

# **UCLA**

## **UCLA Electronic Theses and Dissertations**

### **Title**

Optimization Methods for Resource Allocation and Machine Learning Applications

### **Permalink**

<https://escholarship.org/uc/item/7wt8n5wk>

### **Author**

Ahuja, Kartik

### **Publication Date**

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Optimization Methods for Resource Allocation and Machine Learning Applications

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Kartik Ahuja

2019

© Copyright by

Kartik Ahuja

2019

## ABSTRACT OF THE DISSERTATION

Optimization Methods for Resource Allocation and Machine Learning Applications

by

Kartik Ahuja

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Los Angeles, 2019

Professor Gregory J. Pottie, Chair

In many engineering and machine learning applications, we often encounter optimization problems (e.g., resource allocation, clustering) for which finding the exact solution is computationally intractable. In such problems, ad-hoc approximate solutions are often used, which have no performance guarantees. Our goal is to develop approximate optimization methods with the following features a) provable performance guarantees, and b) computational tractability. In this dissertation, we focus on several challenging problems in resource allocation and machine learning and develop optimization methods for the same.

In the first part of this dissertation, we develop optimization methods to solve fundamental resource allocation problems encountered in the design of different systems, namely wireless networks, crowdsourcing systems, and healthcare systems.

Dense deployment of heterogeneous small cells (e.g., picocells, femtocells) is becoming the most effective way to combat the exploding demand for the wireless spectrum. Given the large-scale nature of these deployments, developing resource sharing policies using a centralized system can be computationally and communicationally prohibitive. To this end, we propose a general framework for distributed multi-agent resource sharing. We show that the proposed framework significantly outperforms the state-of-the-art. We prove quite general constant factor approximation guarantees with respect to the optimal solutions.

Matching platforms for freelancing (e.g., Upwork) are becoming mainstream. These platforms are faced with the challenging task of allocating workers to clients in order to

generate maximum revenues, taking into consideration that both sides are self-interested, have limited information about the other, and desire to be matched with the best possible partners. We propose a dynamic matching mechanism that takes these challenges into account and achieves many of the aforesaid properties.

Screening plans are used for the early detection of several diseases, such as breast cancer and colon cancer. These screening plans are not personalized to the history and demographics of the subject and can often lead to a delay in the detection of the disease and in other cases cause unnecessary invasive tests such as biopsies. We show that constructing exactly optimal personalized screening plans that minimize the number of screens given a tolerance on the delay is computationally intractable. We develop a framework to solve the proposed problem approximately. We establish general performance guarantees and show that the proposed solution is computationally tractable. We apply the framework to breast cancer screening and establish its utility in comparison to the existing clinical guidelines.

In the second part of this dissertation, we develop optimization methods useful for machine learning applications. Machine learning models are increasingly becoming a part of many of the decision making systems, for instance, clinical decision support systems. Many of the machine learning models are hard to interpret and thus are often called “black-box” models. We propose a method that approximates the black-box models using piecewise-linear approximations. This approach helps explain the model using linear models in different regions of the feature space. We provide provable fidelity, i.e., how well does approximation reflect the black-box, guarantees and show that the method is computationally tractable. We carry out experiments on different datasets and establish the utility of our approach.

Kullback-Leibler divergence is a fundamental quantity used in many disciplines, such as machine learning, statistics, and information theory. We develop an optimization-based approach to estimate the Kullback-Leibler divergence, which relies on the Donsker-Varadhan representation. The state-of-the-art estimator based on this representation relies on solving a non-convex optimization problem and hence, is not consistent. We propose a convex reformulation to construct an estimator, which we show is consistent. We also carry out experiments to show that the proposed estimator is better than the competing estimator.

The dissertation of Kartik Ahuja is approved.

Paulo Tabuada

Lieven Vandenberghe

Yingnian Wu

Gregory J. Pottie, Committee Chair

University of California, Los Angeles

2019

*To mom, dad, Shiney, and Prateek*

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation	1
1.2	Roadmap	2
1.2.1	Optimization Methods for Resource Allocation	3
1.2.2	Optimization Methods for Machine Learning Applications	5
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Background for Resource Allocation in Multi-Agent Systems	6
2.1.1	Standard Formulation of Optimization Problem	6
2.1.2	Convex Sets	7
2.1.3	Convex Functions	7
2.1.4	Standard Formulation of Convex Optimization Problem	7
2.1.5	Local Minima	7
2.1.6	Global Minima	8
2.1.7	Duality	8
2.1.8	The Dual Problem	9
2.1.9	Centralized vs Distributed Optimization	10
2.1.10	Dual Decomposition	10
2.1.11	Augmented Lagrangian	11
2.1.12	Alternating Direction Method of Multipliers	12
2.1.13	Graph Theory Basics and Algorithms	12
2.2	Background for Dynamic Matching with Strategic Agents	15
2.2.1	Non-Cooperative Game Theory	15

2.3	Background for Dynamic Resource Allocation Planning . . . . .	16
2.3.1	Fully Observed Markov Decision Processes . . . . .	16
2.3.2	Partially Observable Markov Decision Process . . . . .	18
2.4	Conclusion . . . . .	21
<b>3</b>	<b>Centralized Large Scale Multi-Agent Resource Sharing . . . . .</b>	<b>22</b>
3.1	Introduction . . . . .	22
3.2	Related Works . . . . .	26
3.2.1	Interference Management Policies Based on Constant Power Control .	26
3.2.2	Interference Management Based on Spatial Time/Frequency Reuse .	27
3.2.3	Other Interference Management Policies . . . . .	28
3.3	System Model . . . . .	29
3.3.1	Heterogeneous Network of Macrocells and Femtocells . . . . .	29
3.3.2	Interference Management Policies . . . . .	31
3.4	Problem Formulation . . . . .	32
3.4.1	Policy Design Problem . . . . .	32
3.4.2	Motivating Example . . . . .	33
3.5	Design Framework . . . . .	36
3.5.1	Description of the Proposed Design Framework . . . . .	36
3.5.2	Constructing Optimal Interference Graphs . . . . .	40
3.5.3	Optimality of the Proposed Design Framework . . . . .	40
3.5.4	Complexity for Computing the Policy . . . . .	43
3.5.5	Impact of the Density of Femtocells and Macrocells . . . . .	44
3.6	Efficient Interference Management for Large-Scale Networks . . . . .	45
3.6.1	Efficient Computation of a Subset of MISs . . . . .	45

