figure 2.12: Average sum throughput of different algorithms versus the data rate constraint in 4-link networks

It is not surprising to see that the sum throughputs of all algorithms drop as the data rate constraints become more stringent. One interesting observation is that the gap between GP and MAPEL becomes smaller when the data rate constraints are high. This is due to the fact that links are forced to operate in the high SINR regime when a high data rate is to be ensured. The assumption made by GP becomes more reasonable in this case.

2.6 Summary

In this chapter, we proposed the MAPEL algorithm that solves the open problem of system utility maximization in general interference-limited wireless networks. The MAPEL algorithm is guaranteed to globally converge to an optimal solution despite the nonconvexity of the problem. The key idea behind the algorithm is to reformulate the SUM problem into a MO problem, and then construct a sequence of shrinking polyblocks that eventually closely approximate the upper boundary of the feasible region around the global optimum.

We have also established the tradeoff relationship between performance and convergence time of the MAPEL algorithm.

Being a centralized algorithm, MAPEL provides an important benchmark for performance evaluation of existing and newly proposed power control heuristics in this area. For example, by comparing with MAPEL through extensive simulations, we have gained deeper understanding of two state-of-the-art centralized and distributed power control algorithms: SPC algorithm and ADP algorithm. Simulations show that both algorithms achieve close-to-optimal average performance in the general SINR regime.

The MAPEL algorithm presented in this chapter is not the only way to efficiently obtain the global optimal solution. Variants of the algorithm can be developed to expedite the convergence and reduce the computational complexity. For example, it can be proved that the projection of a vertex of $\mathcal{S}$ on $\mathcal{G}$ must contain at least one element equal to $P_i^{\max}$. Such characteristics could be used to design a faster projection algorithm to replace Algorithm 2.1. Another possibility is to exploit the shape of the feasible region $\mathcal{G}$.

2.7 Appendix

2.7.1 Proof of Theorem 2.2

The MAPEL algorithm generates a sequence $\{z_k\}$ for $k = 1, 2, \cdots$. Each component is calculated as (2.3.3) for each newly constructed polyblock. We can find a subsequence $\{z_{k_n}\}$ within the sequence $\{z_k\}$ such that

$$\begin{aligned} z_{k_1} &= z_1 - (z_{1,i_0} - \pi_{i_0}^{\mathcal{G}}(z_1))e_{i_0}, \cdots, \\ z_{k_{n+1}} &= z_{k_n} - (z_{k_n,i_n} - \pi_{i_n}^{\mathcal{G}}(z_{k_n}))e_{i_n}, \end{aligned} \tag{2.7.1}$$

where $1 < k_1 < k_2 < \cdots < k_n < \cdots$. z_{k_n,i_n} denotes the i_nth element of vector z_{k_n}, where i_n is the only position in which $z_{k_{n+1}}$ differs from z_{k_n}. This subsequence can be thought as the "off-springs" of vertex z_1 through a series of projections, and they are not necessarily adjacent since there might be projections of other vertices happening in between. It can be shown that there is at least one such subsequence that has infinite length. With a slight abuse of notation, let $\{z_{k_n}, \forall n \geq 1\}$ denote one such subsequence. Since $\pi^{\mathcal{G}}(z_{k_n}) \preceq z_{k_n}$, (2.7.1) implies that $z_1 \succeq z_{k_1} \succeq \cdots \succeq z_{k_n} \succeq \cdots \succeq (2^{r_i^{\min}}, \forall i \in \mathcal{M})$. Hence, $\lim_{n\to\infty} \|z_{k_n} - z_{k_{n+1}}\| \to 0$. From (2.7.1) we know that z_{k_n} and $z_{k_{n+1}}$ only differ in the i_n's position, thus

$$\|z_{k_n} - z_{k_{n+1}}\| = z_{k_n,i_n} - z_{k_{n+1},i_n} = z_{k_n,i_n} - \pi_{i_n}^{\mathcal{G}}(z_{k_n}) \to 0 \text{ when } n \to \infty. \tag{2.7.2}$$

Since $\pi^{\mathcal{G}}(z_{k_n}) = \lambda_{k_n} z_{k_n}$ and $z_{k_n} \succeq (2^{r_i^{\min}}, \forall i \in \mathcal{M})$, (2.7.2) implies that $\lim_{n\to\infty} \lambda_{k_n} = 1$. That is,

$$\lim_{n\to\infty} z_{k_n} \to \pi^{\mathcal{G}}(z_{k_n}). \tag{2.7.3}$$

Eqn. (2.7.3) implies that the subsequence $\{z_{k_n}\}$ converges to the boundary of the feasible region $\mathcal{G}$. Since it is a maximizer over the set $\mathcal{S}_{k_n}$, it is also the global optimum of Problem (P2.2). Note that the MAPEL algorithm terminates once the optimal solution to Problem (P2.2) is found. Therefore, the convergence of the subsequence $\{z_{k_n}\}$ guarantees the convergence of the algorithm to the global optimal solution. ■

2.7.2 Proof of Theorem 2.3

MAPEL terminates when $\max_i \frac{z_{k,i} - \pi_i^{\mathcal{G}}(z_k)}{z_{k,i}} \leq \delta$. Consequently, together with $\sum_{i=1}^{M} w_i = 1$,

$$\Phi(z_k)(1-\delta) \leq \Phi(\pi^{\mathcal{G}}(z_k)) \leq \Phi(z^*) \leq \Phi(z_k)$$

leading to

$$\frac{\Phi(z_k) - \Phi(\pi^{\mathcal{G}}(z_k))}{\Phi(z_k)} \leq \delta.$$

Note that $\Phi(z^*) \leq \Phi(z_k)$ implies

$$\frac{\Phi(z^*) - \Phi(\pi^{\mathcal{G}}(z_k))}{\Phi(z_k)} \leq \delta.$$

Consequently,

$$\frac{\Phi(z^*) - \Phi(\pi^{\mathcal{G}}(z_k))}{\Phi(\pi^{\mathcal{G}}(z_k))} \leq \frac{\Phi(z_k)}{\Phi(\pi^{\mathcal{G}}(z_k))}\delta \leq \frac{\delta}{1-\delta},$$

which leads to the following inequality that proves Theorem 3:

$$\Phi(z^*) \leq \Phi(\pi^{\mathcal{G}}(z_k)) \left(1 + \frac{\delta}{1-\delta}\right).$$

■

Chapter 3

M-MAPEL: Monotonic Optimization for Non-concave Power Control in Multi-carrier Wireless Networks

In this chapter, we consider a multi-carrier wireless networks where more than one link can transmit non-zero power on the same subcarrier simultaneously. As discussed in Chapter 2, power control is still an important component for resource allocation and interference management in this context. However, due to the complicated coupling between the mutual interferences of links on each subcarrier, maximizing the overall system utility through power control is in general an NP-hard problem. Thus, it is difficult to solve despite its paramount importance. In this chapter, we present M-MAPEL based on the ideas of MAPEL, to find the optimal power allocation that maximizes the overall system utility while satisfying individual data rate constraints. The M-MAPEL algorithm is guaranteed to converge to a global optimal solution, as long as the utility func-

tion of each link is monotonically increasing with its data rate. Therefore, M-MAPEL can be used as an important benchmark for performance evaluation of other heuristic algorithms targeting the same problem. With the help of M-MAPEL, we evaluate the performance of a state-of-the-art algorithm through extensive simulations.

This chapter is organized as follows. System model and problem formulation are discussed in Section 3.1. In Section 3.2, we transform the power control optimization problem into a MO problem. Some properties of the feasible region of MO problem are also discussed. The M-MAPEL algorithm is proposed and analyzed in Section 3.3. In Section 3.4, we evaluate the performance of M-MAPEL through several simulations. Additionally, with the benchmark established by M-MAPEL, we evaluate the performance of a state-of-the-art algorithm. Section 3.5 summaries our conclusions. The global convergence of M-MAPEL is proved in Section 3.6.

3.1 System Model and Problem Formulation

We consider a multi-carrier ad hoc network with a set $\mathcal{M} = \{1, \cdots, M\}$ of *distinct* links. Each link consists of a transmitter node T_i and a receiver node R_i. We assume that each link can transmit information over L parallel subcarriers (subchannels). Let $\mathcal{L} = \{1, \cdots, L\}$ denote the subcarrier index set. The channel gain between nodes T_i and R_j over subcarrier l is denoted by $G_{l,ij}$, which is determined by various factors such as path loss, shadowing and fading effects. Thus, the complete channel matrix is denoted by $\boldsymbol{G} = [\boldsymbol{G}_1, \cdots, \boldsymbol{G}_L]$, where $\boldsymbol{G}_l = [G_{l,ij}]$ is the channel submatrix over subcarrier l. Let $p_{l,i}$ denote the transmission power of link i over subcarrier l. For link i, the transmission power over each subcarrier l is subject to the constraints $0 \leq p_{l,i} \leq P_{l,i}^{\max}$ and

$\sum_{l=1}^{L} p_{l,i} \leq P_i^{\max}$. Herein, $P_{l,i}^{\max}$ is the maximum power that can be transmitted over subcarrier l by link i and $P_i^{\max}$ is the upper bound on the total power transmitted by link i on all the L subcarriers. With loss of generality, assume that $P_i^{\max}$ is smaller than $\sum_{l=1}^{L} P_{l,i}^{\max}$, for otherwise the constraint $\sum_{l=1}^{L} p_{l,i} \leq P_i^{\max}$ can simply be removed. For notational convenience, we use $\boldsymbol{p}_l = (p_{l,i}, \forall i \in \mathcal{M})$ and $\boldsymbol{p} = (p_{1,1}, \cdots, p_{1,M}, \cdots, p_{L,1}, \cdots, p_{L,M})$ to represent the transmission power vector over subcarrier l and the transmission power vector over all subcarriers, respectively. Likewise, let $n_{l,i}$ denote the received noise of link i over subcarrier l. The received SINR of link i over subcarrier l is

$$\gamma_{l,i}(\boldsymbol{p}_l) = \frac{G_{l,ii} p_{l,i}}{\sum_{j \neq i} G_{l,ji} p_{l,j} + n_{l,i}}, \tag{3.1.1}$$

and the corresponding data rate $r_{l,i}(\boldsymbol{p}_l)$ calculated based on the Shannon capacity formula is $\log_2(1 + \gamma_{l,i}(\boldsymbol{p}_l))$.[1] Thus, the obtained total data rate on link i is denoted as

$$r_i = \sum_{l=1}^{L} r_{l,i}(\boldsymbol{p}_l) = \sum_{l=1}^{L} \log_2(1 + \gamma_{l,i}(\boldsymbol{p}_l)).$$

We endeavor to find the optimal power allocation $\boldsymbol{p}^*$ that maximizes the overall system utility subject to individual user's QoS requirements. To satisfy the QoS requirement, each link must maintain a minimum data rate $r_i^{\min}$. Math-

[1] As discussed in Chapter 2, to better model the achievable rates in a practical system, we can re-normalize $\gamma_{l,i}(\boldsymbol{p}_l)$ by $\Gamma_i \gamma_{l,i}(\boldsymbol{p}_l)$, where $\Gamma_i \in [0, 1]$ represents the SINR gap. Such modification, however, does not change the analysis in this chapter.

ematically, the optimal power control is formulated into the following form:

$$\begin{aligned} \text{maximize} \quad & U(\boldsymbol{r}) \\ \text{subject to} \quad & r_i = \sum_{l=1}^{L} \log_2(1+\gamma_{l,i}(\boldsymbol{p}_l)) \geq r_i^{\min} \\ & \sum_{l=1}^{L} p_{l,i} \leq P_i^{\max}, \ \forall i \in \mathcal{M} \\ \text{variables} \quad & 0 \leq p_{l,i} \leq P_{l,i}^{\max}, \forall i \in \mathcal{M}, \forall l \in \mathcal{L}, \end{aligned} \tag{P3.1}$$

where $U(\boldsymbol{r})$ represents the system-wide utility, with $\boldsymbol{r}$ being the vector of r_i. It is often assumed to be additive across links, i.e., $U(\boldsymbol{r}) = \sum_{i=1}^{M} U_i(r_i)$, with $U_i(r_i)$ being the utility of link i. Notably, most previous work assumes that $U_i(r_i)$ is a concave and increasing function [29–44]. In practice, however, $U_i(r_i)$ is not necessarily concave. Typically, $U_i(r_i)$ is either a concave non-decreasing function for elastic traffic or a non-concave non-decreasing function for delay-sensitive traffic. In our work, we do not impose the concavity assumption on $U_i(r_i)$. The only assumption is $U_i(r_i)$ being increasing, which is a valid assumption for most applications. By appropriately choosing the utility function $U_i(r_i)$, we can strike different balances between spectrum efficiency and fairness. Some possible utility functions have been introduced previously, and interested readers are referred to Chapter 2.

Due to the complicated coupling of the mutual interferences across links Problem (P3.1) is in general non-convex even if $U_i(r_i)$ is a concave function, let alone the cases with non-concave $U_i(r_i)$'s. Thus, it is difficult to find a global optimal solution $\boldsymbol{p}^*$ efficiently even in a centralized fashion.

Note that if $r_i^{\min}$ is too large, there may not exist a feasible solution to Problem (P3.1). However, the feasible region of Problem (P3.1) has a complicated shape, and thus the feasibility check procedure proposed in Chapter 2 cannot be applied to Problem (P3.1). Before proceeding further, we address this problem by

defining $\hat{U}_i(r_i)$ to be

$$\hat{U}_i(r_i) = \begin{cases} U_i(r_i) & \text{if } r_i \geq r_i^{\min}, \\ -\infty & \text{otherwise.} \end{cases} \tag{3.1.2}$$

Similar to $U_i(r_i)$, $\hat{U}_i(r_i)$ is a monotonically increasing function. With (3.1.2), Problem (P3.1) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{M} \hat{U}_i(r_i) \\ \text{subject to} \quad & \sum_{l=1}^{L} p_{l,i} \leq P_i^{\max}, \ \forall i \in \mathcal{M} \\ \text{variables} \quad & 0 \leq p_{l,i} \leq P_{l,i}^{\max}, \forall i \in \mathcal{M}, \forall l \in \mathcal{L}. \end{aligned} \tag{P3.2}$$

Notably, Problem (P3.1) is equivalent to Problem (P3.2) when Problem (P3.1) is feasible. On the other hand, the objective function of Problem (P3.2) is equal to $-\infty$ if and only if Problem (P3.1) is infeasible.

Although Problem (P3.2) is still a non-convex optimization Problem, we will show that Problem (P3.2) can be transformed to a MO problem, which can then be solved efficiently by the M-MAPEL algorithm presented in Section 3.3.

3.2 Power Control as Monotonic Optimization

Due to the non-convex nature of Problem (P3.2), it is impossible to find the optimal solution based on the theory of convex optimization [47]. Fortunately, it is recently found in operations research that monotonicity is another important property besides convexity that can be exploited to efficiently solve an optimization problem. By exploiting the hidden monotonicity of optimization problems, we can bypass the non-convexity issue and obtain the global optimal solution efficiently. In this section, we first show how Problem (P3.2) can be

transformed into a MO problem. We then discuss several key properties of the reformulation that are critical for the design of the M-MAPEL algorithm that solves the optimization problem.

Let z denote the vector $(z_{11}, \cdots, z_{1M}, \cdots, z_{L1}, \cdots, z_{LM})$. Since the function $\hat{U}_i(r_i)$ is increasing, it is easy to see that the function $\Phi(z) = \sum_{i=1}^{M} \hat{U}_i(\sum_{l=1}^{L} \log_2(z_{li}))$ is an increasing function on $\mathbb{R}_+^{M\times L}$. That is, for any two vector z_1 and z_2 such that $z_1 \succeq z_2$, we have $\Phi(z_1) \geq \Phi(z_2)$. We further note that $1 + \gamma_{l,i}(\boldsymbol{p}_l)$ for all i and l is strictly positive due to the existence of positive noise power $n_{l,i}$. Based on these two observations and the fact that $r_i = \sum_{l=1}^{L} \log_2(1 + \gamma_{l,i}(\boldsymbol{p}_l))$, Problem (P3.2) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \Phi(z) = \sum_{i=1}^{M} \hat{U}_i(\sum_{l=1}^{L} \log_2(z_{li})) \\ \text{variables} \quad & z \in \mathcal{G}, \end{aligned} \tag{P3.3}$$

where the feasible set

$$\mathcal{G} = \{z | 0 \leq z_{li} \leq 1 + \gamma_{l,i}(\boldsymbol{p}_l), \forall i \in \mathcal{M}, \forall l \in \mathcal{L}, \boldsymbol{p} \in \mathcal{P}\} \tag{3.2.1}$$

with

$$\mathcal{P} = \{\boldsymbol{p} | 0 \leq p_{l,i} \leq P_{l,i}^{\max}, \sum_{l=1}^{L} p_{l,i} \leq P_i^{\max}, \ \forall i \in \mathcal{M}, \forall l \in \mathcal{L}\}.$$

Since $\Phi(z)$ is an increasing function in z, the optimal solution to Problem (P3.3), denoted by z^*, must occur at places where $z_{li} = 1 + \gamma_{l,i}(\boldsymbol{p}_l)$ for all i and l. If we can find a power allocation $\boldsymbol{p}^*$ corresponding to the optimal solution z^* such that $z_{li}^* = 1 + \gamma_{l,i}(\boldsymbol{p}_1^*)$ for all i and l, then such $\boldsymbol{p}^*$ is clearly the optimal solution to Problem (P3.2). Finding such $\boldsymbol{p}^*$ requires solving $M \times L$ linear equations $z_{li}^*(\sum_{j\neq i} G_{l,ji} p_{l,j}^* + n_{l,i}) - (\sum_{j=1}^{M} G_{l,ji} p_{l,j}^* + n_{l,i}) = 0$ with $M \times L$ variables $p_{1,1}^*, \cdots, p_{L,1}^*, \cdots, p_{1,M}^*, \cdots, p_{L,M}^*$. As the channel gains $G_{l,ij}$'s are generally random, we can show with probability 1 that the $M \times L$ equations are linearly independent, implying there is a unique solution $\boldsymbol{p}^*$. Hence Problems (P3.1),

(P3.2) and (P3.3) are all equivalent with each other. *We will focus on how to solve Problem (P3.3) efficiently in the rest of the chapter.*

Before attempting to solve Problem (P3.3), it is critical to understand several important properties of the feasible set $\mathcal{G}$ in (3.2.1).

According to the definition of box, the feasible set $\mathcal{G}$ can be characterized as a union of infinite number of boxes with vertices of all boxes belonging to the set $\{\boldsymbol{c}|c_{li} = 1 + \gamma_{l,i}(\boldsymbol{p}_l), \forall i \in \mathcal{M}, \forall l \in \mathcal{L}, \boldsymbol{p} \in \mathcal{P}\}$, where $\boldsymbol{c}$ is denoted by the vector $(c_{11}, \cdots, c_{1M}, \cdots, c_{L1}, \cdots, c_{LM})$. Thus, by Proposition 2.1, the feasible set $\mathcal{G}$ is a normal set. This, together with $\Phi(\boldsymbol{z})$ being an increasing function in $\boldsymbol{z}$, implies that Problem (P3.3) is a MO problem. Therefore, by Proposition 2.2, the optimal solution to Problem (P3.3), must occur at the upper boundary of set $\mathcal{G}$.

Note that set $\mathcal{G}$ is a non-convex set. However, convexity is not important in obtaining the global optimal solution. Since Problem (P3.3) is a MO problem, it is the monotonicity of the objective function and the normality of the feasible set in the reformulated problem (P3.3) that facilitates efficient calculation of the global optimal solution.

Before leaving this section, notice that $1 + \gamma_{l,i}(\boldsymbol{p}_l)$ is lower bounded by 1 for any feasible $\boldsymbol{p}_l$. Consequently, the optimal solution $\boldsymbol{z}^*$ to Problem (P3.3), which occurs only at places (i.e., the upper boundary of set $\mathcal{G}$) where $z_{li} = 1 + \gamma_{l,i}(\boldsymbol{p}_l)$ for all i and l, is also lower bounded by $\mathbf{1}$. That is, the optimal solution $\boldsymbol{z}^*$ must reside in the set $\mathcal{G} \cap \Theta$, where $\Theta = \{\boldsymbol{z}|z_{li} \geq 1, \forall i \in \mathcal{M}, \forall l \in \mathcal{L}\}$.

3.3 The M-MAPEL Algorithm

In this section, we propose a novel algorithm, M-MAPEL, to solve Problem (P3.3) based on the special characteristics of such problem.

3.3.1 The M-MAPEL Algorithm

Based on some mathematical preliminaries introduced in Section 2.3.1 of Chapter 2, the M-MAPEL algorithm works as follows. We first construct a polyblock $\mathcal{S}_1$ that contains the feasible set of Problem (P3.3), $\mathcal{G}$. Let $\mathcal{T}_1$ denote the proper vertex set of $\mathcal{S}_1$. By Proposition 2.3, the maximum of the objective function of Problem (P3.3) (i.e., $\Phi(z) = \sum_{i=1}^{M} \hat{U}_i(\sum_{l=1}^{L} \log_2(z_{li})))$ over set $\mathcal{S}_1$ occurs at some proper vertex z_1 of $\mathcal{S}_1$, i.e., $z_1 \in \mathcal{T}_1$. If z_1 happens to reside in $\mathcal{G}$ as well, then it solves Problem (P3.3) and $z^* = z_1$. Otherwise, based on Proposition 2.4, we can construct a smaller polyblock $\mathcal{S}_2 \subset \mathcal{S}_1$ that still contains $\mathcal{G}$ but excludes z_1. This is achieved by constructing the vertex set $\mathcal{T}_2$ by replacing z_1 in $\mathcal{T}_1$ with $M \times L$ new vertices $\{z_{11}, \cdots, z_{1(L\times M)}\}$ and then removing improper vertices, where $z_{1j} = z_1 - (z_{1,j} - \pi_j^{\mathcal{G}}(z_1))e_j$. We can repeat this procedure until an optimal solution is found. This leads to a sequence of polyblocks containing $\mathcal{G}$: $\mathcal{S}_1 \supset \mathcal{S}_2 \supset \cdots \supset \mathcal{G}$. Obviously, $\Phi(z_1) \geq \Phi(z_2) \geq \cdots \Phi(z^*)$, where z_n is the optimal vertex that maximizes $\Phi(z)$ over set S_n. In addition to obtaining the optimal vertex z_n, we set the current best solution (CBS) $z'_n = \text{argmax}\{\Phi(y)|y \in \{\pi^{\mathcal{G}}(z_n), z'_{n-1}\}\}$ and the current best value (CBV) $\Phi(z'_n)$ at each iteration. Specially, $z'_1 = \pi^{\mathcal{G}}(z_1)$. Then, we have $\Phi(z'_1) \leq \Phi(z'_2) \leq \cdots \Phi(z^*)$. The algorithm terminates at the nth iteration if $z_n \in \mathcal{G}$. For practical implementation, we say $z_n \in \mathcal{G}$ when $(1+\varepsilon)\Phi(z'_n) \geq \Phi(z_n)$, where $\varepsilon > 0$ is a small positive number representing the error tolerance level.

We can further expedite the above process by selecting z_n from a smaller set $\mathcal{T}_n \cap \Theta$, where $\Theta = \{z|z_{li} \geq 1, \forall i \in \mathcal{M}, \forall l \in \mathcal{L}\}$. This will not affect the convergence or optimality of the method since the optimal solution z^* is lower bounded **1**.

A critical step in constructing new polyblock and checking the termination

criterion is the calculation of the projection $\pi^{\mathcal{G}}(z_n)$. This is, however, by no means trivial, since the upper boundary of $\mathcal{G}$ is not explicitly known. In particular, $\pi^{\mathcal{G}}(z_n) = \lambda_n z_n$ is obtained by solving the following max-min problem for λ_n:

$$\begin{aligned}
\lambda_n &= \max\{\lambda | \lambda z_n \in \mathcal{G}\} \\
&= \max\{\lambda | \lambda \leq \min_{i\in\mathcal{M}, l\in\mathcal{L}} \frac{1+\gamma_{l,i}(\boldsymbol{p}_l)}{z_{n,li}}, \boldsymbol{p} \in \mathcal{P}\} \\
&= \max_{\boldsymbol{p}\in\mathcal{P}} \min_{i\in\mathcal{M}, l\in\mathcal{L}} \frac{\sum_{j=1}^{M} G_{l,ji}p_{l,j} + n_{l,i}}{z_{n,li}(\sum_{j\neq i} G_{l,ji}p_{l,j} + n_{l,i})} \\
&= \max_{\boldsymbol{p}\in\mathcal{P}} \min_{i\in\mathcal{M}, l\in\mathcal{L}} \frac{f_{li}(\boldsymbol{p}_l)}{z_{n,li}g_{li}(\boldsymbol{p}_l)}
\end{aligned} \tag{3.3.1}$$

This is a fractional programming problem by the definition in [46]. Hence, we can solve this problem using the Dinkelbach algorithm in [46] with slight modifications. The details are shown in Algorithm 3.1:

Algorithm 3.1 Max-Min Projection Algorithm (for finding $\pi^{\mathcal{G}}(\{z_n\})$)

1: **Initialization**: Choose $\boldsymbol{p}^{(0)} \in \mathcal{P}$ and let $m = 0$.
2: **repeat**
3: Given $\boldsymbol{p}^{(m)}$, solve $\lambda_n^{(m)} = \min_{i,l} \frac{f_{li}(\boldsymbol{p}_l^{(m)})}{z_{n,li}g_{li}(\boldsymbol{p}_l^{(m)})}$.
4: Given $\lambda_n^{(m)}$, solve $\boldsymbol{p}^{(m+1)} = \operatorname{argmax}_{\boldsymbol{p}\in\mathcal{P}} \min_{i,l}(f_{li}(\boldsymbol{p}_l) - \lambda_n^{(m)} z_{n,li}g_{li}(\boldsymbol{p}_l))$.
5: $m = m + 1$.
6: **until** $\max_{\boldsymbol{p}\in\mathcal{P}} \min_{i,l}(f_{li}(\boldsymbol{p}_l) - \lambda_n^{(m-1)} z_{n,li}g_{li}(\boldsymbol{p}_l)) \leq 0$.
7: The projection is $\pi^{\mathcal{G}}(z_n) = \lambda_n^{(m-1)} z_n$.