3.6.2	Performance Guarantees for Large Networks . . . . .	46
3.6.3	Complexity for Computing the Subset of MISs . . . . .	49
3.6.4	Extensions . . . . .	49
3.7	Illustrative Results . . . . .	51
3.7.1	Performance Gains Under Varying Interference Levels . . . . .	52
3.7.2	Performance Scaling in Large Networks . . . . .	53
3.7.3	Performance under Dynamic Channel Conditions . . . . .	55
3.8	Discussion on the Generality of the Framework . . . . .	56
3.9	Conclusion . . . . .	57
3.10	Appendix . . . . .	58
3.10.1	Appendix A . . . . .	58
3.10.2	Appendix B . . . . .	60
3.10.3	Appendix C . . . . .	62
3.10.4	Appendix D . . . . .	65
<b>4</b>	<b>Distributed Large Scale Multi-Agent Resource Sharing . . . . .</b>	<b>69</b>
4.1	Introduction . . . . .	69
4.2	Related Works . . . . .	72
4.2.1	Distributed Interference Management Based on Power Control . . . .	73
4.2.2	Distributed Spatial Reuse Based on Maximal Independent Sets . . . .	73
4.2.3	Distributed Power Control and Spatial Reuse For Multi-Cell Networks	75
4.3	System Model . . . . .	75
4.3.1	Heterogeneous Network of Small Cells . . . . .	75
4.3.2	Interference Management Policies . . . . .	76
4.4	Problem Formulation . . . . .	77

4.4.1	The Interference Management Policy Design Problem . . . . .	77
4.5	Design Framework for Distributed Interference Management . . . . .	78
4.5.1	Proposed Design Framework . . . . .	78
4.5.2	A Motivation Example . . . . .	86
4.5.3	Performance Guarantees for Large Networks and Properties of Interference Graphs . . . . .	87
4.5.4	Self-Adjusting Mechanism for Dynamic Entry/Exit of UEs . . . . .	90
4.5.5	Extensions . . . . .	91
4.6	Illustrative Results . . . . .	91
4.6.1	Performance under Time-Varying Channel Conditions . . . . .	92
4.6.2	Performance Scaling in Large Networks . . . . .	94
4.6.3	Self-adjusting Mechanism for Dynamic Entry/Exit of the UEs . . . . .	95
4.7	Discussion on the Generality of the Framework . . . . .	96
4.8	Conclusion . . . . .	97
4.9	Appendix . . . . .	98
4.9.1	Appendix A . . . . .	98
4.9.2	Appendix B . . . . .	103
4.9.3	Appendix C . . . . .	104
4.9.4	Appendix D . . . . .	106
4.9.5	Appendix E . . . . .	108
4.9.6	Appendix F . . . . .	108
4.9.7	Appendix G . . . . .	109
<b>5</b>	<b>Dynamic Matching with Strategic Agents . . . . .</b>	<b>111</b>
5.1	Introduction . . . . .	111

5.2	Dynamic Matching Mechanism Design . . . . .	114
5.2.1	Model and Problem Formulation . . . . .	114
5.2.2	Proposed Mechanism and its Properties . . . . .	122
5.3	Numerical Experiments . . . . .	128
5.4	Extensions . . . . .	130
5.5	Conclusion . . . . .	131
5.6	Appendix . . . . .	131
5.6.1	Appendix A . . . . .	132
5.6.2	Appendix B . . . . .	132
5.6.3	Appendix C . . . . .	136
5.6.4	Appendix D . . . . .	141
5.6.5	Appendix E . . . . .	146
5.6.6	Appendix F . . . . .	149
5.6.7	Appendix G . . . . .	150
5.6.8	Appendix H . . . . .	151
5.6.9	Appendix I . . . . .	152
5.6.10	Appendix J . . . . .	153
<b>6</b>	<b>Dynamic Resource Allocation Planning . . . . .</b>	<b>157</b>
6.1	Introduction . . . . .	157
6.2	Model and Problem Formulation . . . . .	159
6.3	Proposed Approach . . . . .	163
6.3.1	Constructing the Exactly Optimal Policy . . . . .	165
6.3.2	Constructing the Approximately Optimal Policy . . . . .	165
6.3.3	Robustness . . . . .	167

6.4	Illustrative Experiments . . . . .	169
6.5	Related Works . . . . .	174
6.6	Discussion on Generality of the Proposed Framework . . . . .	175
6.7	Conclusion . . . . .	175
6.8	Appendix . . . . .	176
6.8.1	Appendix A . . . . .	176
6.8.2	Appendix B . . . . .	180
6.8.3	Appendix C . . . . .	181
6.8.4	Appendix D . . . . .	182
6.8.5	Appendix E . . . . .	186
6.8.6	Appendix F . . . . .	189
<b>7</b>	<b>Optimal Piecewise Local-Linear Approximations . . . . .</b>	<b>191</b>
7.1	Introduction . . . . .	191
7.1.1	Contributions . . . . .	192
7.2	Related Works . . . . .	193
7.2.1	Local Frameworks . . . . .	194
7.2.2	Global Frameworks . . . . .	195
7.2.3	Relationship of Piecewise Linear and Piecewise Constant Models to Existing Works . . . . .	196
7.3	Problem Formulation . . . . .	196
7.3.1	Toy Example . . . . .	196
7.3.2	Interpretive models . . . . .	197
7.3.3	Loss functions . . . . .	201
7.3.4	Risk Minimization . . . . .	201