Theorem 3.1. *The sequence $\{\lambda_n^{(m)}, m = 1, 2, \cdots\}$ converges Q-super linearly to the optimal solution.*

Proof: Since $f_{li}(\boldsymbol{p}_l)$ and $z_{n,li}g_{li}(\boldsymbol{p}_l)$ are linear affine functions on $\boldsymbol{p}_l$ for all i and l and there is a unique optimal solution to (3.3.1), Theorem 3.1 is immediate from Theorem 8.7 in [46].

Having introduced the basic operations, we now formally present the M-MAPEL algorithm in Algorithm 3.2.

3.3.2 Global Convergence

The following theorem is proved in Section 3.6.1 and it shows that M-MAPEL is a global optimal algorithm of solving the system utility maximization problem.

Theorem 3.2. *The M-MAPEL algorithm globally converges to a global optimal solution of Problem (P3.3).*

Before leaving this subsection, note that although M-MAPEL is proved to converge to the global optimal solution, the convergence speed is still an open problem.

3.3.3 Trade-off between Performance and Convergence Time

The convergence time M-MAPEL is infinite if the error tolerance $\varepsilon = 0$. However, it can be easily shown that M-MAPEL always terminates with finite steps as long as $\varepsilon > 0$ [18]. Next, we analyze the influence of error tolerance ε on performance.

Theorem 3.3. *The solution obtained by M-MAPEL is an ε-optimal solution.*

Proof: M-MAPEL terminates when $(1+\varepsilon)\Phi(z_n') \geq \Phi(z_n)$. Consequently,

$$\Phi(z^*) \leq \Phi(z_n) \leq (1+\varepsilon)\Phi(z_n').$$

According to Definition 2.9, this leads to Theorem 3.3 and z_n' is an ε-optimal solution. ■

Remark 3.1. *ε is generally a conservative estimate. In practice, we often yield an error that is smaller than ε since $\Phi(z^*) \leq \Phi(z_n)$.*

Algorithm 3.2 The M-MAPEL Algorithm

1: **Initialization**: Choose the error tolerance $\varepsilon > 0$, and let $n = 1$.

2: **repeat**

3: If $n = 1$, construct the initial polyblock $\mathcal{S}_1$ with vertex set $\mathcal{T}_1 = \{\boldsymbol{b}\}$, where the $(M \times (l-1) + i)$th element of vector $\boldsymbol{b}$ is

$$b_{li} = \max_{\boldsymbol{p} \preceq \mathcal{P}} \frac{f_{li}(\boldsymbol{p}_l)}{g_{li}(\boldsymbol{p}_l)} = 1 + \frac{G_{l,ii} P_{l,i}^{\max}}{n_{l,i}}, \forall i \in \mathcal{M}, \forall l \in \mathcal{L}.$$

It is clear that polyblock $\mathcal{S}_1$ is a box $[\boldsymbol{0}, \boldsymbol{b}]$ containing $\mathcal{G}$. If $n > 1$, construct a smaller polyblock $\mathcal{S}_n$ with vertex set $\mathcal{T}_n$ by replacing z_{n-1} in $\mathcal{T}_{n-1}$ with $M \times L$ new vertices $\{z_{n-11}, \cdots, z_{n-1(L\times M)}\}$, where $z_{n-1j} = z_{n-1} - (z_{n-1,j} - \pi_j^{\mathcal{G}}(z_{n-1}))\boldsymbol{e}_j$, and removing improper vertices.

4: Find z_n that maximizes the objective function of Problem (P3.3) over set $\mathcal{T}_n \cap \Theta$, i.e.,

$$z_n = \text{argmax}\{\Phi(z) | z \in \mathcal{T}_n \cap \Theta\}. \tag{3.3.2}$$

5: Find $\pi^{\mathcal{G}}(z_n)$ based on Algorithm 3.1.

6: Determine CBS z'_n and CBV $\Phi(z'_n)$, where $z'_n = \text{argmax}\{\Phi(y) | y \in \{\pi^{\mathcal{G}}(z_n), z'_{n-1}\}\}$. Specially, $z'_1 = \pi^{\mathcal{G}}(z_1)$.

7: $n = n + 1$.

8: **until** $(1+\varepsilon)\Phi(z'_{n-1}) \geq \Phi(z_{n-1})$.

9: If $\Phi(z'_{n-1}) > -\infty$, the compute the optimal power allocation $\boldsymbol{p}^*$ (i.e., optimal solution to Problem (P3.1) by solving $z'_{n-1,li} = \frac{f_{li}(\boldsymbol{p}_l^*)}{g_{li}(\boldsymbol{p}_l^*)}$ for all i and l. Otherwise, there is no feasible solution of Problem (P3.1).

3.4 Performance Evaluation

We illustrate the effectiveness of the M-MAPEL algorithm through several examples.

3.4.1 Performance Evaluation of M-MAPEL

Example 1 (Performance and convergence time tradeoff through the error tolerance ε): We first consider a network of four subcarriers with two links (i.e., $M = 2$ and $L = 4$). We assume that the channel gains are i.i.d. (independent and identically distributed) with exponential distribution with mean d_{ji}^{-4}, where d_{ji} denotes the distance between the transmitter node T_j and the receiver node R_i. We consider a realization of the channel gains, i.e., the resultant channel gain matrix is

$$G_A = \begin{bmatrix} 0.0154 & 0.0017 & 0.0633 & 0.0022 & 0.0379 & 0.0227 & 0.0182 & 0.0029 \\ 0.0015 & 0.2955 & 0.0055 & 0.1080 & 0.0314 & 0.0215 & 0.0163 & 0.4791 \end{bmatrix}. \tag{3.4.1}$$

For this realization, we run the M-MAPEL algorithm with different error tolerance ε, and plot the attained system utility as well as the number of iterations needed for convergence in Figs. 3 and 4. In particular, $U_i(r_i) = r_i$ in Fig. 3.1 and $U_i(r_i) = \frac{1}{1+e^{-a_i(r_i-b_i)}}$ with $\{a_i, b_i\} = \{1, 10\}$ in Fig. 3.2. Assume that $P_i^{\max,l} = 0.5m\text{W}$ and $n_i^l = 0.1\mu\text{W}$ for all link i and subcarrier l, and $P_i^{\max} = 1.6m\text{W}$ for all link i. The minimum data rate constraints are set to be zero in this example.

From these two figures, we see that the convergence time of M-MAPEL can be quite different for different objective functions. Regardless, the total number of iterations increases when ε decreases in both figures, and the change is drastic when ε is close to 0. Moreover, the algorithm performance is not sensitive to the

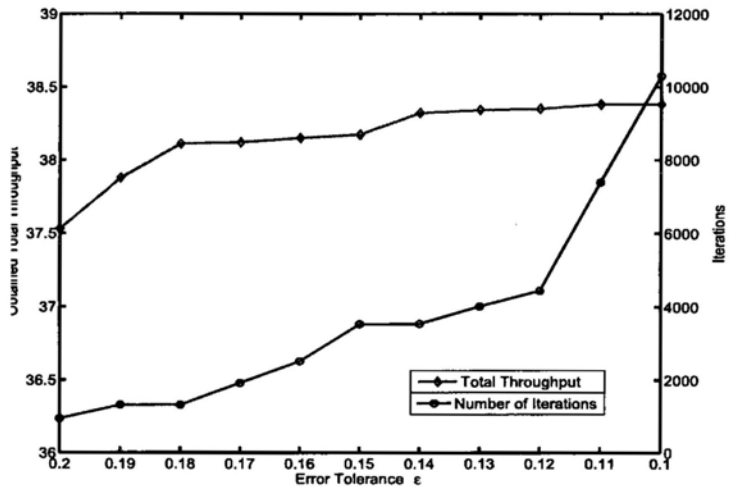


Figure 3.1: Obtained total throughput and number of iterations for different error tolerance ε

value of ε. As shown in Fig. 3.1, when $\varepsilon = 0.2$, we achieve a total throughput of 37.54bps/Hz that is only 2.2% away from the best value obtained when $\varepsilon = 0.10$. Additionally, Fig. 3.2 also shows that when $\varepsilon = 0.01$, the summation loss is only 0.35%, compared with the exact optimum. This observation illustrates that the performance bound obtained in Theorem 3.3 is quite loose, and the actual performance could be much better than the bound. It is also clear that parameter ε provides a tuning knob for achieving various trade-off between algorithm performance and convergence time.

Example 2 (Global optimal power allocation): M-MAPEL attains the global optimal solution[2] of the objective function as given in (P3.1) for an arbitrary wireless network. This is not possible before without exhaustive search. The effectiveness of M-MAPEL provides us with a convenient tool to observe the characteristics of optimal power control solution, which may serve as a guideline for the design of heuristic algorithms. In this example, we use a simple

[2]M-MAPEL will only find one of the possible many global optimal solutions, depending on the choice of initial conditions.

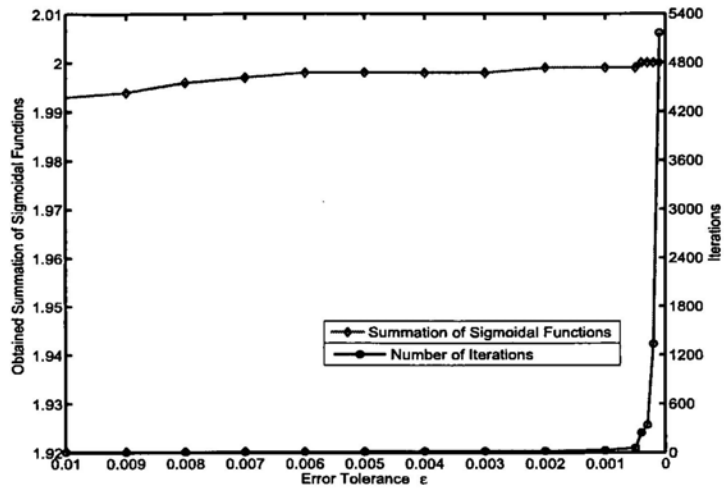


Figure 3.2: Obtained summation of sigmoidal functions and number of iterations for different error tolerance ε

4-subcarrier 4-link network in Fig. 5 to illustrate how M-MAPEL can be used to study the optimal power control strategy. Assume that the channel gains are i.i.d. with exponential distribution with mean d_{ji}^{-4}. Herein, we take one realization of the channel gains as a simple illustrating example. Note that link 4 has the best average channel quality, whereas link 1 has the worst average channel quality. Consider four different objective functions, including the total throughput function (TTF), the proportional fairness function (PFF), the max-min fairness function (MFF) and the summation of sigmoidal functions (SSF). Assume that $P_i^{\max,l} = 0.5m\text{W}$ and $n_i^l = 0.1\mu\text{W}$ for all link i and subcarrier l, and $P_i^{\max} = 1.6m\text{W}$ for all link i. The minimum data rate constraints in this example are set to 1.0bps/Hz for all links.

Table 3.1 shows the optimal power allocation and the corresponding data rates obtained by M-MAPEL for the different objective functions. As shown in Table 3.1, maximizing TTF leads to a highly uneven resource distribution: the good links (e.g., link 4) are allocated more transmission power. As a result, the best link (link 4) has the highest data rate whereas the worst link (link 1) has

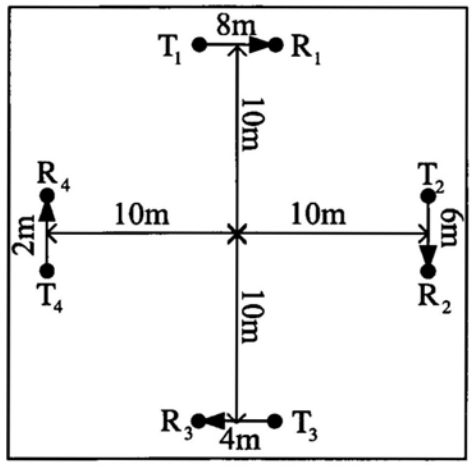


Figure 3.3: A network of four subcarriers with four links

the lowest data rate. Meanwhile, to achieve max-min fairness (i.e., MFF in Table I), more power is allocated to links with small channel gains, so that the data rates of different links are equalized. Proportional fair allocation (i.e., PFF), on the other hand, strikes a balance between total throughput and fairness. Furthermore, it is interesting to note that SSF endeavors to keep the data rate of each link beyond the threshold b_i whenever possible, as shown in the case when $b_i = 10$. When b_i is too high to be exceeded by all links (e.g., the case with $b_i = 20$), the worst link (link 1) is sacrificed first. As shown in the table, link 1 only gets the minimum required data rate 1.0bps/Hz.

3.4.2 Providing Benchmark for Existing Algorithms

A key application of M-MAPEL is to provide performance benchmark for other algorithms that have been (or to be) proposed to maximize the objective functions as given in (P3.1). With M-MAPEL, we are able to give quantitative measurements of these algorithms' performance (e.g., the probability of achieving global optimal solution) under a wide range of network scenarios (e.g. different

network densities and topologies). Notice that the focus here is to show how M-MAPEL can be used to provide effective benchmark for the algorithms that tackle the same problem (i.e., Problem (P3.1)). Readers can choose your favorite algorithm to conduct the study. Our choice in this subsection may be biased, and the selected algorithm may not be "the best".

SCALE algorithm is considered to be one of the best existing algorithm for maximizing the total throughput in the multi-carrier networks. The key idea of SCALE algorithm is to improve the solution of throughput maximization problem through successive convex approximation until a KKT point is reached.

Example 3 (Benchmark provided by M-MAPEL): M-MAPEL always guarantees global optimality, while the SCALE algorithm fails to do so. We consider two networks of four subcarriers with two links (i.e., $M = 2$ and $L = 4$). The $[d_{ji}^{-4}]$ matrices of two networks are as follows:

$$\boldsymbol{D}_1 = \begin{bmatrix} 0.1295 & 0.0036 \\ 0.0018 & 0.2277 \end{bmatrix} \quad \boldsymbol{D}_2 = \begin{bmatrix} 0.7508 & 0.2963 \\ 0.2589 & 0.4933 \end{bmatrix}, \tag{3.4.2}$$

respectively. We assume that the channel gains are i.i.d. with exponential distribution of mean d_{ji}^{-4}. Other system parameters are the same in Example 1 except that there is no constraints on $P_i^{\max,l}$. Herein, the $P_i^{\max,l}$ constraints are not considered. For each network, we simulate both algorithms with 100 random realizations of the channel gains and show the results in Fig. 3.4 and Fig. 3.5, respectively. The figures show that M-MAPEL always converges to the global optimality, regardless of the choice of channel gains. On the other hand, the SCALE is trapped in local optimality from time to time. For example, Fig. 3.4 shows that SCALE algorithm achieves the global optimality 94% of the time. However, Fig. 3.5 shows that in a different topology SCALE algorithms achieve the global optimal solution 62% of the time. In these two figures, we can find that the probability of achieving global optimal solution for SCALE is sensitive

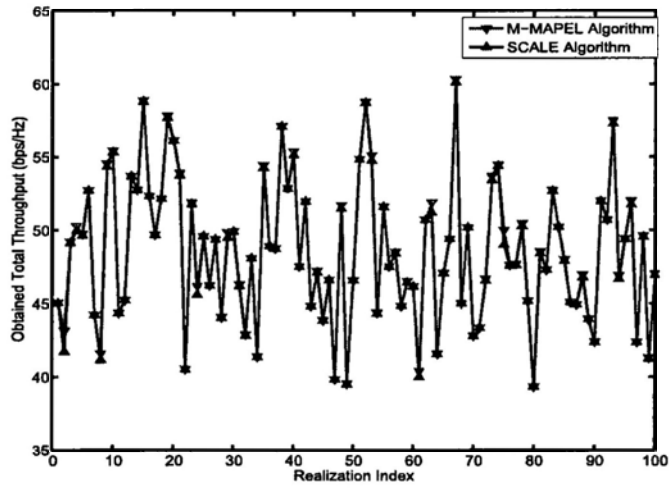


Figure 3.4: Maximum total throughput achieved by M-MAPEL as well as SCALE for 100 channel gain realizations in D_1 network

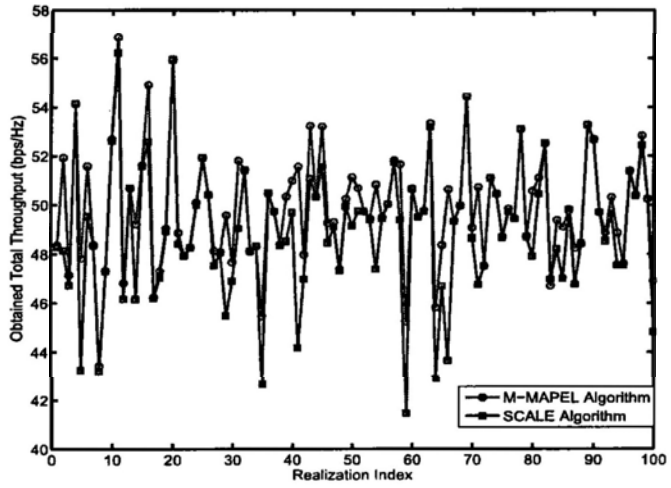


Figure 3.5: Maximum total throughput achieved by M-MAPEL as well as SCALE for 100 channel gain realizations in D_2 network

to the network topology.

3.5 Summary

In this chapter, we proposed the M-MAPEL algorithm that solves the power control problem in multi-carrier wireless networks. The M-MAPEL algorithm is guaranteed to converge to an global optimal solution despite the nonconvexity of the problem. The key idea behind the algorithm is to reformulate an power control problem into a MO problem, and then construct a sequence of shrinking polyblocks that eventually closely approximate the upper boundary of the feasible region around the global optimum. We have also established the tradeoff relationship between performance and convergence time of the M-MAPEL algorithm.

Guaranteed to converge to the global optimal solution, M-MAPEL provides an important benchmark for performance evaluation of other power control heuristics in this area. For example, by comparing with M-MAPEL through extensive simulations, we have gained deeper understanding of the state-of-the-art power control algorithm: SCALE algorithm. Simulations show that the SCALE algorithm achieves the close-to-optimal power allocation.

3.6 Appendix

3.6.1 Proof of Theorem 3.2

The M-MAPEL algorithm generates a sequence $\{z_n\}$ for $n = 1, 2, \cdots$. Each component is calculated as (3.3.2) for each newly constructed polyblock. We

can find a subsequence $\{z_{n_j}\}$ within the sequence $\{z_n\}$ such that

$$\begin{aligned} z_{n_1} &= z_1 - (z_{1,i_0} - \pi^{\mathcal{G}}_{i_0}(z_{n_1}))e_{i_0} \\ &\vdots \\ z_{n_{j+1}} &= z_{n_j} - (z_{n_j,i_j} - \pi^{\mathcal{G}}_{i_j}(z_{n_j}))e_{i_j}, \end{aligned} \tag{3.6.1}$$

where $1 < n_1 < n_2 < \cdots < n_j < \cdots$. z_{n_j,i_j} denotes the i_jth element of vector z_{n_j}, where i_j is the only position in which $z_{n_{j+1}}$ differs from z_{n_j}. This subsequence can be thought of as the "off-springs" of vertex z_1 through a series of projections, and they are not necessarily adjacent since there might be projections of other vertices happening in between. It can be shown that there is at least one such subsequence that has infinite length. With a slight abuse of notation, let $\{z_{n_j}, \forall j \geq 1\}$ denote such one subsequence. Since $z_{n_1} \succeq \pi^{\mathcal{G}}(z_{n_j})$, (3.6.1) implies that $z_1 \succeq z_{n_1} \succeq \cdots \succeq z_{n_j} \succeq \cdots \succeq \mathbf{1}$. Hence, $\lim_{j\to\infty} \|z_{n_j} - z_{n_{j+1}}\| \to 0$. From (3.6.1) we know that z_{n_j} and $z_{n_{j+1}}$ only differ in the i_j's position, thus

$$\|z_{n_j} - z_{n_{j+1}}\| = z_{n_j,i_j} - z_{n_{j+1},i_j} = z_{n_j,i_j} - \pi^{\mathcal{G}}_{i_j}(z_{n_j}) \to 0 \text{ when } j \to \infty. \tag{3.6.2}$$

Since $\pi^{\mathcal{G}}(z_{n_j}) = \lambda_{n_j} z_{n_j}$ and $z_{n_j} \succeq \mathbf{1}$, (3.6.2) implies that $\lim_{j\to\infty} \lambda_{n_j} = 1$. That is,

$$\lim_{j\to\infty} z_{n_j} \to \pi^{\mathcal{G}}(z_{n_j}) \tag{3.6.3}$$

and

$$\lim_{j\to\infty} \Phi(z_{n_j}) \to \Phi(\pi^{\mathcal{G}}(z_{n_j})). \tag{3.6.4}$$

Eqn. (3.6.3) implies that the subsequence $\{z_{n_j}\}$ converges to the boundary of the feasible region $\mathcal{G}$. Since z_{n_j} is a maximizer over the set $\mathcal{S}_{n_j}$, eqn. (3.6.3) also implies that z_{n_j} is the global optimum of Problem (P3.3) when $j \to \infty$. Since the CBS subsequence $\{z'_{n_j}\}$ satisfies $\Phi(z_{n_j}) \geq \Phi(z'_{n_j}) \geq \Phi(\pi^{\mathcal{G}}(z_{n_j}))$ for any n_j, it follows from (3.6.4) that

$$\lim_{j\to\infty} \Phi(z_{n_j}) \to \lim_{j\to\infty} \Phi(z'_{n_j}) \to \Phi(\pi^{\mathcal{G}}(z_{n_j})). \tag{3.6.5}$$

Together with z_{n_j} being the global optimal of Problem (P3.3) when $j \to \infty$, eqn. (3.6.4) implies z'_{n_j} is also the global optimum of Problem (P3.3) when $j \to \infty$. Note that the M-MAPEL algorithm terminates once the optimal solution to Problem (P3.3) is found. Therefore, the convergence of the subsequence $\{z_{n_j}\}$ guarantees the convergence of the algorithm to the global optimal solution. ■

Table 3.1: The optimal data rates and power allocations obtained by M-MAPEL for different objective functions

Objective Function	Optimal Data Rates $\boldsymbol{R} = (R_1, R_2, R_3, R_4)$bps/Hz	Optimal Power Allocations $\boldsymbol{p} = (p_{1,1}, \cdots, p_{1,4}, \cdots, p_{4,1}, \cdots, p_{4.4})mW$
TTF	(5.1433 28.8597 4.9351 49.4050)	(0 0.1440 0.0144 0.2122 0 0.5 0.0115 0.2845 0.3298 0.0914 0.0312 0.1411 0.0010 0.5 0.0001 0.0990)
MFF	(15.5743 17.2929 15.6316 15.8840)	(0.3070 0.1418 0.0405 0.0098 0 0.4720 0.0018 0.0003 0.5 0 0.0063 0.0009 0.0882 0 0.1443 0.0037)
PFF	(14.1539 13.7719 16.8192 31.8451)	(0.0014 0.5 0.0334 0.0433 0.0735 0.2894 0.3682 0.1159 0.5 0.0016 0.0458 0.0353 0.5 0.0009 0.0146 0.0172)
SSF $\{a_i, b_i\} = \{1, 10\}, \forall i$	(15.1081 16.0493 15.3895 20.2449)	(0.2638 0.1974 0.1581 0.3321 0 0.5 0.0065 0.0010 0.3598 0.0062 0.0005 0 0.2774 0.0045 0.1753 0.0202)
SSF $\{a_i, b_i\} = \{1, 20\}, \forall i$	(1.0008 21.8800 22.9110 30.8631)	(0 0.1819 0.2228 0.0216 0 0.3596 0.5 0.0023 0 0.5 0 0.0158 0.23 0.270 0.1488 0.2644)

Chapter 4

S-MAPEL: Monotonic Optimization for Non-convex Joint Power Control and Scheduling Problems

In this chapter, we start with a interference-limited wireless network where simultaneous transmissions on nearby links heavily interfere with each other. The motivating example in Chapter 1 reveals that power control alone is not sufficient to eliminate strong levels of interference between close-by links in this context. Correspondingly, scheduling, which allows close-by links to take turns to be active, plays a crucial rule for achieving high system performance. Joint power control and scheduling that maximizes the system utility has long been a challenging problem. The complicated coupling between the signal-to-interference ratio of concurrently active links as well as the flexibility to vary power allocation over time gives rise to a series of non-convex optimization problems, for which the global optimal solution is hard to obtain. Our work is a first attempt to solve the non-convex joint power control and scheduling problems efficiently in a global optimal manner. In particular, it is the mono-

tonicity rather than the convexity of the problem that we exploit to devise an efficient algorithm, referred to as S-MAPEL, to obtain the global optimal solution. To further reduce the complexity, we propose an accelerated algorithm, referred to as A-S-MAPEL, based on the inherent symmetry of the optimal solution. The optimal joint-power-control-and-scheduling solution obtained by the proposed algorithms serves as a useful benchmark for evaluating other existing schemes. With the help of this benchmark, we find that on-off scheduling is of much practical value in terms of system utility maximization if "off-the-shelf" wireless devices are to be used.