7.3.5	Empirical Risk Minimization . . . . .	201
7.4	Piecewise Local-Linear Interpreter . . . . .	202
7.5	Main Results . . . . .	203
7.5.1	PAC learnability of Algorithms 1 and 2: . . . . .	203
7.6	Experiments . . . . .	206
7.6.1	Metrics . . . . .	206
7.6.2	Synthetic Dataset . . . . .	206
7.6.3	Interpret RF regression on Boston Housing Dataset . . . . .	208
7.6.4	Large dataset . . . . .	214
7.7	Connection with K-means clustering . . . . .	215
7.7.1	Ordered Partitions . . . . .	216
7.8	Conclusion . . . . .	218
7.9	Appendix . . . . .	218
7.9.1	Appendix A . . . . .	218
7.9.2	Appendix C . . . . .	221
7.9.3	Appendix D . . . . .	225
7.9.4	Appendix E . . . . .	226
7.9.5	Appendix F . . . . .	230
7.9.6	Appendix G . . . . .	235
<b>8</b>	<b>Optimization Based Approach to Estimating Kullback-Leibler Divergence</b>	<b>236</b>
8.1	Introduction . . . . .	236
8.2	Problem Formulation and Approach . . . . .	237
8.2.1	The Donsker-Varadhan Representation . . . . .	238
8.2.2	MINE . . . . .	238

8.2.3	KKLE: Kernel Based KL Divergence Estimation . . . . .	239
8.2.4	Analyzing the Consistency of KKLE . . . . .	244
8.2.5	Convergence and Complexity . . . . .	246
8.3	Experiments . . . . .	247
8.3.1	Comparisons . . . . .	247
8.3.2	Explaining KKLE’s performance . . . . .	249
8.3.3	Application to Metrics for Fairness . . . . .	250
8.4	Conclusion . . . . .	251
<b>9</b>	<b>Conclusions and Future Work . . . . .</b>	<b>252</b>
<b>References</b>	. . . . .	<b>254</b>

## LIST OF FIGURES

3.1	Coloring based scheduling in a) schedules less than two UEs per time slot on an average, while MIS based scheduling in b) is more efficient and schedules two UEs per time slot. . . . .	27
3.2	A) System model for the three-cell network, B) The interference graph for the three-cell network. . . . .	30
3.3	Steps in the design framework. . . . .	36
3.4	An example to illustrate the optimality of proposed framework. . . . .	43
3.5	Different interference graphs for the $3 \times 3$ BS grid. . . . .	51
3.6	Optimal interference graph selection for $3 \times 3$ grid. . . . .	52
3.7	Performance comparison of the proposed policy for different size of the grid. . .	52
3.8	Illustration of the setup with 3 rooms. . . . .	53
3.9	Comparing the proposed policy against others for a) sum throughput as the metric, b) minimum throughput as the metric. . . . .	55
3.10	Illustrating the robustness of interference graph selection based on path loss. . .	55
4.1	Illustration of a heterogeneous small cell network. . . . .	75
4.2	Steps in the design framework. . . . .	78
4.3	Illustration of the distributed generation of MISs in Step 2. . . . .	79
4.4	A heterogeneous network of 2 PBS and 2 FBS and their corresponding UEs. . .	85
4.5	Comparison of the proposed policy with state of the art under different interference strength and time-varying channel conditions. . . . .	93
4.6	a) Comparison of max-min fairness under different grid sizes, b) Sample paths of sum throughput under dynamic entry/exit of UEs in the network. . . . .	94
4.7	a) Different interference graphs for the $3 \times 3$ BS grid, b) Illustration of setup with 3 rooms. . . . .	94

4.8	Comparison of max-min fairness and average throughput per UE against state of the art for large networks. . . . .	95
4.9	Problems used to transit from the Coupled Problem (CP) to Decoupled Problem (DP). . . . .	108
5.1	Comparison of the proposed mechanism with other approaches. . . . .	130
6.1	Illustration of dynamic personalization. . . . .	168
6.2	Impact of the type of disease. . . . .	172
6.3	Graphical model for the screening setting. . . . .	181
7.1	Comparison of the black-box model versus the piecewise approximation. . . . .	197
7.2	Comparison of the black-box model versus the piecewise approximation. . . . .	198
7.3	Example of a 2-D partition. . . . .	200
7.4	Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 4, W = 1$ ) EQ, b) ( $H = 2, W = 2$ ) EQ, c) ( $H = 2, W = 2$ ) OP. . . . .	208
7.5	Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 4, W = 1$ ) OP, b) ( $H = 1, W = 4$ ) OP. . . . .	208
7.6	Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 2, W = 2$ ) EQ, b) ( $H = 2, W = 2$ ) OP, c) ( $H = 4, W = 1$ ) EQ. . . . .	213
7.7	Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 4, W = 1$ ) OP, b= ( $H = 1, W = 4$ ) OP. . . . .	213
7.8	Data points selected based on PLLI and corresponding explanations (for the top five features.) . . . . .	214



## LIST OF TABLES

2.1 Utility matrix for prisoner's dilemma. . . . .	16
3.1 Comparisons against spatial reuse TDMA based policies. . . . .	35
3.2 The algorithm run by each UE $i$ . . . . .	41
3.3 Algorithm run by the designer. . . . .	47
4.1 Comparisons in terms of max-min fairness & sum throughput criterion. . . . .	88
4.2 Generating MISs in a distributed manner, algorithm for UE $i$ . . . . .	99
4.3 Phase 2 of the distributed MIS generation. . . . .	100
4.4 ADMM update algorithm for UE $i$ . . . . .	101
5.1 Comparison of the different works. AS: Adverse Selection, MH: Moral Hazard .	155
5.2 Summary of the knowledge structure. . . . .	156
6.1 Comparison of the proposed policy with annual screening for both high and low risk group. . . . .	173
6.2 Comparison of the proposed policy with biennial policy. . . . .	173
6.3 Comparison of the proposed policies across different risk groups. . . . .	174
7.1 Comparison of RF Regressor with other methods. . . . .	208
7.2 Comparison of RF Regressor with other methods. . . . .	209
7.3 Model Summary of RF Regressor: Synthetic Data. . . . .	209
7.4 Comparison of RF Regressor with other methods. . . . .	210
7.5 Comparison of PI interpreter different hyperparameter configurations. . . . .	211
7.6 Comparison of RF Regressor with other methods. . . . .	212
7.7 Comparison of RF Regressor with other methods. . . . .	212

7.8	Comparison of coverage of various selection methods.	214
7.9	Comparison of RF Regressor with other methods.	216
7.10	Comparison of PLLI interpreter different hyperparameter configurations.	216
8.1	KKLE vs MINE estimator for large data.	248
8.2	KKLE vs MINE estimator for small data.	249

## ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Prof. Gregory J. Pottie, without whose support and guidance this dissertation would not exist. I have learned a lot from Prof. Gregory J. Pottie. His work ethic and motivation to do high quality research will always be a source of inspiration.

I am grateful to all the other committee members, Prof. Paulo Tabuada, Prof. Lieven Vandenberghe, and Prof. Yingnian Wu for evaluating my research and guiding me. I am thankful to Prof. Mihaela van der Schaar, Prof. William Zame, Prof. Yuanzhang Xiao and all my collaborators for their support in all our research projects.

I would also like to thank the following organizations for funding my research, UCLA Graduate Division for Dissertation Year Fellowship, Guru Krupa Fellowship Foundation, Office of Naval Research (ONR) and National Science Foundation (NSF).

I want to thank my friends Abhimanyu, Venkata, Mai, Onur, Tina, Lu, who were very supportive of me during my stay at UCLA. My family back in India was always there for me during all the ups and downs. I don't have words to express gratitude towards my mom, dad, Shiney and Prateek who I missed a lot throughout my PhD. Without their support none of this would be possible.

## VITA

2013 B.Tech. (Electrical Engineering), Indian Institute of Technology, Kanpur

2013 M.Tech. (Electrical Engineering), Indian Institute of Technology, Kanpur

2013–present Graduate Student Researcher, Electrical and Computer Engineering Department, University of California, Los Angeles.

## PUBLICATIONS

1. **Kartik Ahuja**, Yuanzhang Xiao, and Mihaela van der Schaar, “Efficient interference management policies for femtocell networks.” *IEEE Transactions on Wireless Communications* **14**(9) (2015): 4879-4893.
2. **Kartik Ahuja**, Yuanzhang Xiao, and Mihaela van der Schaar, “Distributed interference management policies for heterogeneous small cell networks.” *IEEE Journal on Selected Areas in Communications* **33**(6) (2015): 1112-1126.
3. Yuanzhang Xiao, **Kartik Ahuja**, and Mihaela van der Schaar, “Spectrum sharing for delay-sensitive applications with continuing QoS guarantees.” In *IEEE Global Communications Conference*, (2014): 1265-1270.
4. **Kartik Ahuja**, William Zame, and Mihaela van der Schaar, “Dpscreen: Dynamic personalized screening.” In *Advances in Neural Information Processing Systems*, (2017): 1321-1332.

5. **Kartik Ahuja** and Mihaela van der Schaar, “Dynamic matching and allocation of tasks.” accepted in *ACM Transactions on Economics and Computation* and a short version appeared in *30th International Conference on Game Theory*, (2019).
6. **Kartik Ahuja**, William Zame, and Mihaela van der Schaar, “Optimal piecewise approximations for model interpretations.” In *53rd Annual Asilomar Conference on Signals, Systems, and Computers*, (2019).
7. **Kartik Ahuja**, “Estimating Kullback-Leibler divergence using kernel machines.” In *53rd Annual Asilomar Conference on Signals, Systems, and Computers*, (2019).
8. **Kartik Ahuja**, and Mihaela van der Schaar, “Joint Concordance Index.” In *53rd Annual Asilomar Conference on Signals, Systems, and Computers*, (2019).
9. **Kartik Ahuja**, Simpson Zhang, and Mihaela van der Schaar, “Towards a theory of societal co-evolution: individualism versus collectivism.” In *IEEE Global Conference on Signal and Information Processing*, (2014): 769-773.
10. **Kartik Ahuja**, William Zame, and Mihaela van der Schaar, “Working alone and working with others: implications for the Malthusian era.” Under second round of review at *Economic Theory*.
11. Ahmed M. Alaa, **Kartik Ahuja**, and Mihaela van der Schaar, “A micro-foundation of social capital in evolving social networks.” *IEEE Transactions on Network Science and Engineering* 5(1) (2017): 14-31.
12. Ahmed M. Alaa, **Kartik Ahuja**, and Mihaela van der Schaar, “Self-organizing networks of information gathering cognitive agents.” *IEEE Transactions on Cognitive Communications and Networking* 1(1) (2015): 100-112.

# CHAPTER 1

## Introduction

### 1.1 Motivation

The design and analysis of engineering systems often requires us to solve intractable optimization problems (for instance, non-convex optimization problems, combinatorial optimization problems). We often encounter problems of this nature in engineering applications (e.g., resource allocation), and also in machine learning applications (e.g., clustering). In many of these problems the solutions that are proposed are ad-hoc in nature, i.e., there are no provable performance guarantees. In this dissertation, we develop approximate optimization methods to solve several such optimization problems with two desirable features a) provable performance guarantees, and b) computational tractability. We focus on problems in the areas of resource allocation and machine learning.