The rest of this chapter is organized as follows. Section 4.1 introduces the system model and the problem formulation. In Section 4.2, we reformulate the joint power control and scheduling problem into a MO problem with finite size. Some properties of the feasible region of MO problem are also discussed. The S-MAPEL algorithm is proposed and analyzed in Section 4.3. Section 4.4 presents an accelerated algorithm, referred to as A-S-MAPEL, to reduce the computational complexity for the joint power control and scheduling problem. In Section 4.5, we evaluate the performance of the proposed algorithms through several simulations. We show that the primal/dual decomposition fails to achieve the global optimal solution of joint power control and scheduling in Section 4.6. The conclusions are drawn in Section 4.7. The global convergence of S-MAPEL is proved in Section 4.8.

4.1 System Model and Problem Formulation

We consider a single-hop wireless ad hoc network with a set $\mathcal{M} = \{1, \cdots, M\}$ of *distinct* links. Each link consists of a transmitter node T_i and a receiver node R_i. The channel gain between node T_i and node R_j is denoted by G_{ij}, which is

determined by various factors such as path loss, shadowing and fading effects. To condense the notation, we write the channel gains into a channel matrix form $\boldsymbol{G} = [G_{ij}]$. Assume that the channel gains are constant during the time interval T under consideration. For such a network, there are typically two strategies in the current literature, to maximize the overall system utility, namely *pure power control* and *joint power control and scheduling*. We have already shown how the pure power control strategy can be formulated into monotonic optimization problems in Chapter 3. In what follows, we further show how the joint power control and scheduling strategy can be formulated into monotonic optimization problems. Notably, the joint power control and scheduling problem is much more complicated due to the additional freedom in the time domain, although pure power control is already an NP-hard problem by itself.

Let $p_i(t)$ denote the transmission power of link i (i.e., from node T_i) at time instant t, with $P_i^{\max}$ being its maximum allowable value. For notational convenience, we write $\boldsymbol{p}(t) = (p_i(t), \forall i \in \mathcal{M})$ and $\boldsymbol{P}^{\max} = (P_i^{\max}, \forall i \in \mathcal{M})$ as the transmission power vector at time instant t and the maximum transmission power vector, respectively. Likewise, let the received noise on link i be n_i. Thus, the received SINR of link i at time instant t is

$$\gamma_i(\boldsymbol{p}(t)) = \frac{G_{ii}p_i(t)}{\sum\limits_{j \neq i} G_{ji}p_j(t) + n_i}, \tag{4.1.1}$$

and the corresponding data rate $r_i(t)$ calculated based on the Shannon capacity formula is $\log_2(1 + \Gamma\gamma_i(\boldsymbol{p}(t)))$, where $\Gamma \in [0, 1]$ is the SINR gap that reflects a particular modulation and coding scheme. Without loss of generality, we assume $\Gamma = 1$ hereafter.

As the motivating example shows in Chapter 1, power control alone is not sufficient to eliminate strong levels of interference between close-by links when links are close to each other. In dense networks, joint power control and schedul-

ing leads to much higher system utility and throughput than pure power control schemes. We aim to find the optimal joint power control and scheduling that maximizes the overall system utility subject to individual user's QoS requirements. To satisfy the QoS requirement, each link must maintain a minimum data rate $r_i^{\min}$. Mathematically, the joint power control and scheduling problem can be formulated as follows

$$\begin{aligned} &\underset{\{\boldsymbol{p}(t)\}}{\text{maximize}} \ U(\boldsymbol{r}) \\ &\text{subject to } r_i = \frac{1}{T}\int_0^T \log_2(1+\gamma_i(\boldsymbol{p}(t)))dt \geq r_i^{\min}, \forall i \in \mathcal{M} \\ &\qquad \mathbf{0} \preceq \boldsymbol{p}(t) \preceq \boldsymbol{P}^{\max}, \forall t \in [0,T], \end{aligned} \tag{P4.1}$$

where $U(\boldsymbol{r})$ represents the system-wide utility, with $\boldsymbol{r}$ being the vector of r_i. It is assumed to be additive across links, i.e., $U(\boldsymbol{r}) = \sum_{i=1}^{M} U_i(r_i)$, with $U_i(r_i)$ being the utility of link i. Typically, $U_i(r_i)$ is either a concave non-decreasing function for elastic traffic or a non-concave non-decreasing function for delay-sensitive traffic. In our work, we do not impose the concavity assumption on $U_i(r_i)$. The only assumption is $U_i(r_i)$ being increasing, which is a valid assumption for most applications. By appropriately choosing the utility function $U_i(r_i)$, we can strike different balances between spectrum efficiency and fairness. Some possible utility functions have been introduced previously, and interested readers are referred to Chapter 2. Note that scheduling has been integrated into the time-varying power allocation $\boldsymbol{p}(t)$, as choosing a subset of links to transmit during a certain time interval is equivalent to allocating zero transmit power to the unscheduled links and positive power to the scheduled ones.

Joint power control and scheduling in (P4.1) is much more challenging than pure power control in (P3.1) because of the following two reasons. First, the problem size of (P4.1) is infinite as t is continuous during $[0,T]$. Second, the shape of the feasible set is further complicated by the coupling between $\boldsymbol{p}(t)$ at

different t. Fortunately, a close look at (P4.1) reveals that it possesses similar hidden monotonicity as the power control problem (P3.1), due to the fact that the problem maximizes an objective function monotonically increasing with SINR. This opens the possibility to solve the problem through MO methods.

4.2 Joint Power Control and Scheduling as Monotonic Optimization

Before we can formulate Problem (P4.1) into a MO problem that is amenable to efficient solutions, we must address the two challenges mentioned in the last sections, namely, infinite problem size and complicated shape of the feasible region. In the next subsection, we show how Problem (P4.1) can be reformulated to get rid of the hurdles.

4.2.1 Discretization

- **The issue of infinite problem size**

The issue of infinite problem size can be addressed by the following Lemma and Theorem. Before presenting the Lemma and Theorem, we first define the achievable instantaneous data rate set $\mathcal{R}(t)$ and the achievable average data rate set $\bar{\mathcal{R}}$:

$$\mathcal{R}(t) = \left\{ \boldsymbol{r}(t) | r_i(t) = \log_2(1 + \gamma_i(\boldsymbol{p}(t))) \text{ and } \mathbf{0} \preceq \boldsymbol{p}(t) \preceq \boldsymbol{P}^{\max}, \forall i \in \mathcal{M} \right\}, \tag{4.2.1}$$

and

$$\begin{aligned} \bar{\mathcal{R}} = \Big\{ \boldsymbol{r} | r_i = \frac{1}{T} \int_0^T \log_2(1 + \gamma_i(\boldsymbol{p}(t)))dt \\ \text{and } \mathbf{0} \preceq \boldsymbol{p}(t) \preceq \boldsymbol{P}^{\max}, \forall i \in \mathcal{M}, \forall t \in [0, T] \Big\}. \end{aligned} \tag{4.2.2}$$

The instantaneous data rate set $\mathcal{R}(t)$ is the set of all data rates achievable by power allocation at time instant t. On the other hand, the average data rate set $\bar{\mathcal{R}}$ is the set of all achievable *average* data rates during a scheduling period T through time-varying power allocation.

Remark 4.1. *Notably, $\mathcal{R}(t)$ is the same for all $t \in [0, T]$, since the channel is constant during the period of T.*

By the standard convexity argument, Remark 4.1 leads to Lemma 4.1 [47].

Lemma 4.1. *The achievable average data rate set $\bar{\mathcal{R}}$ is the convex hull of the instantaneous data rate set $\mathcal{R}(t)$, i.e, $\bar{\mathcal{R}} = \text{Convex Hull}\{\mathcal{R}(t)\}$.*

Theorem 4.1. *By Caratheodory theorem [47] and Lemma 4.1, the number of elements in $\mathcal{R}(t)$ that is needed to construct an arbitrary average data rate vector $\boldsymbol{r} = (r_i, \forall i \in \mathcal{M})$ in $\bar{\mathcal{R}}$ is no more than $M + 1$.*

Theorem 4.1 implies that an arbitrary average data rate vector $\boldsymbol{r}$ can be achieved by dividing $[0, T]$ into $M + 1$ intervals with lengths $\beta_1, \cdots, \beta_{M+1}$ and assigning power vectors $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{M+1}$ to these intervals. In particular, if less than $M + 1$ intervals are needed for the optimal solution, then some β_k's and $\boldsymbol{p}_k$'s are equal to zero. Therefore, the constraints of Problem (P4.1) can be replaced by

$$\begin{aligned} r_i &= \sum_{k=1}^{M+1} \beta_k \log_2(1 + \gamma_i(\boldsymbol{p}_k)) \geq r_i^{\min}, \forall i \in \mathcal{M} \\ &\sum_{k=1}^{M+1} \beta_k = 1, \boldsymbol{\beta} = (\beta_1, \cdots, \beta_{M+1}) \succeq \mathbf{0} \\ &\mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}, \forall k \in \mathcal{K} = \{1, 2, \cdots, M+1\}, \end{aligned} \tag{4.2.3}$$

where we have normalized β_k with respect to T. By doing so, we have turned an infinite number of variables $\boldsymbol{p}(t)$ to a finite number of variables $\boldsymbol{p}_k$ without compromising the optimality of the problem. The joint power control and

scheduling optimization problem is now equivalent to find a piecewise constant power allocation that has $M+1$ degrees of freedom in the time domain.

- **The issue of complicated feasible region**

Having transformed the problem into one with a finite number of variables, we are still confronted with the problem that the feasible region specified by (4.2.3) has a complicated shape. We address this problem by defining $\hat{U}_i(r_i)$ to be

$$\hat{U}_i(r_i) = \begin{cases} U_i(r_i) & \text{if } r_i \geq r_i^{\min}, \\ -\infty & \text{otherwise.} \end{cases} \tag{4.2.4}$$

Similar to $U_i(r_i)$, $\hat{U}_i(r_i)$ is a monotonically increasing function. With (4.2.4), Problem (P4.1) can be rewritten as

$$\begin{aligned} \underset{\boldsymbol{\beta},\{\boldsymbol{p}_k,\forall k\}}{\text{maximize}} \quad & \sum_{i=1}^{M} \hat{U}_i\Big(\sum_{k=1}^{M+1} \beta_k \log_2(1+\gamma_i(\boldsymbol{p}_k))\Big) \\ \text{subject to} \quad & \sum_{k=1}^{M+1} \beta_k \leq 1, \boldsymbol{\beta} \succeq \mathbf{0} \\ & \mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}, \forall k \in \mathcal{K}. \end{aligned} \tag{P4.2}$$

Note that Problem (P4.2) is equivalent to Problem (P4.1) when Problem (P4.1) is feasible. On the other hand, the objective function of Problem (P4.2) is equal to $-\infty$ if and only if Problem (P4.1) is infeasible.

4.2.2 Monotonic Optimization

Due to the non-convex nature of Problem (P4.2), it is impossible to find the optimal solution based on the theory of convex optimization [47]. As introduced in Chapter 3, monotonicity is another important property besides convexity that can be exploited to efficiently solve an optimization problem. In this subsection,

we first show how Problem (P4.2) can be transformed into a MO problem by exploiting its hidden monotonicity. We further discuss several key properties of the reformulation that are critical for the design of the S-MAPEL algorithm that solves the optimization problem.

Let $(\boldsymbol{\delta}, \boldsymbol{z})$ denote the concatenation of two vectors $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_{M+1})$ and $\boldsymbol{z} = (z_{11}, \cdots, z_{1M}, \cdots, z_{(M+1)1}, \cdots, z_{(M+1)M})$. Since the function $\hat{U}_i(r_i)$ is non-decreasing in r_i, it is easy to see that the function $\Phi((\boldsymbol{\delta}, \boldsymbol{z})) = \sum_{i=1}^{M} \hat{U}_i(\sum_{k=1}^{M+1} \delta_k \log_2 (1 + z_{ki}))$ is a non-decreasing function on $\mathbb{R}_+^{(M^2+2M+1)}$. That is, for any two vectors $(\boldsymbol{\delta}_1, \boldsymbol{z}_1)$ and $(\boldsymbol{\delta}_2, \boldsymbol{z}_2)$ such that $(\boldsymbol{\delta}_1, \boldsymbol{z}_1) \succeq (\boldsymbol{\delta}_2, \boldsymbol{z}_2)$, we have $\Phi((\boldsymbol{\delta}_1, \boldsymbol{z}_1)) \geq \Phi((\boldsymbol{\delta}_2, \boldsymbol{z}_2))$. We further note that $\gamma_i(\boldsymbol{p}_k)$ for all i and k is nonnegative and the time fraction vector $\boldsymbol{\beta}$ is nonnegative as well. Based on these observations, Problem (P4.2) can be rewritten as

$$\begin{aligned} \underset{(\boldsymbol{\delta},\boldsymbol{z})}{\text{maximize}} \quad & \Phi((\boldsymbol{\delta}, \boldsymbol{z})) = \sum_{i=1}^{M} \hat{U}_i(\sum_{k=1}^{M+1} \delta_k \log_2(1 + z_{ki})) \\ \text{subject to} \quad & (\boldsymbol{\delta}, \boldsymbol{z}) \in \mathcal{G}, \end{aligned} \tag{P4.3}$$

where the feasible set

$$\mathcal{G} = \left\{ (\boldsymbol{\delta}, \boldsymbol{z}) | 0 \leq \delta_k \leq \beta_k \text{ and } 0 \leq z_{ki} \leq \gamma_i(\boldsymbol{p}_k), \forall i \in \mathcal{M}, \forall k \in \mathcal{K}, (\boldsymbol{\beta}, \boldsymbol{p}) \in \mathcal{P}_\beta \right\} \tag{4.2.5}$$

with

$$\mathcal{P}_\beta = \left\{ (\boldsymbol{\beta}, \boldsymbol{p}) | \sum_{k=1}^{M+1} \beta_k \leq 1, \beta_k \geq 0 \text{ and } 0 \leq p_{k,i} \leq P_i^{\max}, \forall i \in \mathcal{M}, \forall k \in \mathcal{K} \right\}. \tag{4.2.6}$$

Here, $(\boldsymbol{\beta}, \boldsymbol{p})$ is the concatenation of vectors $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_{M+1})$ and $\boldsymbol{p} = (p_{1,1}, \cdots, p_{1,M}, \cdots, p_{M+1,1}, \cdots, p_{M+1,M})$. In the following, we establish the equivalence between Problems (P4.3) and (P4.2).

The feasible set $\mathcal{G}$ of Problem (P4.3) is essentially a union of infinite number of boxes with vertices of all boxes belonging to the set $\{(\boldsymbol{\tau}, \boldsymbol{c}) | \tau_k = \beta_k \text{ and } c_{k,i} =$

$\gamma_i(\boldsymbol{p}_k), \forall i \in \mathcal{M}, \forall k \in \mathcal{K}, (\boldsymbol{\beta}, \boldsymbol{p}) \in \mathcal{P}_\beta\}$. Thus, by Proposition 2.1, the feasible set is a normal set. This, together with the fact that $\Phi((\boldsymbol{\delta}, \boldsymbol{z}))$ is an increasing function of $(\boldsymbol{\delta}, \boldsymbol{z})$, implies that Problem (P4.3) is a MO problem. Thus, by Proposition 2.2, the optimal solution of Problem (P4.3), denoted by $(\boldsymbol{\delta}^*, \boldsymbol{z}^*)$, must occur at the upper boundary of set $\mathcal{G}$ where $\delta_k = \beta_k$ and $z_{ki} = \gamma_i(\boldsymbol{p}_k)$ for all i and k. If we can find a joint time fraction and power allocation $(\boldsymbol{\beta}^*, \boldsymbol{p}^*)$ corresponding to the optimal solution $(\boldsymbol{\delta}^*, \boldsymbol{z}^*)$ such that $\delta_k^* = \beta_k^*$ and $z_{ki}^* = \gamma_i(\boldsymbol{p}_k^*)$ for all i and k, then such $(\boldsymbol{\beta}^*, \boldsymbol{p}^*)$ is clearly the optimal solution to Problem (P4.2). Finding such $(\boldsymbol{\beta}^*, \boldsymbol{p}^*)$ requires solving $M+1$ uncoupled sets of M linear equations $z_{ki}^*(\sum_{j \neq i} G_{ji} p_{k,i}^* + n_i) = G_{ii} p_{k,i}^*$ with M variables $p_{k,1}^*, \cdots, p_{k,M}^*$. As the channel gains G_{ij}'s are generally random, we can show with probability 1 that the $M^2 + M$ equations are linearly independent, implying that there is a unique solution $(\boldsymbol{p}_1^*, \cdots, \boldsymbol{p}_{M+1}^*)$. Hence, Problems (P4.1), (P4.2) and (P4.3) are all equivalent with each other. *We will focus on how to solve Problem (P4.3) efficiently based on the recent advance in monotonic optimization in the rest of the chapter.*

Before leaving this section, note that set $\mathcal{G}$ is a non-convex set. However, convexity is not important in obtaining the global optimal solution. In the next section, we show that it is the monotonicity of the objective function and the normality of the feasible set in the reformulated Problem (P4.3) that facilitates efficient calculation of the global optimal solution.

4.3 THE S-MAPEL ALGORITHM

In this section, we propose a novel algorithm, S-MAPEL, to solve Problem (P4.3) based on the special characteristics of the problem. We first review the general MO algorithm before presenting the algorithm for joint power control and scheduling.

4.3.1 General MO Algorithm

As discussed in Chapter 3, an algorithm that solves a general MO problem P_M works as follows. We first construct a polyblock $\mathcal{S}_1$ that contains the feasible set $\mathcal{H}$ of Problem P_M. Let $\mathcal{T}_1$ denote the proper vertex set of $\mathcal{S}_1$. By Proposition 2.3, the maximum of the objective function of Problem P_M (i.e., $\Phi(\boldsymbol{x})$) over set $\mathcal{S}_1$ occurs at some proper vertex $\boldsymbol{x}_1$ of $\mathcal{S}_1$, i.e., $\boldsymbol{x}_1 \in \mathcal{T}_1$. If $\boldsymbol{x}_1$ happens to reside in set $\mathcal{H}$ as well, then it solves Problem P_M and the optimal solution $\boldsymbol{x}^*$ is equal to $\boldsymbol{x}_1$. Otherwise, based on Proposition 2.4, we can construct a smaller polyblock $\mathcal{S}_2 \subset \mathcal{S}_1$ that still contains $\mathcal{H}$ but excludes $\boldsymbol{x}_1$. This is achieved by constructing the vertex set $\mathcal{T}_2$ by replacing $\boldsymbol{x}_1$ in $\mathcal{T}_1$ with N new vertices $\{\boldsymbol{x}_{11}, \cdots, \boldsymbol{x}_{1N}\}$ and removing improper vertices, where $\boldsymbol{x}_{1j} = \boldsymbol{x}_1 - (x_{1,j} - \pi_j^{\mathcal{H}}(\boldsymbol{x}_1))\boldsymbol{e}_j$. We can repeat this procedure until an optimal solution is found. This leads to a sequence of polyblocks containing $\mathcal{H} : \mathcal{S}_1 \supset \mathcal{S}_2 \supset \cdots \supset \mathcal{H}$. Obviously, $\Phi(\boldsymbol{x}_1) \geq \Phi(\boldsymbol{x}_2) \geq \cdots \geq \Phi(\boldsymbol{x}^*)$, where $\boldsymbol{x}_n$ is the optimal solution that maximizes $\Phi(\boldsymbol{x})$ over the polyblock $\mathcal{S}_n$. The algorithm terminates at nth iteration if $\boldsymbol{x}_n \in \mathcal{H}$. This general algorithm is guaranteed to converge to a global optimal solution *only if* the optimal solution $\boldsymbol{x}^*$ has a positive lower bound $\boldsymbol{x}_l$ (i.e., $\boldsymbol{x}_l \succ \boldsymbol{0}$) [18][45].

The MAPEL algorithm proposed in Chapter 3 is one of the successful applications of MO to engineering designs in wireless networks. The MO problem formulated in Chapter 3 has a nice property that the optimal solution is lower bounded by a positive value, which guarantees the convergence of MAPEL to the global optimal solution. On the contrary, Problem (P4.3) does not impose a positive lower bound on the optimal solution $(\boldsymbol{\delta}^*, \boldsymbol{z}^*)$. In fact, some δ_k^*'s and z_{ki}^*'s can be equal to zero at the optimal solution. Consequently, the general MO algorithm described above (and hence MARL) cannot be directly applied even though Problem (P4.3) is a MO problem. In the next subsection, we propose an enhanced algorithm, referred to as S-MAPEL, to obtain the global optimal so-

lution of (P4.3) through non-trivial modifications of the general MO algorithm.

4.3.2 The S-MAPEL Algorithm

In the general MO algorithm, a critical step to construct new polyblocks is calculating the projection $\pi^{\mathcal{H}}(\boldsymbol{x}_n)$ as defined in Definition 6. To regain the convergence that was destroyed due to the absence of a positive lower bound on the optimal solution to (P4.3), S-MAPEL adopts a modified projection $\pi_{\boldsymbol{o}}(\cdot)$ where a vertex is projected onto the upper boundary of the feasible set along a line connecting the vertex and a negative point $\boldsymbol{o} \prec \mathbf{0}$. To make sure that the projection always exists, the projection region $\mathcal{G}$ is extended to a new region $\mathcal{G}' = \{(\boldsymbol{\delta}', \boldsymbol{z}') | \boldsymbol{o} \preceq (\boldsymbol{\delta}', \boldsymbol{z}') \preceq (\boldsymbol{\delta}, \boldsymbol{z}), \forall (\boldsymbol{\delta}, \boldsymbol{z}) \in \mathcal{G}\}$ without sacrificing the optimality of (P4.3). Mathematically, the modified projection is $\pi_{\boldsymbol{o}}^{\mathcal{G}'}((\boldsymbol{\delta}_n, \boldsymbol{z}_n)) = \lambda_n((\boldsymbol{\delta}_n, \boldsymbol{z}_n) - \boldsymbol{o}) + \boldsymbol{o}$, where $\lambda_n = \max\{\lambda | \lambda_n((\boldsymbol{\delta}_n, \boldsymbol{z}_n) - \boldsymbol{o}) + \boldsymbol{o}\} \in \mathcal{G}'$. This is illustrated in Fig. 4.1. Without loss of generality, set $\boldsymbol{o} = -\mathbf{1}$ hereafter. Note that the optimal solution $(\boldsymbol{\delta}^*, \boldsymbol{z}^*)$ to (P4.3) is always component-wise larger than or equal to $\mathbf{0}$. This implies that the optimal solution $(\boldsymbol{\delta}^*, \boldsymbol{z}^*)$ has a "positive" lower bound with respect to the "new origin" $-\mathbf{1}$. This crucial modification guarantees the convergence of S-MAPEL, as we will prove in the next subsection.