Resource allocation is central to many disciplines such as engineering, operations research, and statistics. Examples of resource allocation in engineering are channel/time-slot allocation for interference management, medium access control, etc. Some examples from operations research and statistics are task allocation for crowdsourcing platforms such as Upwork, Amazon Mechanical Turk, and designing screening policies for early detection of rare diseases such as different types of cancers. Each of these problems is very different in nature and present very different challenges. For instance, interference management requires understanding how different devices contending for the resources impact each other. In a crowdsourcing system, task allocation mechanism design requires understanding of how different strategic workers contending for different tasks behave under different mechanisms. Screening policy design for different diseases requires understanding the disease dynamics

for different diseases and the costs associated with screening. In the first part of this dissertation, we develop optimization methods that address the various challenges presented by these resource allocation problems. We show that the proposed methods outperform the state-of-the-art methods. While we apply our methods to the particular applications we described, we show that the proposed methods are very general and can be applied in many other settings.

In the second part of this dissertation, we develop optimization methods for machine learning applications. In recent years, machine learning models are increasingly being used in different decision making systems such as clinical decision support systems, and security systems. Deployment of these systems in real life is faced with several challenges. One main challenge is that of interpretability. These systems are based on machine learning models, which are hard to interpret and are thus referred to as “black-boxes”. The European Union’s Law on Data Regulation that took effect in 2018 [GF16] makes it mandatory for “black-box” models to explain how they arrive at the decisions before implementing them in practice. We propose new methods to better understand these black-box models and interpret their outcomes.

Kullback-Leibler (KL) divergence is a fundamental quantity used in machine learning, statistics, and information theory. At the end, we propose an optimization-based approach to estimate Kullback-Leibler divergence. We establish the utility of the proposed estimator in comparison to the competing estimators.

## 1.2 Roadmap

In Chapter 2, we provide some background and references that would be useful for a reader not familiar with the area. In Chapter 3-6, we focus on optimization methods for resource allocation and in Chapters 7-8, we focus on optimization methods for machine learning applications. In Chapter 9, we conclude this dissertation. In several chapters we provide an Appendix at the end, which consists of the proofs for results (propositions, theorems etc.) in that chapter.

## 1.2.1 Optimization Methods for Resource Allocation

### 1.2.1.1 Resource Allocation in Multi-Agent Systems

In the first part of this dissertation (based on works [AXS15a] [AXS15b], [XAS14]), we study large scale resource allocation in multi-agent systems with *strong negative local externalities* (i.e. strong interference and congestion), where the decisions are made by the agents in a distributed fashion. In Chapter 3, we start with a simpler problem. We relax the constraint that the agents have to act in a distributed manner and instead we let one centralized agent make the decisions. Even with a centralized agent, the optimization problem at hand is intractable to solve exactly. We propose an approximately optimal polynomial time solution that is guaranteed to achieve a constant factor approximation of the optimal value under many scenarios. One of the key ideas that the approach rests on is to abstract the local interference constraints as a graph. We then combine ideas from graph theory and optimization theory to arrive at the proposed optimal solutions. The whole framework is presented with the application to interference management in wireless networks in mind but all the ideas are general and transfer to other domains. We show that the proposed framework can achieve an improvement of up to 130 % over the state-of-the-art interference management policies.

Since the framework we developed was centralized we need to extend it to a distributed setting, which is fairly non-trivial. In Chapter 4, we leverage ideas from graph theory to propose new distributed maximal independent set generation algorithms and combine them with state-of-the-art distributed Alternating Direction Method of Multipliers (ADMM) to arrive at the proposed distributed solutions. We are able to show that the proposed solution is distributed, achieves a constant factor approximation (for many scenarios that extend beyond interference management) and is computationally efficient. We show that the proposed framework can achieve an improvement of up to 700 % over the state-of-the-art distributed interference management policies.

### 1.2.1.2 Dynamic Matching with Strategic Agents

In Chapter 5, we study multi-agent resource allocation with strategic agents in a matching environment (based on [AS16])). Suppose the two sides to be matched are workers and clients/tasks such as in crowdsourcing. Each client wants to hire one worker, i.e., the workers contend for one slot, which is the resource. Both the client side and the worker side learn preference for each other by being matched over time. Although we describe this matching environment with workers and clients, the setup proposed is general and applies to other matching environments as well. We derive mechanisms that ensure that in equilibrium the final matches that are achieved satisfy stability (an appropriate notion defined later) and achieve social optimality (for instance, maximum total output). We also carry out numerical experiments to show the effectiveness of the proposed mechanism.

### 1.2.1.3 Dynamic Resource Allocation Planning

In Chapter 6 (based on [AZS17]), we focus on resource allocation in stochastic environments. The abstract formulation of the problem is described as follows. We are given a budget of how many times we are allowed to sample from a stochastic process. Hence, sampling is a limited resource. Every time we sample we gain some information on the underlying state and our goal is to track the stochastic process as well as possible without missing certain key states. Existing frameworks in the literature formulate this problem as a Partially Observable Markov Decision Process (POMDP) and often assume that the underlying stochastic process is a Markov Process. We derive a general framework that does not require us to make the assumption that the underlying stochastic process is Markov. We provide provable performance guarantees for the proposed framework and show that the performance can be achieved in polynomial time.

The proposed framework is motivated from screening for cancers. In diseases such as breast cancer, it is common to have screening programs. For instance, in the US, the women between the age of 45-54 are advised to do a mammogram every year. In other countries, such as Canada and Japan, the women are advised to do a mammogram every two years after

the age of 40. Taking a mammogram can result in false positive and lead to unnecessary biopsies, which is detrimental to the subject. Therefore, the screening policies should be personalized to the patient’s history of past tests and to the patient’s family history and other static features. We use our methodology and apply it to breast cancer screening. We are able to reduce the number of screens that are needed by 40-50 %.