The projection $\pi_{-\mathbf{1}}^{\mathcal{G}'}((\boldsymbol{\delta}_n, \boldsymbol{z}_n)) = \lambda_n((\boldsymbol{\delta}_n, \boldsymbol{z}_n) + \mathbf{1}) - \mathbf{1}$ is obtained by solving the following max-min problem for λ_n:

$$\begin{aligned}
\lambda_n &= \max\left\{\lambda | \lambda((\boldsymbol{\delta}_n, \boldsymbol{z}_n) + \mathbf{1}) - \mathbf{1} \in \mathcal{G}'\right\} \\
&= \max\left\{\lambda | \lambda \leq \min_{i \in \mathcal{M}, k \in \mathcal{K}} \left\{\frac{1 + \gamma_i(\boldsymbol{p}_k)}{1 + z_{n,ki}}, \frac{1 + \beta_k}{1 + \delta_{n,k}}\right\}, (\boldsymbol{\beta}, \boldsymbol{p}) \in \mathcal{P}_\beta\right\} \\
&= \max_{(\boldsymbol{\beta}, \boldsymbol{p}) \in \mathcal{P}_\beta} \min_{i \in \mathcal{M}, k \in \mathcal{K}} \left\{\frac{1 + \gamma_i(\boldsymbol{p}_k)}{1 + z_{n,ki}}, \frac{1 + \beta_k}{1 + \delta_{n,k}}\right\}.
\end{aligned} \tag{4.3.1}$$

This is known as a generalized fractional linear programming problem and can be directly solved by the Dinkelbach-type algorithm [46] with slight modifica-

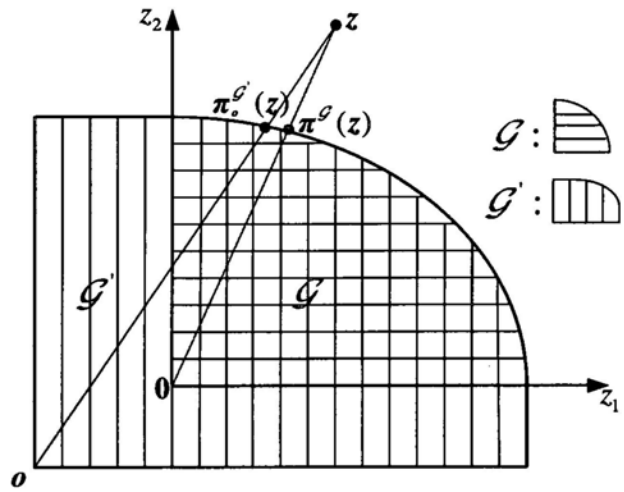


Figure 4.1: The modified projection $\pi_o^{\mathcal{G}'}(z)$.

tion. For the max-min problem (4.3.1), the Dinkelbach-type algorithm is a serial algorithm involving $(M+1)^2$ variables. A close look at (4.3.1) indicates that the terms in the max-min operation are uncoupled functions of either $\boldsymbol{\beta}$ or $\boldsymbol{p}_k$'s, but not both. Hence, we can further write (4.3.1) as follows.

$$\begin{aligned}
\lambda_n &= \max_{(\boldsymbol{\beta},\boldsymbol{p})\in\mathcal{P}_\beta} \min\Bigg\{\min_{i\in\mathcal{M}}\left\{\frac{1+\gamma_i(\boldsymbol{p}_1)}{1+z_{n,1i}}\right\},\cdots, \\
&\qquad \min_{i\in\mathcal{M}}\left\{\frac{1+\gamma_i(\boldsymbol{p}_{M+1})}{1+z_{n,(M+1)i}}\right\}, \min_{k\in\mathcal{K}}\left\{\frac{1+\beta_k}{1+\delta_{n,k}}\right\}\Bigg\} \\
&= \min\Bigg\{\max_{\boldsymbol{p}_1\in[\mathbf{0},\boldsymbol{P}^{\max}]}\min_{i\in\mathcal{M}}\frac{1+\gamma_i(\boldsymbol{p}_1)}{1+z_{n,1i}},\cdots, \\
&\qquad \max_{\boldsymbol{p}_{M+1}\in[\mathbf{0},\boldsymbol{P}^{\max}]}\min_{i\in\mathcal{M}}\frac{1+\gamma_i(\boldsymbol{p}_{M+1})}{1+z_{n,(M+1)i}}, \\
&\qquad \max_{\sum_{k=1}^{M+1}\beta_k\le 1,\boldsymbol{\beta}\succeq 0}\min_{k\in\mathcal{K}}\frac{1+\beta_k}{1+\delta_{n,k}}\Bigg\} \\
&= \min\Big\{\lambda_{n,1},\cdots,\lambda_{n,M+1},\lambda_{n,\beta}\Big\},
\end{aligned} \tag{4.3.2}$$

where $\lambda_{n,k} = \max_{\boldsymbol{p}_k\in[\mathbf{0},\boldsymbol{P}^{\max}]}\min_{i\in\mathcal{M}}\frac{1+\gamma_i(\boldsymbol{p}_k)}{1+z_{n,ki}}$ for $k=1,\cdots,M+1$, and $\lambda_{n,\beta} = \max_{\sum_{k=1}^{M+1}\beta_k\le 1,\boldsymbol{\beta}\succeq 0}$ $\min_{k\in\mathcal{K}}\frac{1+\beta_k}{1+\delta_{n,k}}$. Eqn. (4.3.2) indicates that the calculation of λ_n can be decomposed

into calculating $\lambda_{n,k}$'s and $\lambda_{n,\beta}$. This suggests the possibility of a parallel algorithm for fast computation. In particular, each $\lambda_{n,k}$ can be obtained through the Dinkelbach-type algorithm with M variables $p_{k,i}$'s, and $\lambda_{n,\beta}$ can be obtained through Dinkelbach-type algorithm with $M+1$ variables β_k's. Since $\lambda_{n,k}$'s and $\lambda_{n,\beta}$ are all the generalized fractional linear programming, their calculations are polynomial time solvable [46]. The calculations of $\lambda_{n,k}$'s and $\lambda_{n,\beta}$ are presented in Algorithm 4.1, where we have focused on an arbitrary $\lambda_{n,k}$ without loss of generality.

Having calculated $\lambda_{n,k}$'s and $\lambda_{n,\beta}$ through Dinkelbach-type Algorithm (i.e., Algorithm 4.1), we can obtain the projection $\boldsymbol{\pi}_{-\mathbf{1}}^{\mathcal{G}'}((\boldsymbol{\delta}_n, \boldsymbol{z}_n)) = \lambda_n((\boldsymbol{\delta}_n, \boldsymbol{z}_n) + \mathbf{1}) - \mathbf{1}$ with $\lambda_n = \min\{\lambda_{n,1}, \cdots, \lambda_{n,M+1}, \lambda_{n,\beta}\}$.

Algorithm 4.1 Dinkelbach-type Algorithm (for finding $\lambda_{n,k}$)

1: **Initialization**: Choose $\boldsymbol{p}_k^{(0)} \in [\mathbf{0}, \boldsymbol{P}^{\max}]$ and let $m = 0$.

2: **repeat**

3: Given $\boldsymbol{p}_k^{(m)}$, solve $\lambda_{n,k}^{(m)} = \min\limits_{1 \le i \le M} \frac{1+\gamma_i(\boldsymbol{p}_k^{(m)})}{1+z_{n,ki}}$.

4: Given $\lambda_{n,k}^{(m)}$, solve $\boldsymbol{p}_k^{(m+1)} = \underset{\boldsymbol{p}_k \in [\mathbf{0}, \boldsymbol{P}^{\max}]}{\operatorname{argmax}} \min\limits_{i \in \mathcal{M}} ((\sum\limits_{j=1}^{M} G_{ji} p_{k,j} + n_i) - \lambda_{n,k}^{(m)}(z_{n,ki}+1)(\sum\limits_{j \neq i} G_{ji} p_{k,j} + n_i))$ for all k.

5: $m = m + 1$.

6: **until** $\min\limits_{i \in \mathcal{M}} ((\sum\limits_{j=1}^{M} G_{ji} p_{k,j}^{(m-1)} + n_i) - \lambda_{n,k}^{(m-1)}(z_{n,ki}^{+}1)(\sum\limits_{j \neq i} G_{ji} p_{k,j}^{(m-1)} + n_i)) \le 0$.

7: $\lambda_{n,k} = \lambda_{n,k}^{(m-1)}$.

With the modified projection, the execution of the S-MAPEL largely follows the general MO algorithm. Specifically, in the nth iteration, the optimal vertex $(\boldsymbol{\delta}_n, \boldsymbol{z}_n)$ is selected from the vertex set $\mathcal{T}_n$ of the outer polyblock $\mathcal{S}_n$ through $(\boldsymbol{\delta}_n, \boldsymbol{z}_n) = \operatorname{argmax}\{\Phi((\boldsymbol{\delta}, \boldsymbol{z})) | (\boldsymbol{\delta}, \boldsymbol{z}) \in \mathcal{T}_n\}$. Then, the new polyblock $\mathcal{S}_{n+1}$ is

obtained by replacing (δ, z) in $\mathcal{T}_n$ with $(M+1)^2$ new vertices $\{(\delta_{n1}, z_{n1}), \cdots (\delta_{n(M+1)^2}, z_{n(M+1)^2})\}$, where $(\delta_{nj}, z_{nj}) = (\delta_n, z_n) - ((\delta_n, z_n)_j - \pi^{\mathcal{G}'}_{-\mathbf{1}j}((\delta_n, z_n)))e_j$, and removing improper vertices and all vertices (δ, z) that do not satisfy $(\delta, z) \succeq \mathbf{0}$.

S-MAPEL terminates at the nth iteration if $(\delta_n, z_n) \in \mathcal{G}$. In practice, we say $(\delta_n, z_n) \in \mathcal{G}$ when $(1+\epsilon)\Phi((\delta'_n, z'_n)) \geq \Phi((\delta_n, z_n))$, where $\epsilon > 0$ is a small positive number representing the error tolerance level, and $(\delta'_n, z'_n) \in \mathcal{G}$ is the current best feasible solution (CBS) that is known so far. Such a stopping criteria can guarantee to obtain an ϵ-optimal solution, as shown in MAPEL. In particular, CBS is updated in each iteration as $(\delta'_n, z'_n) = \text{argmax}\{\Phi(y) | y \in \{\pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_n, z_n)), (\delta'_{n-1}, z'_{n-1})\}\}$ if $\pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_n, z_n)) \succeq \mathbf{0}$, and $(\delta'_n, z'_n) = (\delta'_{n-1}, z'_{n-1})$ otherwise. The corresponding objective function value $\Phi((\delta'_n, z'_n))$ is referred to as the current best value (CBV). For initialization, $(\delta'_1, z'_1) = \pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_1, z_1))$ if $\pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_1, z_1)) \succeq \mathbf{0}$, and $(\delta'_1, z'_1) = \mathbf{0}$ otherwise. Obviously, we have $\Phi((\delta'_1, z'_1)) \leq \Phi((\delta'_2, z'_2)) \leq \cdots \leq \Phi((\delta^*, z^*))$.

Having introduced the basic operations, we now formally present S-MAPEL in Algorithm 4.2.

4.3.3 Global Convergence

With the modified projection $\pi_o(\cdot)$, Theorem 4.2 shows that S-MAPEL converges to the optimal solution of Problem (P4.3).

Theorem 4.2. *The S-MAPEL algorithm globally converges to a global optimal solution of Problem (P4.3).*

The proof of Theorem 4.2 is relegated to Section 4.8.1.

Before leaving this subsection, note that although S-MAPEL is proved to con-

Algorithm 4.2 The S-MAPEL Algorithm

1: **Initialization**: Choose the error tolerance $\epsilon > 0$, and let $n = 1$.

2: **repeat**

3: If $n = 1$, construct the initial polyblock $\mathcal{S}_1$ with vertex set $\mathcal{T}_1 = \{(\boldsymbol{b}, \boldsymbol{v})\}$, where $b_k = 1$ for all k, and

$$v_{ki} = \max_{p_k \in [\mathbf{0}, \boldsymbol{P}^{\max}]} \gamma_i(\boldsymbol{p}_k) = \frac{G_{ii} P_i^{\max}}{n_i}, \forall i \in \mathcal{M}, \forall k \in \mathcal{K}.$$

It is clear that polyblock $\mathcal{S}_1$ is a box $[\mathbf{0}, (\boldsymbol{b}, \boldsymbol{v})]$ containing $\mathcal{G}$. If $n > 1$, construct a smaller polyblock $\mathcal{S}_n$ with vertex set $\mathcal{T}_n$ by replacing (δ_{n-1}, z_{n-1}) in $\mathcal{T}_{n-1}$ with $(M+1)^2$ new vertices $\{(\delta_{n-11}, z_{n-11}), \cdots (\delta_{n-1(M+1)^2}, z_{n-1(M+1)^2})\}$ and removing improper vertices and all vertices (δ, z) which do not satisfy $(\delta, z) \succeq \mathbf{0}$. Herein, $(\delta_{n-1j}, z_{n-1j}) = (\delta_{n-1}, z_{n-1}) - ((\delta_{n-1}, z_{n-1})_j - \pi^{\mathcal{G}'}_{-1j}((\delta_{n-1}, z_{n-1})))e_j$,

4: Find (δ_n, z_n) that maximizes the objective function of Problem (P4.3) over set $\mathcal{T}_n$, i.e.,

$$(\delta_n, z_n) = \text{argmax}\{\Phi((\delta, z)) | (\delta, z) \in \mathcal{T}_n\}, \tag{4.3.3}$$

5: Find $\boldsymbol{\pi}^{\mathcal{G}'}((\delta_n, z_n))$ based on Algorithm 5.

6: Determine CBS (δ'_n, z'_n) and CBV $\Phi((\delta'_n, z'_n))$, where $(\delta'_n, z'_n) = \text{argmax}\{\Phi(y) | y \in \{\boldsymbol{\pi}^{\mathcal{G}'}_{-\mathbf{1}}((\delta_n, z_n)), (\delta'_{n-1}, z'_{n-1})\}\}$ if $\boldsymbol{\pi}^{\mathcal{G}'}_{-\mathbf{1}}((\delta_n, z_n)) \succeq \mathbf{0}$, and $(\delta'_n, z'_n) = (\delta'_{n-1}, z'_{n-1})$ otherwise. For initialization, $(\delta'_1, z'_1) = \boldsymbol{\pi}^{\mathcal{G}'}_{-\mathbf{1}}((\delta_1, z_1))$ if $\boldsymbol{\pi}^{\mathcal{G}'}_{-\mathbf{1}}((\delta_1, z_1)) \succeq \mathbf{0}$, and $(\delta'_1, z'_1) = \mathbf{0}$ otherwise.

7: $n = n + 1$.

8: **until** $(1+\epsilon)\Phi((\delta'_{n-1}, z'_{n-1})) \geq \Phi((\delta_{n-1}, z_{n-1}))$.

9: If $\Phi((\delta'_{n-1}, z'_{n-1})) > -\infty$, then compute the optimal time fraction and power allocation $(\boldsymbol{\beta}^*, \boldsymbol{p}^*)$ (i.e., optimal solution to Problem (P4.1)) by solving $\delta'_{n-1,k} = \beta^*_k$ and $z'_{n-1,ki} = \gamma_i(\boldsymbol{p}^*_k)$ for all i and k. Otherwise, there is no feasible solution of Problem (P4.1).

verge to the global optimal solution, the convergence speed is still an open problem. It is, however, proved in [18] that a MO algorithm is guaranteed to converge within finite iterations as long as $\epsilon > 0$. An advantage of the S-MAPEL algorithm is that we can trade off performance for convergence time by tuning ϵ. The smaller ϵ, the longer the algorithm runs and the more accurate the optimal solution is.

4.4 An Accelerated Algorithm for Joint Power Control and Scheduling

In S-MAPEL, the size of the vertex set $\mathcal{T}_n$ grows quickly in each iteration when M is large. This could potentially lead to long convergence time, as a best vertex has to be selected by comparing all vertices in $\mathcal{T}_n$ in each iteration. Such a problem is not avoidable if the global optimal solution is to be guaranteed, as Problem (P4.3) is essentially NP-hard. In this section, we present an accelerated algorithm A-S-MAPEL (where the prefix "A" stands for accelerated) that prevents the size of the vertex set from growing quickly with each iteration. By doing so, the convergence of the algorithm is drastically expedited. Our numerical results will show that the gap between the solution obtained by A-S-MAPEL and the global optimal solution is negligible.

The intuition behind A-S-MAPEL algorithm is as follows. Consider an optimal solution to Problem (P4.3) $(\delta^*, z^*) = (\delta_1^*, \cdots, \delta_{M+1}^*, z_{11}^*, \cdots, z_{1M}^*, \cdots, z_{(M+1)1}^*, \cdots, z_{(M+1)M}^*)$. A close look at Problem (P4.3) suggests that a new vector obtained by swapping the values of $(\delta_i^*, z_{i1}^*, \cdots, z_{iM}^*)$ and $(\delta_j^*, z_{j1}^*, \cdots, z_{jM}^*)$ for any pair of i and j is also an optimal solution. This is because the ordering of the time segments does not affect the average data rate of each user, and hence does

not affect the value of the utility functions. This inherent symmetry implies that there could exist more than one equally optimal polyblock vertex (δ_n, z_n) at each iteration of the S-MAPEL algorithm. Specifically, these equally optimal vertices would lead to the same optimality just with different ordering for the time segments. Therefore, selecting any one of these vertices at each iteration while eliminating the others would not affect the optimality of the algorithm.

One difficulty in carrying out this idea lies in the identification of symmetric vertices from all vertices that yield the same optimal value at an iteration. To address this issue, A-S-MAPEL makes a simplifying assumption that all equally optimal vertices are symmetric vertices. Denoting the equally optimal vertices at the nth iteration as $\mathcal{Z}_n = \{(\delta, z) | \Phi((\delta, z)) = \Phi((\tilde{\delta}, \tilde{z}))$ and $(\delta, z) \in \mathcal{T}_n\}$, where $(\tilde{\delta}, \tilde{z}) = \text{argmax}\{\Phi((\delta, z)) | (\delta, z) \in \mathcal{T}_n\}$. A-S-MAPEL then selects the one with the smallest difference between $\Phi((\delta, z))$ and $\Phi(\pi_{-1}^{\mathcal{G}'}((\delta, z)))$, and deletes all other vertices in $\mathcal{Z}_n$.

For practical implementation, we allow the existence of an error tolerance *tol* to further expedite the computational speed. Thus, the set $\mathcal{Z}_n$ is extended to

$$\mathcal{Z}_n = \{(\delta, z) | (1 + tol) \times \Phi((\delta, z)) \geq \Phi((\tilde{\delta}, \tilde{z}))\text{ and } (\delta, z) \in \mathcal{T}_n\}. \tag{4.4.1}$$

With the above notions, A-S-MAPEL is the same as S-MAPEL except for Step 4, which is modified as follows.

Step 4. Select (δ_n, z_n) according to

$$(\delta_n, z_n) = \text{argmin}\{\Phi((\delta, z)) - \Phi(\pi_{-1}^{\mathcal{G}'}((\delta, z))) | (\delta, z) \in \mathcal{Z}_n\}, \tag{4.4.2}$$

and delete all other vertices in $\mathcal{Z}_n$.

Similar to S-MAPEL algorithm, an advantage of the A-S-MAPEL algorithm is that we can trade off performance for convergence time by tuning ϵ and *tol*. The smaller ϵ and *tol*, the longer the algorithm runs and the more accurate the

optimal solution is. In addition, it is due to the tolerance *tol* and the assumption that all equally optimal vertices are symmetric vertices that A-S-MAPEL is a suboptimal method instead of an optimal one.

4.5 Performance Evaluation

4.5.1 Near Optimality of A-S-MAPEL

We first consider a four-link network as shown in Fig. 4.2, where $d = 5$ meters. Assume the channel gain between the transmitter node T_i and the receiver node R_j is d_{ij}^{-4}, where d_{ij} denotes the distance between the two nodes. Assume that $\boldsymbol{P}^{\max} = (1.0\ 1.0\ 1.0\ 1.0)$mW, and $n_i = 0.1\mu$W for all links. There are no minimum data rate constraints in this example. In Fig. 4.3 and Fig. 4.5, we investigate the system utilities obtained by A-S-MAPEL under different error tolerances ϵ and *tol* when $U_i(r_i) = \log r_i$ and $U_i(r_i) = \frac{1}{1+\exp(-r_i+2)}$, respectively. For comparison, the optimal system utilities are also plotted. Correspondingly, the numbers of iterations for convergence are plotted in Fig. 4.4 and Fig. 4.6, respectively.

From Figs. 4.3 and 4.5, it can be seen that the A-S-MAPEL algorithm can perform very close to the global optimal solution. For example, when $tol = 0.0005$, A-S-MAPEL obtains a system utility that is only 0.33% away from the optimum for proportional fair utility, and 0.17% away from the optimum for sigmoidal utility. On the other hand, it is not surprising to see that the algorithm performance improves with the decrease of either ϵ or *tol*. In general, the algorithm performance is not sensitive to the value of ϵ when *tol* is small enough. For example, when $tol = 0.0005$, the obtained system utility at any $\epsilon \in [0, 0.5]$ is the same for both proportional fair utility and sigmoidal utility. This observation

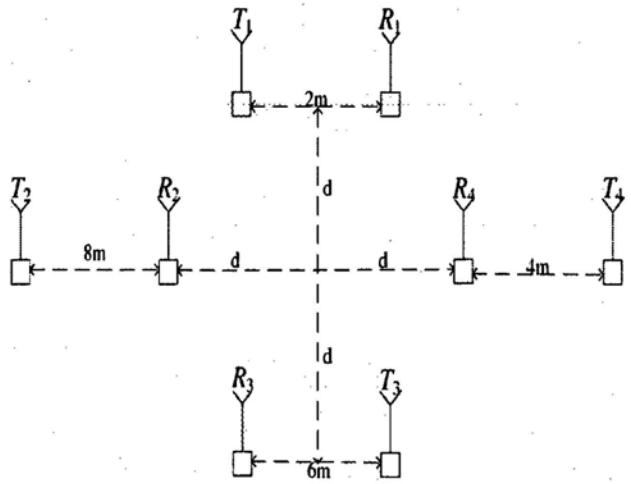


Figure 4.2: A network topology with four links.

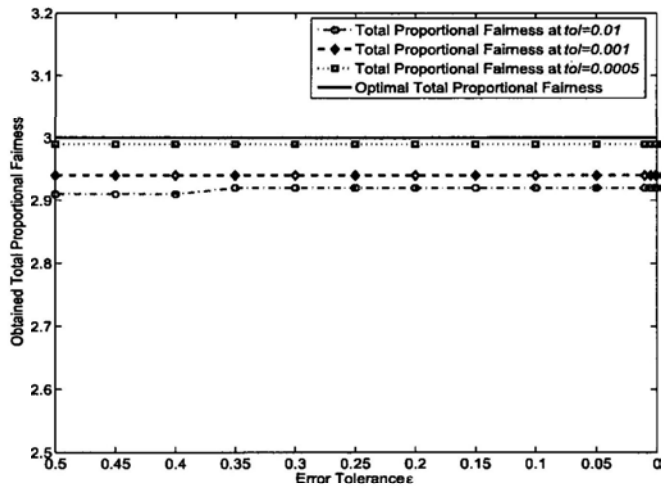


Figure 4.3: Obtained total proportional fairness for different error tolerance ϵ.

illustrates the algorithm performance can be guaranteed as long as choosing small enough ϵ and *tol*. It can be seen from Figs. 4.4 and 4.6 that the convergence time of A-S-MAPEL can be quite different for different objective functions. Regardless, the total number of iterations needed increases when either ϵ or *tol* decreases, and the change is drastic when *tol* is close to 0. Obviously, parameters ϵ and *tol* provide a tuning knob for achieving various trade-off between algorithm performance and convergence time.

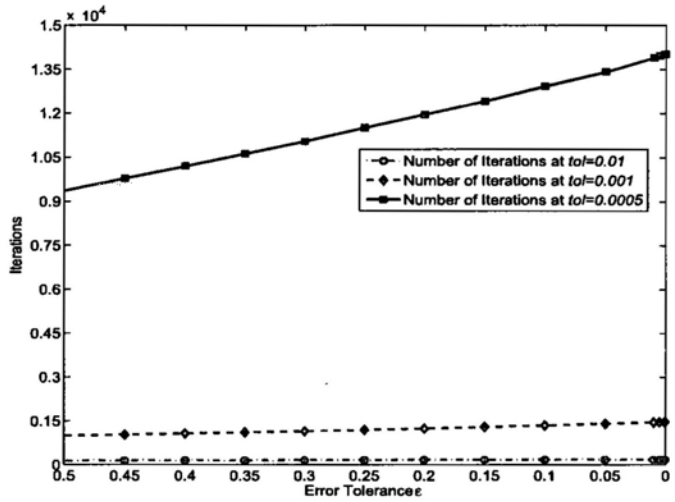


Figure 4.4: The number of iterations needed for different error tolerance ϵ when obtaining total proportional fairness.

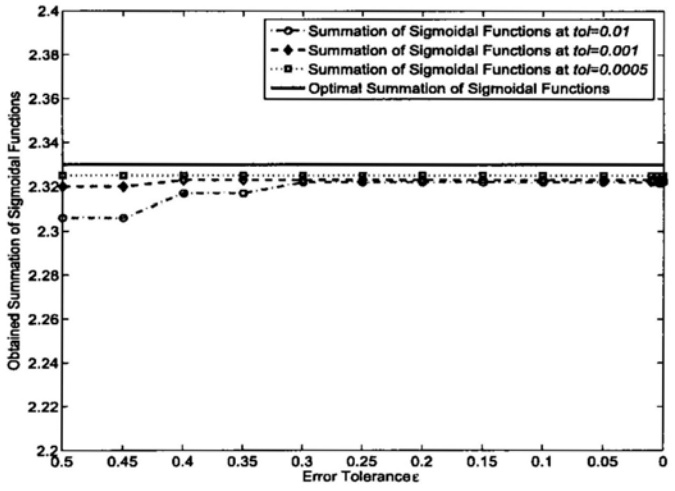


Figure 4.5: Obtained summation of sigmoidal functions for different error tolerance ϵ.