### 1.2.2 Optimization Methods for Machine Learning Applications

#### 1.2.2.1 Black-Box Model Interpretation

Machine learning models are increasingly being used in many critical decision making systems such as clinical decision support systems. Some systems that we described above are based on data-driven models such as the breast cancer screening system. Many of these data-driven models (for e.g., random forest based models, deep neural network based models) are often hard to interpret and thus are regarded as black-boxes. In Chapter 7, we propose a framework (based on our work in [AZS18]) that takes as input a black-box model and returns as output a piecewise linear approximation of it. The main premise of the work is that linear models are easier to interpret and hence, breaking a model into piecewise linear functions can be very useful in certain cases. In general, constructing optimal piecewise linear approximations is a non-trivial problem because the number of ways to divide the feature space into pieces is extremely large. We provide provable guarantees to show that our method outputs efficient approximation of the black-box and also carry out simulations on several real datasets to establish the utility of our proposed approach.

#### 1.2.2.2 Kullback-Leibler divergence estimation

Kullback-Leibler (KL) divergence is a fundamental quantity used in machine learning and statistics. In Chapter 8 (based on our work in [Ahu19]), we propose an optimization based approach for estimating the KL divergence.

# CHAPTER 2

## Background

In this chapter, we provide a brief introduction to the different areas of optimization and game theory that would be useful for understanding this dissertation. This chapter only provides an introduction to some basic concepts that can help a reader not familiar with the areas get started. To gain an in-depth understanding of these areas, please refer to the detailed materials in [BV04] [BPC11] [Kri16] [Erc13] [SS01] [BBB95].

### 2.1 Background for Resource Allocation in Multi-Agent Systems

The first part of this section describes optimization problems and some of their important properties. This part is heavily based on [BV04]. This section would be useful to understand Chapters 3 and 4.

#### 2.1.1 Standard Formulation of Optimization Problem

Consider a set of functions  $\{f_i\}_{i=0}^m$  and  $\{h_i\}_{i=1}^p$ , where  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ . We define a standard optimization problem below.

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } f_i(x) \leq 0, i = 1, \dots, m \\ & \quad h_i(x) = 0, i = 1, \dots, p \end{aligned} \tag{2.1}$$

where  $x \in \mathbb{R}^n$  is the optimization variable,  $f_0$  is the objective function,  $f_i$  is the inequality constraints, and  $h_i$  is the equality constraint.

### 2.1.2 Convex Sets

A set is defined as a convex set if any line segment joining two points in the set also belongs to the set. Formally stated, a set  $C$  is convex if for any  $x_1, x_2$  that are in the set  $C$ , the line segment  $z = \theta x_1 + (1 - \theta)x_2$  is also in the set  $C$ , where  $0 \leq \theta \leq 1$ .

### 2.1.3 Convex Functions

A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex if the domain of  $f$  is a convex set and

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$$

for all  $x_1, x_2$  in the domain  $f$  and for all  $0 \leq \theta \leq 1$ .

### 2.1.4 Standard Formulation of Convex Optimization Problem

If the functions  $\{f_i\}_{i=0}^m$  are convex (2.1), and the equality constraints  $\{h_i\}_{i=1}^p$  are affine, then the optimization problem in (2.1) is a standard form of a convex optimization problem. Hence, we can rewrite the convex optimization problem below as follows.

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } f_i(x) \leq 0, i = 1, \dots, m \\ & \quad a_i^t x = b_i, \quad i = 1, \dots, p \end{aligned} \tag{2.2}$$

### 2.1.5 Local Minima

Local minima of a function is a point in the domain of the function, where the function is lower than the neighborhood points. Suppose  $x$  is a local minimum. Then it satisfies the following conditions.  $x$  is feasible and  $\exists R > 0$  such that

$$f_0(x) = \inf\{f_0(z); z \text{ is feasible and } \|z - x\| \leq R\}$$

### 2.1.6 Global Minima

If the optimization problem is a convex optimization problem, then a local minimum of that problem is also the global minimum. This property of convex optimization problems makes them unique. There are many computational procedures such as the gradient descent method and Newton's method that can be used to find the local minimum of an optimization problem. Hence, it is easy to find the global minimum of a convex optimization problem. In many cases, a problem is formulated as a non-convex optimization problem but it can be equivalently reformulated as a convex optimization problem.

### 2.1.7 Duality

We first define the Lagrange dual function. We consider the optimization problem in (2.1). Define the domain  $\mathcal{D}$  as the intersection of the domains of functions  $\{f_i\}_{i=0}^m$  and  $\{h_i\}_{i=1}^p$ . We assume that the optimization problem in (2.1) is feasible and thus an optimal value exists, which is equal to  $p^*$ . Define the Lagrangian  $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$  as follows.

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{j=1}^p \nu_j h_j(x) \quad (2.3)$$

where  $\lambda_i$  is the Lagrange multiplier for  $f_i(x) \leq 0$ ,  $\nu_j$  is the Lagrange multiplier for  $h_j(x) = 0$ ,  $\lambda = [\lambda_1, \dots, \lambda_m]$ , and  $\nu = [\nu_1, \dots, \nu_p]$ .

The Lagrange dual function is defined as follows.

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \quad (2.4)$$

$g(\lambda, \nu)$  is a point-wise infimum of a set of affine functions, thus we can conclude that it is concave [BV04]. We now state some important properties of the dual function. If  $\lambda_i \geq 0$ ,  $\forall i$ , then the dual function provides a lower bound on the optimal value  $p^*$ . We can write this as  $p^* \geq g(\lambda, \nu)$ .

### 2.1.8 The Dual Problem

The dual problem is stated as follows.

$$\begin{aligned} & \max g(\lambda, \nu) \\ & \lambda_i \geq 0, \forall i \in \{1, \dots, m\} \end{aligned} \tag{2.5}$$

We denote the optimal solution of the above dual problem as  $d^*$ .

#### 2.1.8.1 Weak Duality

The optimal solution of (2.1), which is also referred to as the primal problem, is always greater than or equal to the optimal solution of the dual problem  $p^* \geq d^*$ . The above inequality is always true for both convex and non-convex problems. Since  $g(\lambda, \nu)$  is concave, the above dual problem is a concave maximization problem and hence, it is easier to solve. Therefore, we can use weak duality combined with the concave nature of the problem to arrive at the conclusion that we can find a lower bound to the optimal solution  $p^*$  tractably.

#### 2.1.8.2 Strong Duality

The optimal solution of (2.1) is equal to the optimal solution of the dual problem  $p^* = d^*$ . This condition is usually true for convex optimization problems. There are different conditions in the literature that when satisfied ensure that strong duality holds [BV04]. One example is if the optimization problem is convex, and it is strictly feasible, i.e., there exists a solution to the constraints  $f_i(x) < 0, \forall i \in \{1, \dots, m\}$ .

We refer the readers to further explore the theory and applications of convex optimization in [BV04]. Since the first part of the dissertation is focused on distributed optimization we next introduce some key concepts in distributed optimization. Our discussion on distributed optimization is based on [BPC11].

### 2.1.9 Centralized vs Distributed Optimization

A problem is centralized if one centralized decision maker/computing unit solves the problem. A problem is distributed if there is more than one computing unit/decision maker cooperating to solve the problem. There are multiple reasons that make distributed optimization important. Firstly, one centralized computing unit may not be equipped with enough memory to handle the data or there might be a constraint on the time to compute. With multiple computational units, which work distributedly, both the memory and time can be potentially addressed. In many cases, the decision makers are not co-located, which is another reason why distributed optimization is so important. For instance, in ad-hoc wireless communication networks, the devices are geographically distributed and need to make decisions in a distributed manner.

### 2.1.10 Dual Decomposition

We assume that the objective  $f_0$  is separable, i.e.,  $f_0(x_1, \dots, x_N) = \sum_{k=1}^N f_0^k(x_k)$ , where  $f_0^k(x_k)$  is the objective for the  $k^{th}$  decision maker. Suppose we are considering the following equality constrained optimization problem.

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } \sum_{i=1}^n A_i x_i = b \end{aligned} \tag{2.6}$$

where  $A_i \in \mathbb{R}^{p \times n_i}$  is the matrix associated with  $x_i \in \mathbb{R}^{n_i}$  and  $b \in \mathbb{R}^p$ . A more succinct representation of the above problem is

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } Ax = b \end{aligned} \tag{2.7}$$

where  $A = [A_1, \dots, A_N]$  and  $x = (x_1, \dots, x_N)$

We write the Lagrangian for the above problem as

$$L(x, \lambda) = \sum_{k=1}^N f_0^k(x_k) + \lambda^t (\sum_{k=1}^N A_k x_k - b)$$

Define  $L_k(x_k, \lambda) = f_0^k(x_k) + \lambda^t A_k x_k$ . Hence, we can write

$$L(x, \lambda) = \sum_{k=1}^N L_k(x_k, \lambda) - \lambda^t b \quad (2.8)$$

The dual decomposition method's key steps are described as follows.

- $x_k^{l+1} = \arg \min L_k(x_k^l, \lambda), \forall k \in \{1, \dots, N\}$
- $\lambda^{l+1} = \lambda^l + \alpha^l (\sum_k A_k x_k - b)$

where  $\alpha^l > 0$  is the step size. Dual decomposition methods converge to the optimal solution under strong assumptions. The main advantage of the dual decomposition is that the  $x_k$  can be computed in parallel. The computation of  $\lambda$  by all the decision makers requires communication of current estimates of  $x_k$  unless there is a centralized entity gathering  $x_k$  to compute  $\lambda$ .

### 2.1.11 Augmented Lagrangian

In this section, we discuss augmented Lagrangian that were developed to make the dual decomposition method more robust. The objective function is modified as follows:  $f_0(x) + \rho \|Ax - b\|^2$ . We define the augmented Lagrangian  $L_\rho$  as  $L_\rho(x, \lambda) = f_0(x) + \lambda^t(Ax - b) + \rho \|Ax - b\|^2$ . We write the optimality conditions (Karush-Kuhn-Tucker also known as the KKT conditions). Suppose  $x^*$  and  $\lambda^*$  are the optimal values.

$$\begin{aligned} Ax^* - b &= 0 \\ \nabla f(x^*) + A^t \lambda^* &= 0 \end{aligned} \quad (2.9)$$

The steps under the augmented Lagrangian method are given below.

- $x^{l+1} = \arg \min L_\rho(x, \lambda^l)$
- $\lambda^{l+1} = \lambda^l + \rho(Ax^{l+1} - b)$

The above approach converges to the optimal solution  $x^*, \lambda^*$  under much more relaxed conditions in comparison to the dual decomposition approach. The strict convexity due to

adding the penalty  $\rho\|Ax - b\|^2$  ensures that we do not need  $f_0$  to be differentiable. However, the disadvantage now is that the update for  $x$  cannot be carried out in parallel as we lost separability by adding the penalty term.

### 2.1.12 Alternating Direction Method of Multipliers

In this section, we briefly discuss Alternating Direction Method of Multipliers (ADMM) that is built to overcome the limitations of the methods described in the previous sections. We consider two convex functions  $f$  and  $g$  and say the objective is  $f(x) + g(z)$ . The optimization problem under consideration is

$$\begin{aligned} & \min f(x) + g(z) \\ & \text{subject to } Ax + Bz = c \end{aligned} \tag{2.10}$$

We define the Lagrangian for the above problem as follows:  $L_\rho(x, z, \lambda) = f(x) + g(z) + \rho\|Ax + Bz - c\|^2 + \lambda^t(Ax + Bz - c)$ . The steps in the ADMM method are described below.

- $x^{l+1} = \arg \min L_\rho(x, z^l, \lambda^l)$
- $z^{l+1} = \arg \min L_\rho(x^l, z, \lambda^l)$
- $\lambda^{l+1} = \lambda^l + \rho(Ax^{l+1} + Bz^{l+1} - c)$

This method allows  $x$  and  $z$  to be updated separately in parallel. Update of  $\lambda$  requires both  $x$  and  $z$ . Note this method overcomes the limitation of both dual decomposition, which requires that the function be differentiable and the augmented Lagrangian based approach, which does not permit parallel updates.