4.5.2 Optimal Joint Power Control and Scheduling vs. Node Density

In this subsection, we vary d in Fig. 4.2 to investigate the effect of node density on the optimal power control scheduling. As an example, we set $U_i(r_i) = \log r_i$,

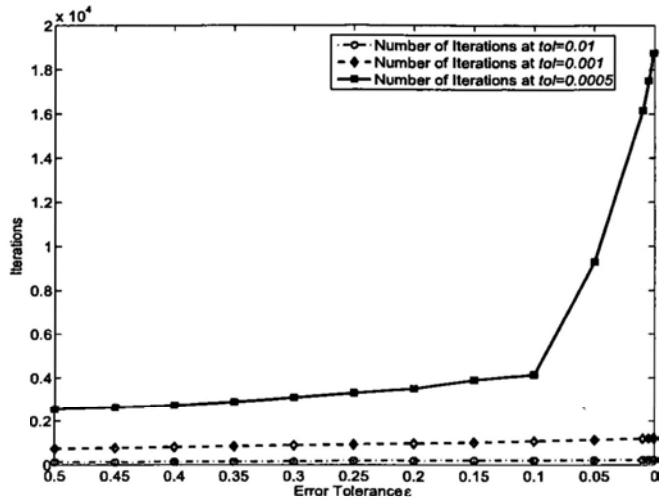


Figure 4.6: The number of iterations needed for different error tolerance ϵ when obtaining summation of sigmoidal functions.

and let the minimum data rate constraints be 1.0bps/Hz for all links. Other settings are the same as the previous example. In Fig. 4.7, we let $d = 5, 10, 15$ meters, and set the scheduling period to be 10 seconds for each d.

It is not surprising to see from Fig. 4.7 that the optimal power and scheduling heavily depends on the node density. Specifically, when the four links are close to each other (e.g., $d = 5$m, from 0 to 10 seconds), the optimal transmission power varies with time, implying that scheduling is an indispensable component in dense networks. On the other hand, scheduling is no longer necessary when the node density is small. For example, when $d = 10$m (i.e., from 10 to 20 seconds), the optimal transmission power does not vary with time any more. Furthermore, when links are enough far away from each other (e.g., $d = 15$m, from 20 to 30 seconds), it is optimal to have all links transmit at the maximum power at the same time.

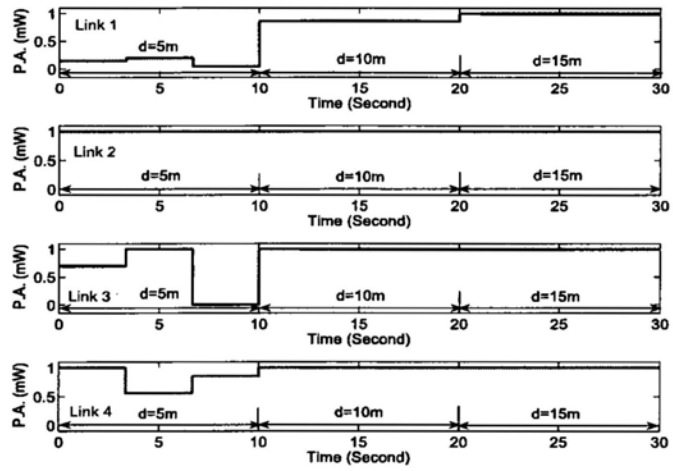


Figure 4.7: The optimal allocation vs. node density (P.A.: Power Allocation).

4.5.3 Performance Study of On-off Power Control, Pure Power Control and On-off Scheduling

One key application of S-MAPEL and A-S-MAPEL is to provide a benchmark to evaluate the performance of other schemes. As an illustration, we evaluate the performance of three widely accepted schemes in the literature, namely pure power control, on-off power control, and on-off power control with scheduling (also referred to as on-off scheduling). In particular, MAPEL is used to obtain the optimal power control solution. With on-off power control, each transmitter either transmits at the maximum power level $P_i^{\max}$ or does not transmit at all. Meanwhile, on-off scheduling is the same as joint power control and scheduling except that transmitters either transmit at the maximum power $P_i^{\max}$ or do not transmit at all. It can be seen that Theorem 1 also applies to this case, and hence no more than $M+1$ slots are needed to achieve the optimal performance of on-off scheduling.

We consider a collection of n-link networks[1]. Links are randomly placed in

[1]Here, on-off scheduling with scheduling is calculated through exhaustive search with

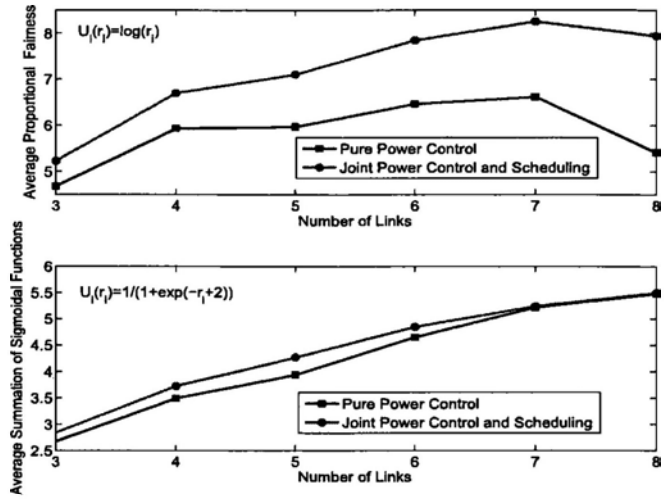


Figure 4.8: Average system performance of pure power control vs. joint power control and scheduling in n-link networks.

a 15m-by-15m area. The length of each link is uniformly distributed within [1m, 2m]. Meanwhile, set $P_i^{\max}$=1mW and $n_i = 0.1\mu$W. In Table 4.1, the performance of optimal joint power control and scheduling, pure power control, on-off power control, and on-off scheduling are given for different utility functions when $n = 3$ and $n = 4$. Besides, the performance of joint power control and scheduling and pure power control can also be found in Fig. 4.8 when n is from 3 to 8. Each value in Table II and Fig. 4.8 is an average over 50 different topologies.

It is not surprising to see that without scheduling, both power control schemes are outperformed by the ones with scheduling. This is because in a dense network, power control alone is not sufficient to eliminate strong levels of interference between close-by links. Interestingly, Fig. 4.8 shows that the performance gap between joint power control and scheduling and pure power con-

$(2^M - 1)^{M+1}$ convex programming. Thus, the calculation is unavailable when the number of links increases, which is why we only consider the network with 3 links or 4 links for on-off scheduling.

Table 4.1: Comparison of pure power control, on-off scheduling versus joint power control and scheduling

Stratigies (Algorithm)	Average Performance					
	$U_i(r_i) = \log(r_i)$		$U_i(r_i) = \frac{1}{1+e^{(-r_i+2)}}$		$U_i(r_i) = \frac{1}{1+e^{(-r_i+4)}}$	
	3-link	4-link	3-link	4-link	3-link	4-link
On-off Power Control without scheduling (Exhaustive Search)	4.4476	4.2951	2.6324	3.4315	1.7123	2.9151
Pure Power Control (MAPEL)	4.6801	5.9330	2.6750	3.4935	2.3273	2.9864
On-off Scheduling (Exhaustive Search)	5.1668	6.5933	2.8361	3.7103	2.3413	3.0047
Joint Power Control and Scheduling (A-S-MAPEL)	5.2276	6.7021	2.8450	3.7255	2.3752	3.0444

trol is wider for $U_i(r_i) = \log r_i$ than it is for $U_i(r_i) = \frac{1}{1+\exp(-r_i+2)}$ when n grows large. This is because for concave utility functions, the derivative of $U_i(r_i)$ is larger for smaller r_i. Hence, the pure power control scheme would force all links to be active but with a low data rate when the network is dense. In this case, scheduling can play an important role in improving the overall system utility. On the other hand, with sigmoidal utility function, the optimal strategy in dense networks is to deactivate some links so that the data rate of other links can exceed the threshold b_i (i.e., 2 in Fig. 4.8), no matter whether scheduling is employed or not. In this case, scheduling does not make a big difference.

Another interesting observation is that without scheduling, on-off power control may lead to a much lower system utility compared with the optimal power control solution. In contrast, the performance gap between on-off scheduling and optimal joint power control and scheduling is negligible. This is due to the fact that the links that are scheduled to transmit in the same time slot are typically far from each other and do not impose excessive interference on one another. As a result, it is likely to be optimal for the links to transmit at the maximum power level. In practice, most off-the-shelf wireless devices are only allowed to either transmit at the maximum power (i.e., be on) or remain silent (i.e., be off). Therefore, scheduling is an indispensable component for system utility maximization if "off-the-shelf" wireless devices are to be used. We conclude this section by noting that the design of efficient algorithms for optimal on-off scheduling is more challenging than that for joint power control and scheduling, due to the combinatorial nature of the on-off scheduling problem. It will be part of our future work to design efficient algorithms to solve (P4.1) when only two power levels ($P^{\max}$ and 0) are allowed.

4.6 Discussion

There has been a lot of recent work using the primal/dual decomposition approach to devise low-complexity and distributed algorithms for optimal resource allocation problems. These algorithms are guaranteed to converge to the global optimal solutions as long as the optimization problem satisfies the slater' s condition [12, 22, 48, 49]. Lemma 4.1 suggests that Problem (P4.1) is convex in terms of average data rate $\boldsymbol{r}$. Recently, there have been emerging interests in solving similar problems using primal/dual decomposition, such as [48, 49]. In this section, we will show that the primal/dual decomposition fails to achieve the global optimal solution of (P4.1) although it is convex in $\boldsymbol{r}$.

As mentioned previously, Problem (P4.1) can be rewritten as

$$\begin{aligned} \underset{\boldsymbol{r}\succeq \boldsymbol{r}^{\min},\boldsymbol{\beta},\{\boldsymbol{p}_k,\forall k\}}{\text{maximize}} \quad & \sum_{i=1}^{M} U_i(r_i) \\ \text{subject to} \quad & r_i \leq \sum_{k=1}^{M+1} \beta_k r_i(\boldsymbol{p}_k), \forall i \in \mathcal{M} \\ & \sum_{k=1}^{M+1} \beta_k = 1, \boldsymbol{\beta} \succeq \mathbf{0} \\ & \mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}, \forall k \in \mathcal{K}. \end{aligned} \tag{4.6.1}$$

Therefore, it seems that Problem (4.6.1) can be solved based on the concept of primal decomposition [48]. More specifically, consider first a primal decomposition of (4.6.1) by fixing the transmission power $\boldsymbol{p}_k$'s and the time fraction $\boldsymbol{\beta}$. Problem (4.6.1) becomes M independent subproblems, one for each link i, i.e.,

$$\begin{aligned} \underset{\boldsymbol{r}\succeq \boldsymbol{r}^{\min}}{\text{maximize}} \quad & U_i(r_i) \\ \text{subject to} \quad & r_i \leq \sum_{k=1}^{M+1} \beta_k r_i(\boldsymbol{p}_k), \forall i \in \mathcal{M}. \end{aligned} \tag{4.6.2}$$

It is easily seen from (4.6.2) that the optimal Lagrange multiplier μ_i^* associated with the constraints in (4.6.2) is equal to $\frac{\partial U_i(r_i)}{\partial r_i}|_{r_i=r_i^*(\{\boldsymbol{p}_k\},\boldsymbol{\beta})}$, where $r_i^*(\{\boldsymbol{p}_k\},\boldsymbol{\beta})$ is

the optimal solution to (4.6.2) .

Then consider the master primal problem of (4.6.1), which is

$$\begin{aligned} \underset{\boldsymbol{\beta},\{\boldsymbol{p}_k\}}{\text{maximize}} \quad & \sum_{i=1}^{M} U_i(r_i^*(\{\boldsymbol{p}_k\},\boldsymbol{\beta})) \\ \text{subject to} \quad & \sum_{k=1}^{M+1} \beta_k = 1, \boldsymbol{\beta} \succeq \mathbf{0} \\ & \mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}, \forall k \in \mathcal{K}. \end{aligned} \tag{4.6.3}$$

The master primal problem (4.6.3) can now be solved with a subgradient method by updating the transmission power and the time fraction as

$$p_{k,i}(n+1) = \left[p_{k,i}(n) + a\beta_k \sum_{j=1}^{M} \mu_j^* \left(\frac{\partial r_j(\boldsymbol{p}_k)}{\partial p_{k,i}} |_{\boldsymbol{p}_k = \boldsymbol{p}_k(n)} \right) \right]_{\mathcal{H}} \forall k, i \tag{4.6.4}$$

and

$$\beta_k(n+1) = \left[\beta_k(n) + b \sum_{i=1}^{M} r_i(\boldsymbol{p}_k(n)) \right]_{\mathcal{H}} \forall k, \tag{4.6.5}$$

where a and b are the positive step sizes, n is the number of iterations, and $[\cdot]_{\mathcal{H}}$ denotes the projection onto the feasible set $\mathcal{H}$ such that $\mathcal{H} \triangleq \{\boldsymbol{p}_{\boldsymbol{\beta}} | \sum_{k=1}^{M+1} \beta_k = 1, \boldsymbol{\beta} \succeq \mathbf{0}$, and $\mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}, \forall k \in \mathcal{K}\}$. After the updating in (4.6.4) and (4.6.5), we use the obtained transmission power and time fraction to update $\boldsymbol{\mu}^*$ again through solving (4.6.2). By doing so, a local optimal solution to Problem (4.6.1) can be achieved at least.

Note that the feasible set $\mathcal{H}$ enjoys the property of naturally decomposing into a Cartesian product $\mathcal{H} = \mathcal{B} \times \mathcal{P} \times \cdots \times \mathcal{P}$, where $\mathcal{B}$ is the set $\{\boldsymbol{\beta} | \sum_{k=1}^{M+1} \beta_k = 1, \boldsymbol{\beta} \succeq \mathbf{0}\}$ and $\mathcal{P}$ is the set $\{\boldsymbol{p} | \mathbf{0} \preceq \boldsymbol{p} \preceq \boldsymbol{P}^{\max}\}$. This implies that a local optimality of Problem (4.6.1) can be achieved with polynomial time [48]. Due to the nonconvex property of each function $\sum_{k=1}^{M+1} \beta_k r_i(\boldsymbol{p}_k)$, however, the global optimality cannot be guaranteed, even though the objective function is concave in $\boldsymbol{r}$.

In addition to primal decomposition, we might want to reduce the computational complexity of Problem (4.6.1) through resorting to dual decomposition. The key idea of dual decomposition is as follows:

At each iteration n:

- The data rates of links are determined by

$$r_i(n) = \underset{r_i \geq r_i^{\min}}{\operatorname{argmax}} [U_i(r_i) - \mu_i r_i]. \tag{4.6.6}$$

- The transmission powers of links at kth time fraction are determined by

$$\boldsymbol{p}_k(n) = \underset{\mathbf{0} \preceq \boldsymbol{p}_k \preceq \boldsymbol{P}^{\max}}{\operatorname{argmax}} \sum_{i=1}^{M} \mu_i \log_2(1 + \gamma_i(\boldsymbol{p}_k)). \tag{4.6.7}$$

- The Lagrange multiplier $\boldsymbol{\mu}$ associated with the constraints $r_i \leq \sum_{k=1}^{M+1} \beta_k r_i(\boldsymbol{p}_k)$'s in (4.6.1) are updated by

$$\mu_i(n+1) = \left[\mu_i(n) - s_i(n)\Big(\log_2(1 + \gamma_i(\boldsymbol{p}_k(n))) - r_i(n)\Big)\right]^+, \tag{4.6.8}$$

  where $s_i(n)$ is a sufficiently small positive stepsize, and $[\cdot]^+$ denotes the projection onto the nonnegative orhant (i.e., $\max(\cdot, 0)$).

As mentioned in [49], if the stepsizes are chosen such that $s_i(n) = h_n s_i^0$, $h_n \to 0$ as $n \to \infty$ and $\sum_{n=1}^{\infty} h_n = +\infty$, then $\boldsymbol{r}(n) \to \boldsymbol{r}^*$ as $t \to \infty$. It implies that after obtaining the global optimum $\boldsymbol{r}^*$ using the above dual decomposition, it is easy to achieve the optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{p}_1^*, \cdots, \boldsymbol{p}_{M+1}^*)$ of Problem (P4.1) as well, according to $r_i^* = \sum_{k=1}^{M+1} \beta_k^* \log_2(1 + \gamma_i(\boldsymbol{p}_k^*))$ for all i. However, our simulations show that the dual decomposition cannot guarantee to converge, even though we choose the stepsizes satisfying the requirements. Here, we consider a four-

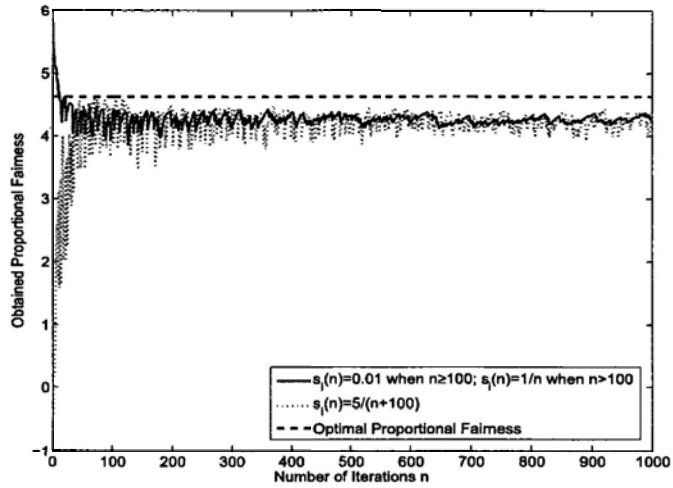


Figure 4.9: Various stepsizes are used for the dual decomposition.

link network with the following channel gain:

$$G_1 = \begin{bmatrix} 0.2321 & 0.1086 & 0.0.0671 & 0.0083 \\ 0.0124 & 0.3390 & 0.0123 & 0.0027 \\ 0.1159 & 0.0481 & 0.1579 & 0.0036 \\ 0.0030 & 0.0018 & 0.0010 & 0.2255 \end{bmatrix}. \tag{4.6.9}$$

Other system settings are the same as in Example 1. We use the dual decomposition to obtain the proportional fairness. From Fig. 4.9, it is seen that the dual decomposition cannot converge, let alone obtaining the global optimum. Accordingly, the dual decomposition is not appropriate to solve Problem (P4.1).

4.7 Summary

In this chapter, we have proposed the S-MAPEL algorithm that efficiently solves the joint power control and scheduling problem in wireless networks. S-MAPEL is guaranteed to converge to an global optimal solution despite the nonconvexity of the problem. The key idea behind the algorithm is to reformulate the non-

convex problem into a MO problem, and then construct a sequence of shrinking polyblocks that approximate the upper boundary of the feasible region with increasing precision. We have also established a convenient tradeoff between performance and convergence time of the algorithm. By exploiting the inherent symmetry of the optimal solutions, an accelerated algorithm, A-S-MAPEL, has been proposed.

Guaranteed to converge to the global optimal solution, S-MAPEL provides an important benchmark for performance evaluation of other joint power control and scheduling heuristics in this area. Using this benchmark, we find that the performance gap between on-off scheduling and joint power control and scheduling is negligible. This implies that scheduling is a crucial component in wireless system design, if off-the-shelf wireless devices that only support binary transmit power levels (on or off) are to be used.

We have briefly discussed the sub-optimality of applying decomposition methods to solve the joint power control and scheduling problem. It is shown that the primal/dual decomposition fails to achieve the global optimal solution of (P4.1) although it is convex in $\boldsymbol{r}$. We also find that the primal decomposition can be used to obtain a local optimal solution to Problem (P4.1) in polynomial time, while the dual decomposition is not guaranteed to converge.

4.8 Appendix

4.8.1 Proof of Theorem 4.2

The S-MAPEL algorithm generates a sequence $\{(\delta_n, z_n)\}$ for $n = 1, 2, \cdots$. Each component is calculated as (4.3.3) for a newly constructed polyblock. We can

find a subsequence $\{(\delta_{n_l}, z_{n_l})\}$ within the sequence $\{(\delta_n, z_n)\}$ such that

$$\begin{aligned}(\delta_{n_1}, z_{n_1}) &= (\delta_1, z_1) - ((\delta_1, z_1)_{i_0} - \pi^{\mathcal{G}'}_{-\mathbf{1}i_0}((\delta_1, z_1)))e_{i_0} \\ &\vdots \\ (\delta_{n_{l+1}}, z_{n_{l+1}}) &= (\delta_{n_l}, z_{n_l}) - ((\delta_{n_l}, z_{n_l})_{i_l} - \pi^{\mathcal{G}'}_{-\mathbf{1}i_l}((\delta_{n_l}, z_{n_l})))e_{i_l},\end{aligned} \tag{4.8.1}$$

where $1 < n_1 < n_2 < \cdots < n_l < \cdots$. $(\delta_{n_l}, z_{n_l})_{i_l}$ denotes the i_lth element of vector (δ_{n_l}, z_{n_l}), where the indices i_l denote the only position in which $(\delta_{n_{l+1}}, z_{n_{l+1}})$ differs from (δ_{n_l}, z_{n_l}). This subsequence can be thought of as the "off-springs" of vertex (δ_1, z_1) through a series of projections, and they are not necessarily adjacent since there might be projections of other vertices happening in between. It can be shown that there is at least one such subsequence that has infinite length. With a slight abuse of notation, let $\{(\delta_{n_l}, z_{n_l}), \forall l \geq 1\}$ denote such one subsequence. Since $(\delta_{n_l}, z_{n_l}) \succeq \pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_{n_l}, z_{n_l}))$, (4.8.1) together with the non-negativeness of the vertex set $\mathcal{T}_n$ implies that $(\delta_1, z_1) \succeq (\delta_{n_1}, z_{n_1}) \succeq \cdots \succeq (\delta_{n_l}, z_{n_l}) \succeq \cdots \succeq \mathbf{0}$. Hence, $\lim_{l\to\infty} \|(\delta_{n_l}, z_{n_l}) - (\delta_{n_{l+1}}, z_{n_{l+1}})\| \to 0$. From (4.8.1) we know that (δ_{n_l}, z_{n_l}) and $(\delta_{n_{l+1}}, z_{n_{l+1}})$ only differ in the i_l's position. Thus

$$\begin{aligned}&\|(\delta_{n_l}, z_{n_l}) - (\delta_{n_{l+1}}, z_{n_{l+1}})\| \\ &= (\delta_{n_l}, z_{n_l})_{i_l} - (\delta_{n_{l+1}}, z_{n_{l+1}})_{i_l} \\ &= (\delta_{n_l}, z_{n_l})_{i_l} - \pi^{\mathcal{G}'}_{-\mathbf{1}i_l}((\delta_{n_l}, z_{n_l})) \to 0 \text{ when } l \to \infty.\end{aligned} \tag{4.8.2}$$

Together with $\pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_{n_l}, z_{n_l})) = \lambda_{n_l}((\delta_{n_l}, z_{n_l}) + \mathbf{1}) - \mathbf{1}$ implying $\pi^{\mathcal{G}'}_{-\mathbf{1}i_l}((\delta_{n_l}, z_{n_l})) + 1 = \lambda_{n_l}((\delta_{n_l}, z_{n_l})_{i_l} + 1)$ and $(\delta_{n_l}, z_{n_l}) \succ \mathbf{0}$ implying $(\delta_{n_l}, z_{n_l})_{i_l} + 1 > 1$, it follows from (4.8.2) that $\lim_{l\to\infty} \lambda_{n_l} = 1$. That is,

$$\lim_{l\to\infty} (\delta_{n_l}, z_{n_l}) \to \pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_{n_l}, z_{n_l})) \tag{4.8.3}$$

and

$$\lim_{l\to\infty} \Phi((\delta_{n_l}, z_{n_l})) \to \Phi(\pi^{\mathcal{G}'}_{-\mathbf{1}}((\delta_{n_l}, z_{n_l}))). \tag{4.8.4}$$

Eqn. (4.8.3) implies that the subsequence $\{(\delta_{n_l}, z_{n_l})\}$ converges to the boundary of the region $\mathcal{G}'$ where any point (δ, z) satisfies $(\delta, z) \succeq \mathbf{0}$. Since the nonnegative boundary of the region $\mathcal{G}'$ is clearly equivalent to the upper boundary of the feasible region $\mathcal{G}$, (4.8.3) also implies that the subsequence $\{(\delta_{n_l}, z_{n_l})\}$ converges to the upper boundary of the feasible set $\mathcal{G}$. Since (δ_{n_l}, z_{n_l}) is a maximizer over the set $\mathcal{S}_{n_l}$, it is also the global optimum of Problem (P4.3) when the sequence converges. Since the CBS subsequence $\{z_{\delta'_{n_l}}\}$ satisfies $\Phi((\delta_{n_l}, z_{n_l})) \geq \Phi((\delta'_{n_l}, z'_{n_l}))$ for any n_l, it follows from (4.8.4) that

$$\lim_{l\to\infty} \Phi((\delta_{n_l}, z_{n_l})) \to \Phi(\pi_{-\mathbf{1}}^{\mathcal{G}'}((\delta_{n_l}, z_{n_l}))) \geq \lim_{l\to\infty} \Phi((\delta'_{n_l}, z'_{n_l})). \tag{4.8.5}$$

On the other hand, according to the computation of $(\delta'_{n_l}, z'_{n_l})$, we can get

$$\lim_{l\to\infty} \Phi((\delta'_{n_l}, z'_{n_l})) \geq \Phi(\pi_{-\mathbf{1}}^{\mathcal{G}'}((\delta_{n_l}, z_{n_l}))). \tag{4.8.6}$$

Together with $\pi_{-\mathbf{1}}^{\mathcal{G}'}((\delta_{n_l}, z_{n_l}))$ being the global optimal of Problem (P4.3) when $l \to \infty$, eqns. (4.8.5) and (4.8.6) imply $(\delta'_{n_l}, z'_{n_l})$ is also the global optimum of Problem (P4.3) when $l \to \infty$. Note that the S-MAPEL algorithm terminates once the optimal solution to Problem (P4.3) is found. Therefore, the convergence of the subsequence $\{(\delta_{n_l}, z_{n_l})\}$ guarantees the convergence of the algorithm to the global optimal solution. ■

Chapter 5

SEER: Globally Optimal Distributed Power Control for Nonconcave Utility Maximization

In this chapter, we address the issue of maximizing a system-wide utility through distributed power control for a single-carrier ad hoc wireless network. In particular, we propose a SEER (*aSynchronous distributEd powEr contRol*) algorithm to distributively and asynchronously achieve the global optimal power allocation. The key idea of SEER largely comes from *Gibbs Sampling*, which is a well-studied optimization algorithm in fields such as statistical physics and image processing [50, 51]. SEER has four distinctions from previous work. First, SEER maximizes the system utility function by exploring the function's entire surface, and thus it has a provable convergence to the optimal strategy that picks global optimal solutions with probability 1 for any system utility function. In particular, the system utility function is allowed to be *non-concave*, *discontinuous*, and *non-monotonic*. Second, SEER achieves the optimal power control noticeably faster than the centralized algorithm MAPEL we proposed earlier. As such, the

algorithm can efficiently handle large-scale wireless networks, as shown in our simulations. Third, SEER requires only limited message passing among links and small memory storage at each link. Last, SEER allows asynchronous power update and message passing.

The rest of this chapter is organized as follows. Section 5.1 introduces the system model and problem formulation. SEER is proposed and analyzed in Section 5.2. We evaluate the performance of SEER through several simulations in Section 5.3. We discuss the effect of message loss and message delay on the performance of SEER in Section 5.4. Section 5.5 concludes our conclusions.

5.1 System Model and Problem Formulation

We consider a snapshot of wireless ad hoc network with a set $\mathcal{M} = \{1, \cdots, M\}$ of *distinct* links. Each link consists of a transmitter node T_i and a receiver node R_i. The channel gain between node T_i and node R_j is denoted by G_{ij}, which is determined by various factors such as path loss, shadowing and fading effects. We write the channel gains into a channel matrix form $\boldsymbol{G} = [G_{ij}]$. Let p_i denote the transmission power of link i (i.e., from node T_i), with $P_i^{\max}$ being its maximum allowable value. For notational convenience, we write $\boldsymbol{p} = (p_i, \forall i \in \mathcal{M})$ and $\boldsymbol{P}^{\max} = (P_i^{\max}, \forall i \in \mathcal{M})$ as the transmission power vector and the maximum transmission power vector, respectively. Likewise, let the received noise on link i be n_i. Thus, the received SINR of link i is

$$\gamma_i(\boldsymbol{p}) = \frac{G_{ii}p_i}{\sum\limits_{j \neq i} G_{ji}p_j + n_i}, \tag{5.1.1}$$

and the corresponding data rate $r_i(\boldsymbol{p})$ calculated based on the Shannon capacity formula is $\log_2(1 + \Gamma_i\gamma_i(\boldsymbol{p}))$, where Γ_i is the SINR gap that indicates the difference between the SINR needed to achieve a certain data rate for a pirat-

ical modulation and coding scheme and the theoretical limit. Without loss of generality, we assume $\Gamma_i = 1$ hereafter.

We aim to find the optimal power allocation $\boldsymbol{p}^*$ that maximizes the overall system utility $U(\gamma_1(\boldsymbol{p}), \cdots, \gamma_M(\boldsymbol{p}))$. Mathematically, the optimal power control is formulated into the following form:

$$\begin{aligned} \textbf{UM}: \quad & \underset{\boldsymbol{p}}{\text{maximize}} \quad U(\gamma_1(\boldsymbol{p}), \cdots, \gamma_M(\boldsymbol{p})) \\ & \text{subject to} \quad 0 \leq p_i \leq P_i^{\max}, \ \forall i \in \mathcal{M}. \end{aligned} \tag{5.1.2}$$

In most previous work (e.g., [11–14]), the function $U(\cdot)$ is often assumed to be additive across links, i.e., $U(\gamma_1(\boldsymbol{p}), \cdots, \gamma_M(\boldsymbol{p})) = \sum_{i=1}^{M} U_i(\gamma_i(\boldsymbol{p}))$, with $U_i(\cdot)$ being the utility of link i. Furthermore, $U_i(\cdot)$ is assumed to be strictly increasing, twice differentiable, and strictly log-concave in the feasible SINR region. Unlike the previous work, we do not impose any assumptions on the function $U(\cdot)$. In particular, $U(\cdot)$ does not need to be additive. Besides, it can be non-concave, discontinuous, and non-monotonic. Thus, we have full freedom to choose the utility function $U(\cdot)$ that accurately reflects system performance. Interested readers are referred to [52] for some commonly used utility functions.

Due to the complicated coupling of SINR across links, Problem (**UM**) is in general non-convex even if the objective function $U(\cdot)$ is a concave function, let alone the cases with non-concave $U(\cdot)$'s. Therefore, it is difficult to find a global optimal solution efficiently even in a centralized fashion, not to mention solving the problem distributedly.

5.2 The SEER Algorithm

In this section, we propose a novel algorithm, SEER, to solve Problem (**UM**) distributedly. Based on the concept of Gibbs Sampling [50], we first derive a

Discrete-SEER algorithm in Section 5.2.2, where the prefix "Discrete" stands for discrete power control, which means each link i can only choose its transmission power level from a discrete and finite set bounded between 0 and $P_i^{\max}$. In Section 5.2.3, we extend the algorithm to continuous power control, referred to as Continuous-SEER. Some mathematical preliminaries about Gibbs Sampling will be introduced first in Section 5.2.1 before we present the algorithm.

5.2.1 Mathematical Preliminaries Related to Gibbs Sampling

Gibbs Sampling was originally introduced by Gibbs in 1902 to model physical interactions between molecules and particles. Later, it was used as an optimization algorithm to maximize the posterior mode estimate in image processing [50]. In particular, Gibbs Sampling solves a global optimization problem with the following form

$$h^* = \min_{x \in \mathcal{X}} \quad H(x), \tag{5.2.1}$$

where the variable x is a N-dim row vector with element $x_n, n = 1, \cdots, N$, the feasible domain $\mathcal{X} = \prod_{n=1}^{N} \mathcal{X}_n \subset \mathbb{R}^N$ is a compact set from the Cartesian product of the discrete sets $\mathcal{X}_n$ corresponding to x_n, and the objective function $H(x)$ does not require any stringent assumptions.

The key idea of Gibbs Sampling is that the value of each x_n is updated iteratively and asynchronously according to a probability distribution, which by itself is also adjusted at each iteration according to the observations of $x_1, \cdots, x_{n-1}$, $x_{n+1}, \cdots, x_N$. Presumably, the value of x_n that yields a smaller $H(x)$ is more likely to be picked. The details of Gibbs Sampling is given as follows.

In (5.2.2), θ reflects the degree of greediness. A very small θ causes all values of x_n to be chosen (nearly) equiprobably. On the contrary, when $\theta \to \infty$, the algorithm becomes a greedy one. Only the x_n's that minimize $H(x)$ for given

Algorithm 5.1 Gibbs Sampling [51]

1: **Initialization:** Randomly select an initial point $\boldsymbol{x} \in \mathcal{X}$.

2: **loop**

3: **for all** n's in any order **do**

4: The element x_n is updated by a sample from the probability distribution $\boldsymbol{\Lambda}_n(\boldsymbol{x}_{-n}) = (\Lambda_n(x_n|\boldsymbol{x}_{-n}), \forall x_n \in \mathcal{X}_n)$ with

$$\Lambda_n(x_n|\boldsymbol{x}_{-n}) = \frac{\exp\Big(-\theta H(x_n, \boldsymbol{x}_{-n})\Big)}{\sum\limits_{x'_n \in \mathcal{X}_n} \exp\Big(-\theta H(x'_n, \boldsymbol{x}_{-n})\Big)}, \tag{5.2.2}$$

where θ is a positive constant, and $\boldsymbol{x}_{-n} = (x_1, \cdots, x_{n-1}, x_{n+1}, \cdots, x_N)$.

5: **end for**

6: **end loop**

$\boldsymbol{x}_{-n}$ will be picked. To avoid being trapped in a local optimal solution by being too greedy, a large enough but finite θ is usually adopted. It allows the algorithm to explore non-greedy actions with small probabilities. Note that Gibbs Sampling does not make any assumptions on $H(\boldsymbol{x})$, as long as it can be evaluated. Furthermore, the updates of each element are fully asynchronous and distributed.

5.2.2 Discrete-SEER

We first consider Problem (**UM**), assuming that each p_i can only take values from a discrete set $\mathcal{P}_i = \{0, \triangle P_i, 2\triangle P_i, \cdots, P_i^{\max}\}$. In this case, Gibbs Sampling can be straightforwardly applied to solve the problem. In what follows, we present the Discrete-SEER algorithm and prove its convergence to the global optimal solution.

Rewrite Problem (**UM**) into

$$\begin{aligned} \mathbf{UM-EQD}: \quad & \underset{\boldsymbol{p}}{\text{minimize}} \quad \frac{1}{U(\boldsymbol{\gamma}(\boldsymbol{p}))} \\ & \text{subject to} \quad p_i \in \mathcal{P}_i, \ \forall i \in \mathcal{M}, \end{aligned} \tag{5.2.3}$$

where $\boldsymbol{\gamma}(\boldsymbol{p})$ is the vector of $\gamma_i(\boldsymbol{p})$. A close look at (5.2.3) shows that it is similar to (5.2.1) considered in Gibbs Sampling.

The key idea of Discret-SEER is as follows. Each link i picks a sequence of time epochs $\{t_{i1}, t_{i2}, \cdots\}$, at which its transmission power is updated. In particular, at the time epoch t_{ik}, the transmission power is updated to $p_i(t_{ik})$ according to the probability distribution $\boldsymbol{\Lambda}_i(\boldsymbol{p}_{-i}(t_{ik}-)) = (\Lambda_i(p_i|\boldsymbol{p}_{-i}(t_{ik}-)), \forall p_i \in \mathcal{P}_i)$, i.e.,

$$\Lambda_i(p_i|\boldsymbol{p}_{-i}(t_{ik}-)) = \frac{\exp\left(\frac{-\theta}{U(\boldsymbol{\gamma}(p_i,\boldsymbol{p}_{-i}(t_{ik}-)))}\right)}{\sum\limits_{p_i' \in \mathcal{P}_i} \exp\left(\frac{-\theta}{U(\boldsymbol{\gamma}(p_i',\boldsymbol{p}_{-i}(t_{ik}-)))}\right)}, \tag{5.2.4}$$

where $\boldsymbol{p}_{-i}(t_{ik}-) = (p_1(t_{ik}-), \cdots, p_{i-1}(t_{ik}-), p_{i+1}(t_{ik}-), \cdots, p_M(t_{ik}-))$ is the transmission power of other links before the time instant t_{ik}, and $\boldsymbol{\gamma}(p_i, \boldsymbol{p}_{-i}(t_{ik}-))$ is the vector of $\gamma_i(p_i, \boldsymbol{p}_{-i}(t_{ik}-))$, which is the SINR of link i under transmission power vector $(p_i, \boldsymbol{p}_{-i}(t_{ik}-))$. Note that, the vectors $\boldsymbol{\gamma}(p_i, \boldsymbol{p}_{-i}(t_{ik}-))$ and $\boldsymbol{\gamma}(p_i', \boldsymbol{p}_{-i}(t_{ik}-))$ in (5.2.4) can be calculated at the transmitter side of link i, through

$$\begin{aligned} \gamma_j(p_i, \boldsymbol{p}_{-i}(t_{ik}-)) &= \begin{cases} \frac{\gamma_j(t_{ik}-)p_i}{p_i(t_{ik}-)}, & j = i \\ \frac{s_j(t_{ik}-)}{\frac{s_j(t_{ik}-)}{\gamma_j(t_{ik}-)} + G_{ij}(p_i - p_i(t_{ik}-))}, & j \neq i, \end{cases} \\ \gamma_j(p_i', \boldsymbol{p}_{-i}(t_{ik}-)) &= \begin{cases} \frac{\gamma_j(t_{ik}-)p_i'}{p_i(t_{ik}-)}, & j = i \\ \frac{s_j(t_{ik}-)}{\frac{s_j(t_{ik}-)}{\gamma_j(t_{ik}-)} + G_{ij}(p_i' - p_i(t_{ik}-))} & j \neq i, \end{cases} \end{aligned} \tag{5.2.5}$$

where $\gamma_j(t_{ik}-)$ and $s_j(t_{ik}-) = G_{jj}p_j(t_{ik}-)$ are the received SINR and received signal power (RSP) of link j just before the time instant t_{ik}, respectively. As-

sume that each link knows the system utility function $U(\cdot)$ and has the memory of transmission power used before the time instant t_{ik} (i.e., $p_i(t_{ik}-)$). Then, all that link i needs to calculate $\mathbf{\Lambda}_i(\boldsymbol{p}_{-i}(t_{ik}-))$ are the SINRs $\gamma_j(t_{ik}-)$'s, the RSPs $s_j(t_{ik}-)$'s, and the channel gains G_{ij}'s for all links j. In particular, the SINR and the RSP of link j can be measured at the receiver side of link j. This information is broadcast to other links in a control packet every time the receiver of link j senses a change in its SINR or RSP. Moreover, assuming reciprocity of the channel, the transmitters can measure G_{ij}'s by measuring the received power of control packets. Note that the convergence of Discrete-SEER would still be guaranteed even if the information about $\gamma_j(t_{ik}-)$'s and $s_j(t_{ik}-)$'s is not updated in time. This will be discussed later.

Having introduced the basic operations, we now formally present the Discrete-SEER algorithm in Algorithm 5.2.

- **Global Convergence**

In what follows, we take into consideration the convergence of Discrete-SEER. Contrary to the convergence to a global optimal solution in deterministic algorithm design, Discrete-SEER has a provable convergence to the optimal strategy that picks global optimal solutions with probability 1. The following Theorem 5.1 shows that given θ, Discrete-SEER converges to a strategy that picks all feasible solutions to Problem **(UM)** in a unique stationary distribution. Especially, as θ becomes large, the strategy corresponds to the one that only selects each global optimal solution to Problem **(UM)** equally likely.

Theorem 5.1. *Starting from any initial power allocation* $(p(t_{11}), p(t_{21}), \cdots, p(t_{M1}))$, *Discrete-SEER corresponds to a Markov chain that converges to a stationary distribu-*

Algorithm 5.2 The Discrete-SEER Algorithm

The implementation at each transmitter node T_i

1: **Initialization:** pick a sequence of time epochs $\{t_{i1}, t_{i2}, \cdots\}$ in continuous time.

2: Choose some feasible power $p_i(t_{i1}) \in \mathcal{P}_i$. Let $k = 1$.

3: **repeat**

4: Transmit the data packet with the power level $p_i(t_{ik})$.

5: Keep sensing the control packets broadcast by receivers, and then update the information of γ_j's and s_j's.

6: $k = k + 1$.

7: Update the feasible power $p_i(t_{ik}) \in \mathcal{P}_i$ according to the probability distribution given in (5.2.4).

8: **until** Link i decides to leave the network

The implementation at each receiver node R_i

1: **repeat**

2: Keep measuring the SINR and the RSP, and broadcast them in a control packet when a change in SINR or RSP is sensed.

3: **until** Link i leaves the network

tion $\mathbf{\Omega} = (\Omega(\boldsymbol{p}), \forall \boldsymbol{p} \in \mathcal{P})$, *i.e.,*

$$\Omega(\boldsymbol{p}) = \frac{\exp\Big(-\theta/U(\gamma(\boldsymbol{p}))\Big)}{\sum\limits_{\boldsymbol{p}' \in \mathcal{P}} \exp\Big(-\theta/U(\gamma(\boldsymbol{p}'))\Big)}, \tag{5.2.6}$$

where $\mathcal{P} = \{\boldsymbol{p} | p_i \in \mathcal{P}_i, \forall i\}$. *When* $\theta \to \infty$, $\Omega(\boldsymbol{p})$ *becomes*

$$\lim_{\theta \to \infty} \Omega(\boldsymbol{p}) = \begin{cases} \frac{1}{|\mathcal{P}^*|} & \text{if } \boldsymbol{p} \in \mathcal{P}^*, \\ 0 & \text{otherwise,} \end{cases} \tag{5.2.7}$$

where $\mathcal{P}^*$ *is the set of global optimal solutions to Problem* (**UM**), *and* $|\mathcal{P}^*|$ *denotes the cardinality of* $\mathcal{P}^*$. *In other words, Discrete-SEER converges to an optimal strategy that selects the global optimal power allocation with probability 1. When there are several equally good global optimal solutions, they are selected equally likely.*

Proof: A close look at the Discrete-SEER algorithm shows that a transition from one transmission power vector $\boldsymbol{p}_1$ to another $\boldsymbol{p}_2$ occurs only when some link, say link i, updates its transmission power according to its observation on the other entries of $\boldsymbol{p}_1$. In other words, the transition does not depend on the power vectors before $\boldsymbol{p}_1$. Hence, Discrete-SEER can be modeled as a Markov chain whose transition matrix corresponding to link i's update is defined as $\mathbf{\Pi}_i = [\Pi_i(\boldsymbol{p}_1, \boldsymbol{p}_2), \forall \boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{P}]$, where

$$\Pi_i(\boldsymbol{p}_1, \boldsymbol{p}_2) = \begin{cases} \Lambda_i(p_{2,i} | \boldsymbol{p}_{1,-i}) & \text{if } \boldsymbol{p}_{1,-i} = \boldsymbol{p}_{2,-i}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2.8}$$

Here, $\boldsymbol{p}_{1,-i} = (p_{1,1}, \cdots, p_{1,i-1}, p_{1,i+1}, \cdots, p_{1,M})$ is the transmission power vector of link i's opponents, and $\boldsymbol{p}_{2,-i}$ is defined likewise. Due to the lack of a central clock in the ad hoc network, each link schedules their power updates at arbitrary time epochs in continuous time. Therefore, there are no simultaneous

updates with probability 1. The resulting Markov chain is actually a composition of single-link transition probabilities. To prove the former part of Theorem 5.1, we just need to prove that for any updating order, the Markov chain specified in (5.2.8) has a stationary distribution $\mathbf{\Omega}$.

A close look at (5.2.8) reveals that $\prod_{i=1}^{M} \mathbf{\Pi}_i$ is strictly positive, which implies that the Markov chain is ergodic. Therefore, its stationary distribution exists. Given an arbitrary updating order $\{i_1, i_2, i_3, \cdots\}$, where the sequence is a permutation of $\{1, 2, \cdots, M\}$ with repetition, the distribution $\mathbf{\Omega}$ given in (5.2.6) always satisfies

$$\mathbf{\Omega} \cdot \mathbf{\Pi}_{i_1} \cdot \mathbf{\Pi}_{i_2} \cdot \mathbf{\Pi}_{i_3} \cdots = \underbrace{\underbrace{\underbrace{\mathbf{\Omega} \cdot \mathbf{\Pi}_{i_1}}_{\mathbf{\Omega}} \cdot \mathbf{\Pi}_{i_2}}_{\mathbf{\Omega}} \cdot \mathbf{\Pi}_{i_3}}_{\mathbf{\Omega}} \cdots = \mathbf{\Omega}. \tag{5.2.9}$$

Consequently, $\mathbf{\Omega}$ is the stationary distribution of the Markov chain (i.e., the stationary distribution of Discrete-SEER) for any updating order.

Next, we prove the latter part of Theorem (5.1). Let $U^* = \max_{\boldsymbol{p} \in \mathcal{P}} \sum_{i \in \mathcal{M}} U(\gamma(\boldsymbol{p}))$ and $\triangle U(\boldsymbol{p}) = \frac{1}{U(\gamma(\boldsymbol{p}))} - \frac{1}{U^*}$. Then,

$$\begin{aligned}
\Omega(\boldsymbol{p}) &= \frac{\exp\Big(-\theta / U(\gamma(\boldsymbol{p}))\Big)}{\sum_{\boldsymbol{p}' \in \mathcal{P}} \exp\Big(-\theta / U(\gamma(\boldsymbol{p}'))\Big)} \\
&= \frac{\exp\Big(-\theta \triangle U(\boldsymbol{p})\Big)}{|\mathcal{P}^*| + \sum_{\boldsymbol{p}' : U(\gamma(\boldsymbol{p}')) < U^*} \exp\Big(-\theta \triangle U(\boldsymbol{p}')\Big)} \\
&\xrightarrow[\theta \to \infty]{} \begin{cases} \frac{1}{|\mathcal{P}^*|} & \text{if } \boldsymbol{p} \in \mathcal{P}^* \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{5.2.10}$$

Consequently, the latter part of Theorem 5.1 follows. ■

Remark 5.1. *When $\triangle P_i$ is set to be $P_i^{\max}$ for each link, Problem (**UM**) reduces to*

on-off power control for the system utility maximization.

We note that when $\triangle P_i$ becomes very small (close to zero), Discrete-SEER asynchronously approximates continuous power allocation, where p_i is any real number in $[0, P_i^{\max}]$. However, one consequence of having small $\triangle P_i$'s is that each link must have an excessively large memory space to store the probability distribution $\boldsymbol{\Lambda}_i(\boldsymbol{p}_{-i}(t_{ik}-))$. To avoid this stringent requirement, we are motivated to propose a continuous power allocation scheme that does not need a large memory space.

5.2.3 Continuous-SEER

We now consider Continuous-SEER, where each link performs continuous power allocation for the system utility maximization. Correspondingly, Problem (**UM**) is converted into

$$\begin{aligned} \mathbf{UM-EQC}: \quad & \underset{\boldsymbol{p}}{\text{minimize}} \quad \frac{1}{U(\gamma(\boldsymbol{p}))} \\ & \text{subject to} \quad 0 \le p_i \le P_i^{\max}, \ \forall i \in \mathcal{M}. \end{aligned} \tag{5.2.11}$$

To avoid the excessively large memory space required by Discrete-SEER, we note that $\Lambda_i(p_i|\boldsymbol{p}_{-i}(t_{ik}-))$ can be written into the following form when $\triangle P_i \to 0$, i.e.,

$$\begin{aligned} & \Lambda_i(p_i|\boldsymbol{p}_{-i}(t_{ik}-)) \\ & = \lim_{\triangle P_i \to 0} \frac{\exp\left(\frac{-\theta}{U(\gamma(p_i,\boldsymbol{p}_{-i}(t_{ik}-)))}\right)\triangle P_i}{\triangle P_i \sum\limits_{p_i' \in \mathcal{P}_i} \exp\left(\frac{-\theta}{U(\gamma(p_i',\boldsymbol{p}_{-i}(t_{ik}-)))}\right)} \\ & = \lim_{\triangle P_i \to 0} \frac{\exp\left(\frac{-\theta}{U(\gamma(p_i,\boldsymbol{p}_{-i}(t_{ik}-)))}\right)\triangle P_i}{\int_0^{P_i^{\max}} \exp\left(\frac{-\theta}{U(\gamma(p_i',\boldsymbol{p}_{-i}(t_{ik}-)))}\right) dp_i'}. \end{aligned} \tag{5.2.12}$$

This can be transformed into a probability density function (PDF)

$$f_i(p_i|\boldsymbol{p}_{-i}(t_{ik}-)) = \lim_{\triangle P_i \to 0} \frac{\Lambda_i(p_i, \boldsymbol{p}_{-i}(t_{ik}-))}{\triangle P_i}$$
$$= \begin{cases} \dfrac{\exp\left(\frac{-\theta}{U(\gamma(p_i, \boldsymbol{p}_{-i}(t_{ik}-)))}\right)}{\int_0^{P_i^{\max}} \exp\left(\frac{-\theta}{U(\gamma(p_i', \boldsymbol{p}_{-i}(t_{ik}-)))}\right) dp_i'}, p_i \in \mathcal{P}_i \\ 0, \quad \text{otherwise}, \end{cases} \tag{5.2.13}$$

where $\mathcal{P}_i = \{p_i | 0 \leq p_i \leq P_i^{\max}\}$. With (5.2.13), Continuous-SEER works as follows: each link i updates its feasible transmission power $p_i \in \mathcal{P}_i$ at each time instant t_{ik} with the PDF (5.2.13). Thus, we can avoid the requirement for memory space. The other necessary interactions for each link i are the same as that needed in Discrete-SEER. Thus, the algorithm of Continuous-SEER is the same as Discrete-SEER except for Step 7, which is modified as follows.

Step 7. Choose some feasible power $p_i(t_{ik}) \in \mathcal{P}_i$ according to the PDF given in (5.2.13).

- **Global Convergence**

The following Theorem 5.2 shows that the optimal solution to Problem (**UM**) can be achieved through Continuous-SEER with large θ.

Theorem 5.2. *Starting from any initial power allocation $\boldsymbol{p}(1)$, Continuous-SEER corresponds to a Markov chain that converges to a stationary joint PDF*

$$f(\boldsymbol{p}) = \begin{cases} \dfrac{\exp\left(-\theta/U(\gamma(\boldsymbol{p}))\right)}{\int_0^{P^{\max}} \exp\left(-\theta/U(\gamma(\boldsymbol{p}'))\right) d\boldsymbol{p}'}, \boldsymbol{p} \in \mathcal{P} \\ 0, \quad \textit{otherwise} \end{cases} \tag{5.2.14}$$

where $\mathcal{P} = \{\boldsymbol{p} | 0 \leq p_i \leq P_i^{\max}, \forall i \in \mathcal{M}\}$. When $\theta \to \infty$, $f(\boldsymbol{p})$ becomes

$$\lim_{\theta \to \infty} f(\boldsymbol{p}) = \frac{1}{|\mathcal{P}^*|} \delta_{\boldsymbol{p}}(\mathcal{P}^*), \tag{15}$$

where the function $\delta_{\boldsymbol{p}}(\mathcal{P}^*)$ satisfies

$$\delta_{\boldsymbol{p}}(\mathcal{P}^*) = \begin{cases} +\infty, \text{ if } \boldsymbol{p} \in \mathcal{P}^* \\ 0, \quad \text{otherwise} \end{cases} \tag{5.2.15}$$

and $\int_{\mathbf{0}}^{\mathbf{p}^{\max}} \delta_{\boldsymbol{p}}(\mathcal{P}^*) d\boldsymbol{p} = |\mathcal{P}^*|$. *In other words, Continuous-SEER converges to an optimal strategy that only selects global optimal power allocation equally likely.*

The proof of Theorem 5.2 is similar to that of Theorem 5.1, and thus omitted here.

5.3 Simulation Results

In this section, we conduct simulations to illustrate the effectiveness of SEER. Due to space limitation, we focus on Continuous-SEER in this section.

5.3.1 Effect of θ

This example is to observe the effect of θ on the performance of Continuous-SEER. We consider a six-link network where the links are randomly placed in a 10m-by-10m area. Let the channel gains G_{ij} to be equal to d_{ij}^{-4}, where d_{ij} denotes the distance between the transmitter node T_i and the receiver node R_j. The resultant channel gain matrix is

$$G = \begin{bmatrix} 0.2595 & 0.0014 & 0.0180 & 0.0553 & 0.0010 & 0.0116 \\ 0.0124 & 0.4886 & 0.0011 & 0.0024 & 0.0018 & 0.0123 \\ 0.0055 & 0.0016 & 0.2601 & 0.6455 & 0.0160 & 0.1787 \\ 0.0250 & 0.0009 & 0.1677 & 0.5629 & 0.0016 & 0.0147 \\ 0.0020 & 0.0025 & 0.0107 & 0.0153 & 0.7786 & 0.1091 \\ 0.0048 & 0.0158 & 0.0049 & 0.0107 & 0.0454 & 0.6347 \end{bmatrix}.$$

For this realization, we run Continuous-SEER with different θ, and plot the attained system utility as well as the number of iterations needed for convergence in Figs. 5.1 and 5.2. In particular, $U(\gamma(\boldsymbol{p})) = \sum_{i=1}^{M} \log(\gamma_i(\boldsymbol{p}))$ in Fig. 5.1 and $U(\gamma(\boldsymbol{p})) = \sum_{i=1}^{M} \log_2(1+\gamma_i(\boldsymbol{p}))$ in Fig. 5.2. Assume that $P_i^{\max} = 1.0\text{mW}$ and $n_i = 0.1\mu\text{W}$ for all links i. In these two figures, the optimal system utility achieved by MAPEL is also plotted for comparison.

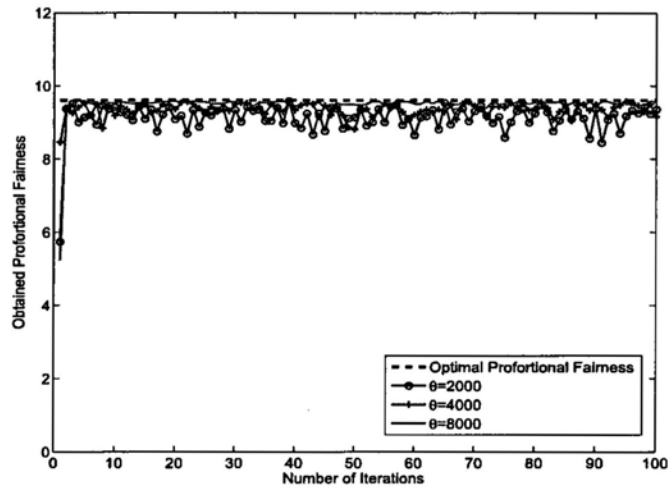


Figure 5.1: Obtained Proportional Fairness and number of iterations for different θ.

From these two figures, we see that both the frequency and the amplitude of oscillations decrease when θ increases. This result is due to the fact that θ reflects the degree of greediness. With a small θ, Continuous-SEER converges to a strategy in which links are more willing to explore power allocations other than the optimal ones. Therefore, there are more oscillations when the algorithm converges. On the other hand, with a large enough θ, the oscillation diminishes. In this case, Continuous-SEER converges to a strategy that only picks the global optimal power allocations. Besides, it can be seen from these two figures that Continuous-SEER achieves the optimal power control noticeably fast (i.e.,

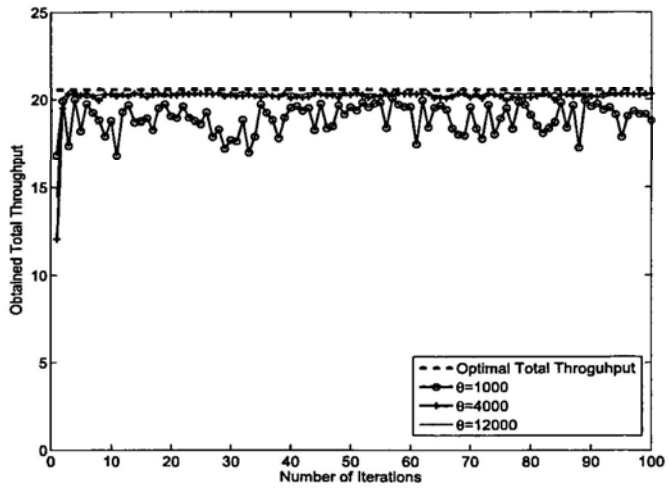


Figure 5.2: Obtained Total Throughput and number of iterations for different θ.

the number of iterations needed for the convergence to the optimal strategy is smaller than 10).

The above figures suggest that a large θ is preferred than a small θ in terms of the quality of the solution when the algorithm converges. In fact, as Theorem 5.2 states, the algorithm converges to a point where only global optimal solutions are selected with non-zero probability when θ is very large. In practice, however, θ also has an effect on the convergence time. A large θ may lead to slower convergence, as discussed in [50]. On the other hand, too large a θ may lead to numerical problems when calculating the probabilities in (5.2.4) or (5.2.13). More details about the selection of θ will be discussed in the next section.

5.3.2 Comparison with Existing Distributed Algorithms

As we mentioned in Introduction, the current exiting distributed power control algorithms only apply to special utility functions that are log-concave in

SINR. Otherwise, when the system utility function is not log-concave in SINR, these algorithms may converge to a local optimal solution. Here, we compare the performance of one "representative" algorithm (i.e., Asynchronous Distributed Pricing (ADP) Algorithm [13]) in this area. The purpose is to show that Continuous-SEER always achieves the global optimality even though the utility function is not log-concave in SINR, but ADP fails to do so. Besides, Continuous-SEER does not have a higher message passing complexity, compared with ADP.

In this subsection, we simulate both algorithms with 100 random initializations for a fifteen-link network, where the links are randomly placed in a 20m-by-20m area. Other system parameters are the same as in the above example. Suppose that $U(\gamma(\boldsymbol{p})) = \sum_{i=1}^{M} \log_2(1+\gamma_i(\boldsymbol{p}))$. Fig. 5.3 shows that Continuous-SEER can guarantee the global optimality regardless of the initial power allocation, as long as *theta* is sufficiently large. On the other hand, the ADP algorithm is always trapped in local optimal solutions for the considered fifteen-link network. This is not surprising, as ADP only applies to system utility functions that are log-concave in SINR.

Fig. 5.4 shows that Continuous-SEER not only converges much faster than the centralized algorithm MAPEL, but also has a convergence speed similar to ADP. We simulate a fifteen-link network and the simulation parameters are the same as Fig. 5.3. The figure shows that Continuous-SEER converges as fast as ADP (i.e., both within a few iterations). Due to the fact that ADP and Continuous-SEER need similar message passing at each iteration, the similar convergence speed between Continuous-SEER and ADP implies that the message passing complexity of Continuous-SEER is similar to that of ADP. Moreover, we find in Fig. 5.4 that, Continuous-SEER is guaranteed to converge to the global optimal solution, while ADP is trapped in a local optimal solution. On

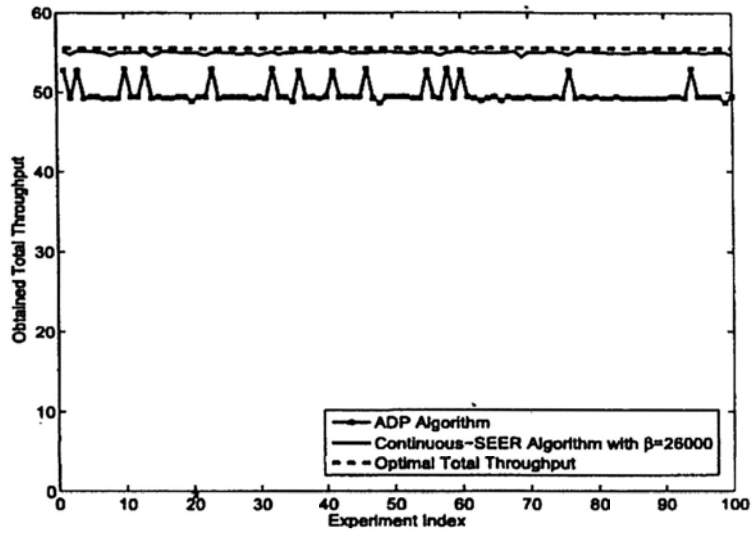


Figure 5.3: Maximal Total throughput achieved by Continuous-SEER algorithm as well as ADP algorithm for 100 different initial feasible power allocations.

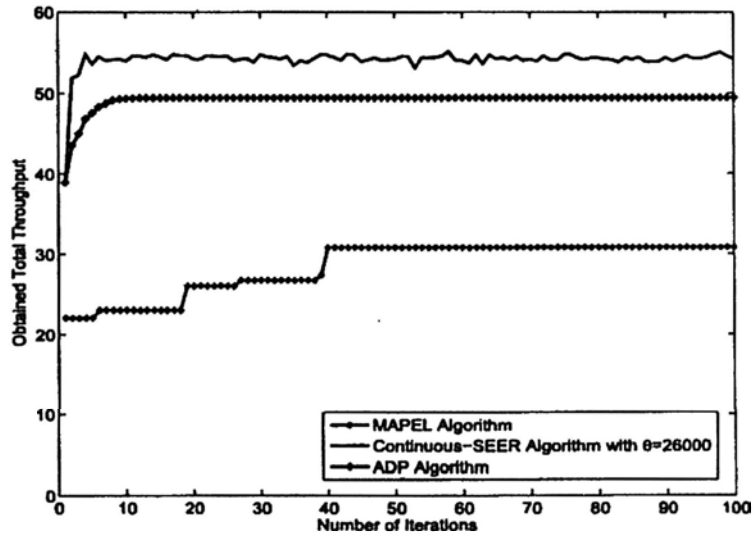


Figure 5.4: The complexity comparison between Continuous-SEER and MAPEL.

the other hand, Fig. 5.4 also shows that MAPEL is far from convergence even with 100 iterations. This implies that Continuous-SEER is a preferable choice for fast convergence in large-scale networks even if there exists a centralized controller.

5.4 Discussion

In this section, we first discuss how to select θ, and then observe the effect of message loss and message delay on the performance of Continuous-SEER.

5.4.1 Selection of θ

As already mentioned above, a large θ is preferable in terms of the quality of the solution when the algorithm converges. Especially, when θ becomes infinite, the algorithm converges to an optimal strategy that only picks each global optimal solution equally likely. However, θ also has an effect on the convergence time. In particular, given too large a fixed θ, the algorithm may have a slow convergence to the optimal strategy. This is because too large a θ may lead to the result that the algorithm is unwilling to explore solutions that does not appear to be good when it happens to reside in a local optimal solution. Therefore, to avoid being trapped in a local optimal solution by being too greedy, θ should be appropriately selected instead of using a fixed large θ.

In particular, the choice of θ involves first using a small value at the beginning of the algorithm, and then repeatedly increasing the value. This is because that first a small θ makes the algorithm far away from the local optimal solutions, and then a large θ expedites the convergence to the global optimality. In the literature [53], the recipes of selecting θ include:

- Exponential Schedule:

$$\theta(k) = \frac{\theta_0}{\alpha^k}, \tag{5.4.1}$$

  where k is the number of iterations for each link, θ_0 is a small initial value, and α is a constant satisfying $0 < \alpha < 1$.

- Logarithmic Scheduling

$$\theta(k) = \frac{\log(1+k)}{c}, \tag{5.4.2}$$

where c is a positive constant independent of k.

Based on these two schemes, the solutions generated by the algorithm converges to those of global optimal solutions with probability converging to one.

On the other hand, too large a θ may lead to numerical problems when calculating the probabilities in (5.2.4) or (5.2.13). Specifically, the numerical problems largely come from the constraint of computational precision of the simulation software. To avoid the numerical problems, a reasonable θ is necessary. However, in a random infrastructureless wireless network, a reasonable θ is generally difficult to guarantee. Therefore, it is crucial to adaptively adjust θ along with the update of transmission power. In what follows, we will show that SEER still can be guaranteed to converge to the global optimality even though each link adaptively chooses θ at each iteration.

For the practical implementation, at the time epoch t_{ik}, link i first determines a reasonable $\theta(k)$ based on $\boldsymbol{p}(t_{ik}-)$ and (5.4.1) or (5.4.2), and then updates its transmission power to $p_i(t_{ik})$ according to the distribution

$$\Lambda_i(p_i|\boldsymbol{p}_i(t_{ik}-)) = \frac{\exp\left(\frac{-\theta(k)}{U(\gamma(p_i,\boldsymbol{p}_{-i}(t_{ik}-)))}\right)}{\sum\limits_{p_i'\in\mathcal{P}_i}\exp\left(\frac{-\theta(k)}{U(\gamma(p_i',\boldsymbol{p}_{-i}(t_{ik}-)))}\right)}, \tag{5.4.3}$$

for Discrete-SEER or according to the PDF

$$f_i(p_i|\boldsymbol{p}_i(t_{ik}-)) = \begin{cases} \dfrac{\exp\left(\frac{-\theta(k)}{U(\gamma(p_i,\boldsymbol{p}_{-i}(t_{ik}-)))}\right)}{\int_0^{p_i^{\max}}\exp\left(\frac{-\theta(k)}{U(\gamma(p_i',\boldsymbol{p}_{-i}(t_{ik}-)))}\right)dp_i'}, p_i\in\mathcal{P}_i \\ 0, \quad \text{otherwise}, \end{cases} \tag{5.4.4}$$

for Continuous-SEER. In particular, $\theta(k)$ is set to be

$$\theta(k) = \min\{U(\gamma(\boldsymbol{p}(t_{ik}-))) \times K, \frac{\theta_0}{\alpha^k}\} \tag{5.4.5}$$

or

$$\theta(k) = \min\{U(\gamma(\boldsymbol{p}(t_{ik}-))) \times K, \frac{\log(1+k)}{c}\} \tag{5.4.6}$$

according to the recipe of selecting θ. Herein, K is a parameter reflecting the computational precision of the software used to run SEER. In general, there exists a limit $\hat{K}$ for K, which implies that the numerical problems is possible to happen once $K > \hat{K}$. This is because of the fact that too large a K leads to (5.4.3) or (5.4.4) being $\frac{0}{0}$ due to the computational precision of software itself. Hence, to avoid numerical problems, K has to not be larger than $\hat{K}$, which can ensure the denominator of (5.4.3) or (5.4.4) to not be equal to zero. We assume that each link is equipped with the same software, and thus it is reasonable to set the same K for each link. Compared with a fixed θ, adaptively adjusting θ facilitates the achievement of a reasonably large but finite θ for each link. More importantly, taking Discrete-SEER for example, we can prove that, with an adaptive θ, Discrete-SEER still corresponds to an ergodic Markov chain that converges to a stationary distribution $\boldsymbol{\Omega}$ satisfying

$$\boldsymbol{\Omega} \cdot \boldsymbol{\Pi}_1 \cdot \cdots \cdot \boldsymbol{\Pi}_M = \boldsymbol{\Omega}, \tag{5.4.7}$$

where $\boldsymbol{\Pi}_i = [\Pi_i(\boldsymbol{p}_1, \boldsymbol{p}_2), \forall \boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{P}]$ with

$$\Pi_i(\boldsymbol{p}_1, \boldsymbol{p}_2) = \begin{cases} \Lambda_i(p_{2,i}|\boldsymbol{p}_{1,i}) & \text{if } \boldsymbol{p}_{1,-i} = \boldsymbol{p}_{2,-i}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.4.8}$$

The following simulation shows that as a large enough K, but smaller than $\hat{K}$[1], is chosen, SEER converges to the global optimal solution to Problem (**UM**).

[1]In this thesis, Matlab 7.0 is used as the simulation software, whose $\hat{K}$ is equal to 745.

We consider a eight-link network where the links are randomly placed in a 10m-by-10m area. Let the channel gains G_{ij} to be equal to d_{ij}^{-4}, and the resultant channel gain matrix is

$$G = \begin{bmatrix} 0.7648 & 0.0046 & 0.0086 & 0.0052 & 0.0131 & 0.0040 & 0.0024 & 0.0110 \\ 0.2713 & 0.0848 & 0.0388 & 0.0240 & 0.0392 & 0.0285 & 0.0230 & 0.0168 \\ 0.0931 & 0.0058 & 0.5152 & 0.1668 & 0.0020 & 0.0017 & 0.0053 & 0.0014 \\ 0.0160 & 0.1104 & 0.4544 & 0.8802 & 0.0034 & 0.0066 & 0.2532 & 0.0020 \\ 0.0297 & 0.3983 & 0.0132 & 0.0105 & 0.1106 & 0.2520 & 0.0432 & 0.0311 \\ 0.0016 & 0.0079 & 0.0009 & 0.0008 & 0.0552 & 0.1794 & 0.0033 & 0.0386 \\ 0.0030 & 0.0427 & 0.0248 & 0.0459 & 0.0015 & 0.0039 & 0.3772 & 0.0009 \\ 0.0026 & 0.0077 & 0.0011 & 0.0010 & 0.3460 & 0.1628 & 0.0030 & 0.2852 \end{bmatrix}.$$

The remaining setting is the same as that in the first simulation of this chapter. For this realization, based on the exponential schedule and different K, we run Continuous-SEER with adaptively adjusting θ, and plot the attained total throughput as well as the number of iterations needed for convergence in Fig. 5.5. In Fig. 5.5, the optimal total throughput achieved by MAPEL is also plotted for comparison.

From Fig. 5.5, we see that both the frequency and the amplitude of oscillations decrease when K increases. This result is due to the fact that K reflects the degree of greediness. With a small K, links are more willing to explore power allocations other than the optimal ones. Therefore, there are more oscillations when the algorithm converges. On the other hand, with a large enough K, the oscillation diminishes. In this case, Continuous-SEER converges to a strategy that only picks the global optimal power allocations.

On the other hand, we know that the numerical problem may appear when Continuous-SEER uses a fixed θ. This is because that the choice of θ heavily depends on the network topology. It is difficult for each link to estimate the

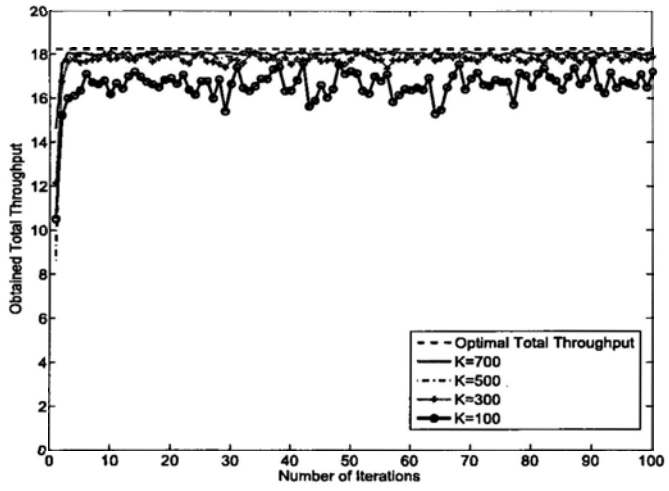


Figure 5.5: Obtained Total Throughput and number of iterations for different adaptive θ.

whole network topology, and thus a θ resulting in numerical problems is chosen with high probability. On the contrary, the adoption of adaptive θ can avoid the numerical problems, because (5.4.5) can avoid too large θ, as long as K does not exceed the computational precision $\hat{K}$ of software. As the matter of fact, the case of K being smaller than $\hat{K}$ can easily be guaranteed.

5.4.2 Effect of Message Delay

In this subsection, we observe the effect of message loss on the convergence and performance of SEER. Due to the existence of propagation delay, transmitters generally cannot be informed the information of SINR and RSP in time when the corresponding receivers sense a change in SINR or RSP. This implies that the information of SINR and RSP obtained by the transmitter may correspond to the previous transmission power allocation, rather than the current transmission power allocation. In this case, some links have to update their transmission power according to the out-of-date SINR and RSP information, which may

affect the convergence and performance of SEER. In the following, we would observe the effect of message delay.

- **Convergence**

In this thesis, we observe the effect of message delay on the convergence of SEER when the setting of wireless network is as follows:

1. Each link i picks a sequence of time epochs $\{t_{i1}, t_{i1} + l, t_{i1} + 2l, \cdots\}$, at which its transmission power is updated;
2. For all links, all first time epochs of updating transmission power satisfy $t_{11} < t_{21} < \cdots < t_{M1} < l$;
3. Message delay d_i satisfies $0 < d_i < l$ for each link i.

For such a wireless network, the following lemma can be easily obtained.

Lemma 5.1. *If $|t_{i-11} - t_{i1}| \geq d_i$ for each link $i > 1$ and $|t_{M1} - t_{11} - l| \geq d_1$, then the message delay has no effect on the convergence and the performance of SEER.*

Lemma 5.1 is followed due to the fact that each link always updates its transmission power according to the information of SINR and RSP reflecting the current transmission power, when the message delay satisfies $|t_{i-11} - t_{i1}| \geq d_i$ for each link $i > 1$ and $|t_{M1} - t_{11} - l| \geq d_1$.

On the other hand, when the message delay of some links j happens to be $|t_{j-11} - t_{j1}| < d_j$ in the case of $j > 1$ or $|t_{M1} - t_{11} - l| < d_1$ in the case of $j = 1$, the following simulation would show that the performance of SEER may be degraded, although the convergence is still guaranteed. This is because a such link j updates its transmission power according to the information of SINR and

RSP corresponding to the previous transmission power allocation, rather than the current transmission power allocation.

We consider a ten-link network where the links are randomly placed in a 10m-by-10m area. Let the channel gains G_{ij} to be equal to d_{ij}^{-4}, and the resultant channel gain matrix is

$$G = 10^{-2} \times \begin{bmatrix} 63.3147 & 0.0085 & 0.0128 & 0.0734 & 3.0482 & 0.0052 & 0.0053 & 0.0137 & 0.1181 & 0.0214 \\ 0.0050 & 22.5749 & 9.6020 & 0.1459 & 0.0141 & 2.1076 & 0.0181 & 0.0070 & 0.0046 & 0.0037 \\ 0.0052 & 40.9412 & 11.5358 & 0.1584 & 0.0147 & 1.6196 & 0.0171 & 0.0069 & 0.0046 & 0.0036 \\ 0.0173 & 1.8899 & 22.5381 & 7.7901 & 0.0789 & 0.2865 & 0.0235 & 0.0148 & 0.0133 & 0.0083 \\ 3.7268 & 0.0115 & 0.0188 & 0.1351 & 38.0668 & 0.0075 & 0.0080 & 0.0222 & 0.2113 & 0.0330 \\ 0.0027 & 0.1574 & 0.2721 & 0.0385 & 0.0078 & 40.3408 & 0.0635 & 0.0111 & 0.0043 & 0.0045 \\ 0.0054 & 0.0284 & 0.0712 & 0.0548 & 0.0221 & 0.1944 & 8.2551 & 0.1834 & 0.0216 & 0.0324 \\ 0.0059 & 0.0060 & 0.0115 & 0.0171 & 0.0207 & 0.0156 & 0.7768 & 43.3574 & 0.0769 & 0.4113 \\ 0.0300 & 0.0062 & 0.0115 & 0.0340 & 0.1464 & 0.0089 & 0.0459 & 1.2286 & 9.4236 & 6.4360 \\ 0.0172 & 0.0064 & 0.0124 & 0.0301 & 0.0781 & 0.0111 & 0.0962 & 11.4590 & 1.1122 & 7.5392 \end{bmatrix}.$$

Likewise, let $\{t_{11}, t_{21}, \cdots, t_{M1}\}$ be $\{0.1187, 0.3291, 0.3392, 0.3427, 0.4186, 0.4865, 0.5186, 0.6933, 0.8565, 0.9716\}$, l be 1 millisecond, and K=600. Other system parameters are the same as that in the first simulation of this chapter.

For this realization, we run Continuous-SEER with different message delay, and plot the attained total throughput as well as the number of iterations needed for convergence in Fig. 5.6. In Fig. 5.6, the optimal total throughput achieved by MAPEL is also plotted for comparison. According to Lemma 5.1, the parameter setting of this simulation shows that the message delay has no effect on the convergence and the performance of SEER, as long as the delay $[d_1, \cdots, d_M] \succeq [0.1471\ 0.2104\ 0.0101\ 0.0035\ 0.0759\ 0.0679\ 0.0321\ 0.1747\ 0.1632\ 0.1151]$. In Fig. 5.6, the third curve (i.e., $d_i = 0.003$ for each link i) reveals this

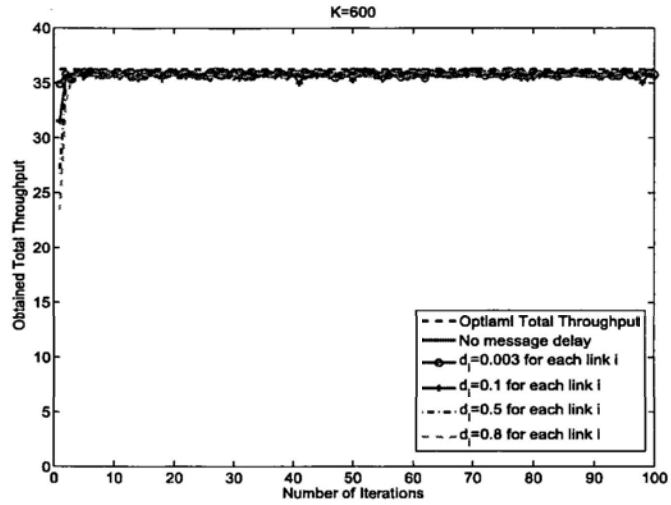


Figure 5.6: The effects of message delay on the convergence and the performance of Continuous-SEER I

conclusion. It can be seen that from Fig. 5.6, the message delay has few effects on the convergence and the performance of SEER for this network topology.

We consider another ten-link network with the channel gain matrix

$$G = 10^{-2} \times \begin{bmatrix} 80.3148 & 0.0983 & 0.0051 & 0.0022 & 0.0038 & 0.0027 & 0.0071 & 0.1430 & 0.0113 & 0.0124 \\ 0.3092 & 11.0005 & 0.0091 & 0.0069 & 0.0038 & 0.0114 & 0.0145 & 0.0726 & 0.0081 & 0.1216 \\ 0.0047 & 0.0052 & 13.7289 & 0.0471 & 0.0769 & 0.0082 & 5.0069 & 0.0173 & 0.0277 & 0.0168 \\ 0.0022 & 0.0055 & 0.0337 & 35.5289 & 0.0048 & 0.0823 & 0.0425 & 0.0041 & 0.0033 & 0.0418 \\ 0.0091 & 0.0038 & 0.3451 & 0.0048 & 23.6047 & 0.0020 & 0.2116 & 0.0692 & 2.4270 & 0.0052 \\ 0.0037 & 0.0192 & 0.0124 & 1.1172 & 0.0026 & 8.7495 & 0.0185 & 0.0048 & 0.0024 & 0.4994 \\ 0.0174 & 0.0199 & 1.4559 & 0.0424 & 0.0483 & 0.0137 & 12.0851 & 0.0862 & 0.0490 & 0.0658 \\ 0.8926 & 0.0271 & 0.0194 & 0.0031 & 0.0189 & 0.0026 & 0.0270 & 51.4990 & 0.1178 & 0.0113 \\ 0.0117 & 0.0037 & 0.1083 & 0.0032 & 3.0174 & 0.0016 & 0.0846 & 0.1029 & 58.7692 & 0.0042 \\ 0.0187 & 0.1430 & 0.0332 & 0.0900 & 0.0059 & 0.1125 & 0.0653 & 0.0261 & 0.0076 & 17.0980 \end{bmatrix}$$

Let $\{t_{11}, t_{21}, \cdots, t_{M1}\}$ be $\{0.0468, 0.1126, 0.1846, 0.1971, 0.3935, 0.4565, 0.6334, 0.7256, 0.7661, 0.8430\}$, l be 1 millisecond, and K=600. Other system parameters are the

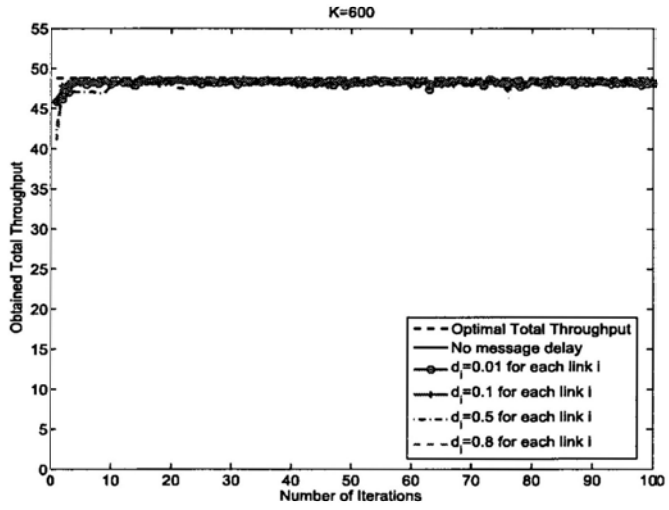


Figure 5.7: The effects of message delay on the convergence and the performance of Continuous-SEER II

same as that in the first simulation of this chapter.

For this realization, we run Continuous-SEER with different message delay, and plot the attained total throughput as well as the number of iterations needed for convergence in Fig. 5.7. In Fig. 5.7, the optimal total throughput achieved by MAPEL is also plotted for comparison. From Fig. 5.7, we find that the results in the third curve verify Lemma 5.1. On the other hand, we further find that the convergence and the performance of SEER is not sensitive to the message delay.

5.4.3 Effect of Message Loss

In this subsection, we observe the effect of message loss on the performance of SEER. Due to the signal strength attenuation and the mutual interference between wireless signals, the transmitter perhaps cannot decode the information of SINR and RSP involved in the control packet, which is regarded as message loss. In this case, the transmitter has to update its transmission power accord-

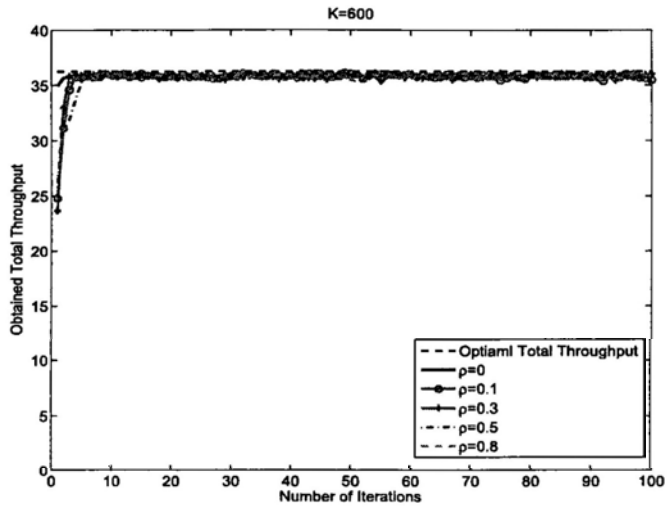


Figure 5.8: The effects of message loss on the convergence and the performance of Continuous-SEER I

ing to the out-of-date SINR and RSP information, which may have effects on the convergence and the performance of SEER. Denote by ρ the probability of message loss. In what follows, we observe the relationship between ρ and the convergence and the performance of SEER.

Based on the above two network topologies, we run Continuous-SEER with different ρ, and plot the attained total throughput as well as the number of iterations needed for convergence in Figs. 5.8 and 5.9. From these two figures, we find that Continuous-SEER can always converge to the global optimality even though there exists the message loss. However, the message loss may affect the convergence speed of Continuous-SEER.

5.5 Summary

We have presented a distributed power control algorithm, SEER, which efficiently converges to the global optimal solutions despite the non-convexity of

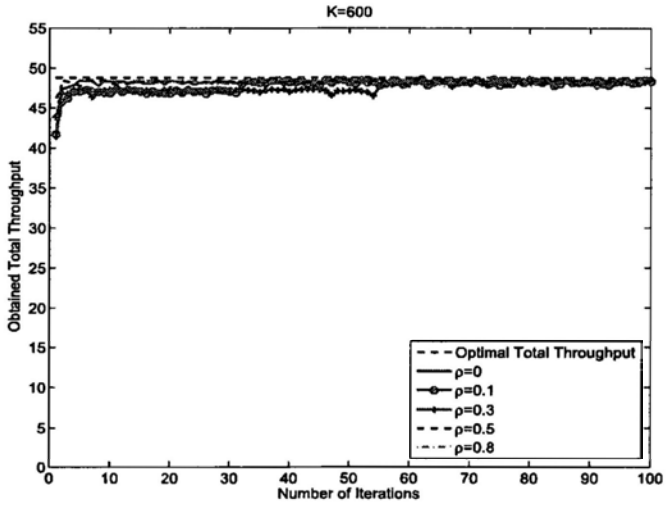


Figure 5.9: The effects of message loss on the convergence and the performance of Continuous-SEER II

the power control problems. Built upon Gibbs Sampling, the key idea of SEER is that each link iteratively and asynchronously update its transmission power by opportunistically exploring the entire surface of the system utility function. Our results show that SEER converges much faster to the global optimal solution than the centralized algorithm MAPEL. This implies that the algorithm is a preferable choice even in systems with a centralized controller. Besides, we find that SEER is robust against message delay and losses.

Chapter 6

Conclusions and Future Work

The final chapter summarizes the major contributions of this thesis and highlights some extensions to this work.

6.1 Summary of Thesis Contributions

This thesis investigates the pure power control issue as well as the joint power control and scheduling issue in interference-limited wireless networks. The main contributions of this thesis are summarized as follows:

In Chapter 2, a monotonic optimization framework for the system utility maximization problem is established, which allows us to bypass the non-convexity issue of the problem and exploit monotonicity to devise an efficient algorithm, MAPEL. Being a global optimal algorithm, MAPEL provides an important benchmark for performance evaluation of existing and newly proposed power control heuristics in this area. For example, by comparing with MAPEL through extensive simulations, we can gain deeper understanding of some existing state-of-the-art centralized and distributed power control algorithms for the same

target.

In Chapter 3, the work is extended to maximize the total system utility in multi-carrier wireless networks. Based on the monotonic optimization framework, we present M-MAPEL to achieve the global optimal solution to the SUM problem in multi-carrier wireless networks. As a global optimal algorithm, M-MAPEL can be used as a benchmark to evaluate the performance of exiting power control algorithms for the same target.

Due to the fact that power control alone is not sufficient to eliminate strong levels of interference between close-by links in dense networks, we investigate maximizing a system utility through joint power control and scheduling in Chapter 4. We first reformulate the non-convex joint power control and scheduling problem into a MO problem, and then propose the S-MAPEL algorithm to efficiently obtain the global optimal joint power control and schedule strategy. By exploiting the inherent symmetry of the optimal solutions, an accelerated algorithm, A-S-MAPEL, is also proposed. Guaranteed to converge to the global optimal solution, S-MAPEL provides an important benchmark for performance evaluation of other joint power control and scheduling heuristics in this area. Using this benchmark, we find that the performance gap between on-off scheduling and joint power control and scheduling is negligible. Besides, we show that the primal/dual decomposition fails to achieve the global optimal solution to the joint power control and scheduling problem.

Finally, in Chapter 5, we present a distributed power control algorithm, SEER, which efficiently and optimally maximizes a system utility despite the non-convexity of the power control problem. Moreover, in SEER, each link distributively and asynchronously updates its transmission power with limited message passing among links. Our results show that SEER converges much faster to the global optimal solution than the centralized algorithm MAPEL. Besides,

we find that SEER is robust against message delay and losses.

In summary, this thesis is a deep and systematic study on power control and scheduling in interference-limited wireless networks. Four key issues of particular interest are investigated: (i) the establishment of a monotonic optimization framework for non-convex power control problems; (ii) the generalization of the proposed framework to joint scheduling and power control in dense networks; (iii) the development of low-complexity algorithms by exploiting the special features of the formulated problems; (iv) the development of distributed power control based on the concept of Gibbs Sampling. To the best of our knowledge, this is a first attempt at exploiting the hidden monotonicity to solve a nonconvex engineering problem efficiently. We believe the new framework opens up many interesting avenues for researchers to devise global optimization algorithms for problems that have been open for years, as most optimization problems in communications and networking are essentially monotonic, although not convex.

6.2 Future Work

We propose the following extensions to the work presented in this thesis.

The results in Chapter 3, 4, and 5 are largely based on the recent advances in monotonic optimization. The idea behind these results help us to pray the way for studying non-convex resource allocation problems in communications and networking. We believe that monotonicity is another important property besides convexity that can be exploited to efficiently solve an optimization problem. By exploiting the hidden monotonicity of optimization problems, we can bypass the non-convexity issue and obtain the global optimal solutions efficiently.

Joint scheduling and power control problem is by nature NP-hard. Hence, it requires exponential computational complexity with any solution method, as long as the global optimal solution is to be obtained. Correspondingly, as a global optimal algorithm, S-MAPEL, has a considerably high computational complexity. In our future work, variants of the algorithms will be developed to expedite the convergence and reduce the computational complexity.

In our current work, we only consider the case of single-hop wireless networks. It would be an interesting future research to extend the joint power control and scheduling work to a multihop network. As such, we can first formulate the flow control problem into a joint power control and scheduling optimization problem, and then solve the corresponding problem by exploiting its hidden monotonicity.

Besides the centralized resource allocation discussed in this thesis, there are also interesting problems on distributed resource allocation in wireless networks. With the proliferation of wireless infrastructureless networks such as ad hoc and sensor networks, it is increasingly crucial to devise a scheme that implements resource allocation in a distributed fashion. SEER is an attempt that each link distributively and asynchronously updates its transmission power with limited message passing among links to maximize the overall system utility. In our future work, we would endeavor to reduce message passing among links under the constraint of meeting the performance requirements. Besides, we would analyze the effect of message loss and message delay on the performance of power control.

In this thesis, we assume that the channels are slow fading. In our future work, we can extend the current work to wireless networks with time-varying fading channels. In such a case, the stochastic resource allocation has to be taken into consideration.

References

[1] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power Control Wireless Cellular Networks," in *Foundations and Trends in Networking*, vol. 2, no. 4, pp. 381-533, July 2008.

[2] J. Zander, "Performance of optimum transmitter power control in cellular radio systems," in *IEEE Trans. Veh. Technol.*, vol. 41, no. 1, pp. 57-62, Feb. 1992.

[3] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," in *IEEE Trans. Veh. Technol.*, vol. 42, no. 6, pp. 641-646, Nov. 1993.

[4] S. A. Grandhi and J. Zander, "Constrained Power Control in Cellular Radio Systems," *Proc. IEEE VTC*, Stockholm, Sweden, June 1994.

[5] S. A. Grandhi, R. Vijayan, D. J. Goodman, and J. Zander, "Centralized Power Control in Cellular Radio Systems," *IEEE Trans. Veh. Technol.*, vol. 42, no. 6, pp. 466-468, Nov. 1993.

[6] R. D. Yates, "A Framework for Uplink Power Control in Cellular Radio Systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341-1347, Sep. 1995.

[7] N. Bambos, S. C. Chen, and G. J. Pottie, "Channel access algorithms with ac-

tive link protection for wireless communication networks with power control," in *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 583-597, Oct. 2000.

[8] C. W. Sung, "Log-Convexity Property of the Feasible SIR Region in Power-Controlled Cellular Systems," *IEEE Commun.Letters*, vol. 6, no. 6, pp 248-249, June 2002.

[9] M. Schubert and H. Boche, "Solution of the Multi-user Downlink Beamforming Problem with Individual Sir Constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18-28, Jan. 2004.

[10] H. Boche and S. Stanczak,"Convexity of some feasible QoS regions and asymptotic behavior of the minimum total power in CDMA systems," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2190 - 2197, Dec. 2004.

[11] D. Julian, M. Chiang, D. O. Neill, and S. Boyd, "Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," *Proc. IEEE INFOCOM*, vol. 2, pp. 477-486, June 2002.

[12] M. Chiang, C. W. Tan, D. Palomar, D. O' Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 1, no. 7, pp. 2640-2651, July 2007.

[13] J. Huang, R. A. Berry, and M. L. Honig, "Distributed Interference Compensation for Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074 - 1084, May 2006.

[14] P. Hande, S. Rangan, and M. Chiang, "Distributed uplink power control for optimal sir assignment in cellular data networks," *Proc. IEEE INFOCOM*, vol. 4, pp. 1929-1941, Apr. 2006.

[15] L. P. Qian, Y. J. Zhang, and J. W. Huang, "MAPEL: Achieving Global Optimality for a Non-convex Wireless Power Control Problem," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553-1563, Mar. 2009.

[16] L. P. Qian and Y. J. Zhang, "Monotonic Optimization for Non-concave Power Control in Multiuser Multicarrier Network Systems," *Proc. IEEE INFOCOM*, pp. 172-180, Apr. 2009.

[17] L. P. Qian and Y. J. Zhang, "S-MAPEL: Monotonic Optimization for Non-convex Joint Power Control and Scheduling Problems," to appear in *IEEE Trans. Wireless Commun.*.

[18] H. Tuy, "Monotonic Optimization: Problems and Solution Approaches," *SIAM. J. OPTIM.*, no. 2, vol. 11, pp. 464-494, 2000.

[19] T. Elbatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad Hoc Networks" ," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 74-85, Jan. 2004.

[20] A. K. Das, R. J. Marks, P. Arabshahi, and A. Gray, "Power Controlled Minimum Frame Length Scheduling in TDMA Wireless Networks with Sectored Antennas," *Proc. IEEE INFOCOM*, Mar. 2005.

[21] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for dynamic multi-server wireless systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1506-1515, June 2006.

[22] R. L. Cruz and A. V. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," *Proc. IEEE INFOCOM*, vol.1, pp. 702-711, Mar. 2003.

[23] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291-303, February 2002.

[24] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, pp. 210-221, March 2003.

[25] T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," *Wireless Networks*, vol. 8, pp. 659-670, 2002.

[50] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans Pattern Anal Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.

[51] G. Winkler, "Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A mathematical Introduction," 2nd ed., Springer Verlag, 2003.

[28] D. Catrein, L. A. Imhof, and R. Mathar, "Power control, capacity, and duality of uplink and downlink in cellular CDMA systems," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1777-1785, Oct. 2004.

[29] C.Y. Wong, R.S. Cheng, K.B. Letaief, and R.D. Murch, "Multiuser OFDM with Adaptive Subcarrier, Bit, and Power Allocation," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1747-1758, Oct. 1999.

[30] S. Pietrzyk and G. Janssen, "Multiuser Subcarrier Allocation for QoS Provision in the OFDMA Systems," *Proc. IEEE VTC*, vol. 2, pp. 1077-1081, Sept. 2002.

[31] D. Kivanc, G. Li, and H Liu, "Computationally Efficient Bandwidth Allocation and Power Control for OFDMA," *IEEE Trans. Wireless Commun.*, vol. 2, no. 6, pp. 1150-1158, Nov. 2003.

[32] G. Munz, S. Pfletschinger, and J. Speidel, "An efficient waterfilling algorithm for multiple access OFDM," *Proc. IEEE GLOBECOM*, vol. 1, no. 5, pp. 681-685, Nov. 2002.

[33] K. Kim, Y. Han, and S. Kim, "Joint Subcarrier and Power Allocation in Uplink OFDMA Systems," *IEEE Commun. Let.*, vol. 9, no. 6, pp. 526-528, June 2005.

[34] I. C. Wong, Z. Shen, B. L. Evans, and J. G. Andrews, "A Low Complexity Algorithm for Proportional Resource Allocation in OFDMA Systems," *Proc. SIPS* 2004.

[35] L. Gao and S. Cui, "Efficient Subcarrier, Power, and Rate Allocation with Fairness Consideration for OFDMA Uplink," *IaEEE Trans. Wireless Commun.*, vol. 7, no. 5, May 2008.

[36] C. Y. Ng and C. W. Sung, "Low Complexity Subcarrier and Power Allocation for Utility Maximization in Uplink OFDMA Systems," *IEEE Trans. Wireless Commun.*, Vol. 7, No. 5, May 2008.

[37] J. Jang and K. B. Lee, "Transmit Power Adaptation for Multiuser OFDM Systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 171-178, Feb. 2003.

[38] H. Li and H. Liu, "An analysis of uplink OFMDA optimality," IEEE Trans. Wireless Commun., vol. 6, no. 8, pp. 2972-2983, Aug. 2007.

[39] F. Meshkati, M. Chiang, H. V. Poor, and S. C. Schwartz, "A Game-Theoretic Approach to Energy-Efficient Power Control in Multi-Carrier CDMA Systems," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 6, pp. 1115 - 1129, June 2006.

[40] S. T. Chung and J. M. Cioffi, "Rate and Power Control in a Two-User Multicarrier Channel with No Coordination: The Optimal Scheme vs. Suboptimal Methods," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1768 - 1772, Nov. 2003.

[41] A. Rashdi, N. M. Sheikh, and A.U.H. Sheikh, "Distributed and Centralized Power Control Algorithms for Very High-Speed Digital Subscriber Lines (VDSL) Upstream Transmission," *Proc. EUSIPCO*, 2005.

[42] J. Papandriopoulos and J. S. Evans, "Low-Complexity Distributed Algorithms for Spectrum Balancing in Multi-User DSL Networks," *Proc. IEEE ICC*, vol.7, pp. 3270-3275, June 2006.

[43] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum management for digital subscriber lines," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 922-933, May 2006.

[44] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 5, pp. 1105-1115, June 2002.

[45] N. T. H. Phuong and H. Tuy, "A Unified Monotonic Approach to Generalized Linear Fractional Programming," Journal of Global Optimization, Kluwer Academic Publishers, pp. 229-259, 2003.

[46] J. B. G. Frenk and S. Schaible, "Fractional Programming," Handbook of Generalized Convexity and Generalized Monotonicity, pp. 335-386, 2006.

[47] S. Boyd and L. Vandenberghe, "*Convex Optimization*," Cambridge, U.K.: Cambridge Univ. Press, 2004.

[48] D. P. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439-1451, Aug. 2006.

[49] X. Lin and N. B. Shroff, "The Impact of of Imperfect Scheduling on Cross-layer Rate Control in Wireless Networks," *IEEE/ACM Trans. Networking* , vol. 14, no. 2, pp. 302-315, Apr. 2006.

[50] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans Pattern Anal Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.

[51] G. Winkler, "Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A mathematical Introduction," 2nd ed., Springer Verlag, 2003.

[52] H. R. Varian, "Intermediate Microeconomics: A Modern Approach,", *Chapeter 4*, Norton, New York, 1999.

[53] P. Salamon, P. Sibani, and R. Frost, "Facts, Conjectures, and Improvements for Simulated Annealing," SIAM Monographs, 2002.