### 2.1.13 Graph Theory Basics and Algorithms

Graph theory based algorithms have been commonly used for many resource allocation problems. In this section, we give a brief overview of some definitions and concepts in graph theory. This part is based on [Erc13]

Define a set of vertices  $V$ . In this dissertation, we will only deal with finite graphs, i.e. the set  $V$  is finite. Define an edge  $e$ , where  $e = (v_1, v_2)$  is an ordered pair of the vertices. Let the set of edges be  $E$ .

**Definition 1** *Graph: A graph is defined as a tuple  $G = (V, E)$ , where  $V$  is the set of vertices and  $E$  is the set of edges.*

**Definition 2** *Vertex Adjacency: Two vertices  $v_1 \in V$  and  $v_2 \in V$  are said to be adjacent if there is an edge  $e \in E$  such that  $e = \{v_1, v_2\}$*

**Definition 3** *Edge Adjacency: Two edges  $e_1 \in E$  and  $e_2 \in E$  are said to be adjacent if there is a vertex  $v \in V$  that is incident to both the edges.*

**Definition 4** *Neighborhood: Neighborhood of a vertex  $v$   $N(v)$  is the set of vertices that are adjacent to  $v$ . Formally,  $N(v) = \{u \in V, s.t. \{u, v\} \in E\}$*

**Definition 5** *Adjacency Matrix: The adjacency matrix  $A$  of a graph with  $n$  vertices is an  $n \times n$  matrix such that the element at  $(i, j)$   $A[i, j]$  is one if the vertex  $i$  and  $j$  are connected and is zero otherwise.*

### 2.1.13.1 Vertex Coloring

The objective is to color the vertices of the graph such that no two neighbors are assigned the same color. Vertex coloring has different applications such as resource allocation in wireless networks, task scheduling etc.

**Definition 6** *Vertex Coloring: Vertex coloring is a procedure of assigning a color  $c_v$  to each vertex  $v \in V$  such that  $c_v$  is different from any of the colors assigned to the neighbors of  $v$ .*

**Definition 7** *Chromatic Number: The chromatic number of a graph  $G$ ,  $\chi(G)$ , is the minimum number of colors needed to color  $G$ .*

Calculation of  $\chi(G)$  is Non-deterministic Polynomial (NP) complete [Erc13].

#### 2.1.13.2 Sequential Vertex Coloring Algorithm

We label the vertices of the graph in a certain order  $v_1, v_2, \dots, v_n$ . We start by describing a simple greedy vertex coloring algorithm. In this algorithm, we pick the uncolored vertex from the remaining vertices uniformly. Suppose that the maximum number of vertices in the neighborhood of a certain vertex is  $\Delta$ . Suppose we have a palette that consists of a total of  $\mathcal{O}(\Delta + 1)$  colors. We define an array of colors  $\text{neigh}_{\text{colors}}$ , where  $\text{neigh}_{\text{colors}}[v]$  is the set of colors that have been used to color the neighbors of  $v$  so far. The algorithm proceeds by assigning the smallest available color to an uncolored vertex from the palette that its neighbors have not been assigned yet. Once a color is assigned to  $v$  that color is added to the array  $\text{neigh}_{\text{colors}}$  for each neighbor of  $v$ . The time complexity of the algorithm is  $\mathcal{O}(n)$  and it uses  $\mathcal{O}(\Delta + 1)$  colors.

#### 2.1.13.3 Maximal Independent Sets

**Definition 8** *Independent Set:* An independent set  $S$  of the graph is a subset of the vertices  $V$  such that no two vertices in the set  $S$  have an edge connecting them

**Definition 9** *Maximal Independent Set (MIS):* Maximal Independent Set of a graph is an independent set to which no further vertices can be added without losing independence.

**Definition 10** *Maximum Independent Set:* Maximum Independent Set is the largest independent set for a graph.

Finding the maximum independent set of a graph is an NP-hard problem [Erc13].

#### 2.1.13.4 Sequential Maximal Independent Set Algorithm

We describe a simple algorithm to compute an MIS. Consider set  $S$ , which at the start of the algorithm consists of all the vertices in the graph. Arbitrarily select a vertex  $v$  from the set of remaining vertices  $S$ . Update  $S$  by removing the neighbors of  $v$  and  $v$  from the set  $S$ .

Repeat the first step, i.e. select a vertex arbitrarily from  $S$  and remove it and its neighbors from  $S$ . Continue to do this procedure until the set  $S$  is empty.

## 2.2 Background for Dynamic Matching with Strategic Agents

Game theory is a study of interactions between strategic agents. These strategic interactions can be found everywhere around us such as poker, chess, designing contracts, international trade or hiring workers. This section is based on [Mih16]. This section would be useful to understand Chapter 5.

### 2.2.1 Non-Cooperative Game Theory

Non-cooperative games characterize interactions between individuals. The main characteristic of non-cooperative games is that there is no external entity to enforce agreements and the individuals behave in such a way that it has to be self-enforcing. We discuss the most basic form of a game, which is referred to as normal or strategic game. Suppose  $N = \{1, \dots, n\}$  is a finite set of players.  $S_i$  is the set of pure strategies for player  $i$ .  $S = S_1 \times S_2 \dots \times S_n$  is the set of pure strategy profiles.  $S_{-i}$  is the set of pure strategy profiles for  $i$ 's opponents.  $u_i : S \rightarrow \mathbb{R}$  is the payoff function of player  $i$ . We write the tuple of the payoffs of all the players as  $u = (u_1, \dots, u_n)$ . The normal form game is defined as the tuple  $(N, S, u)$ . The structure of the game is common knowledge, i.e., every player knows the tuple, every player knows that everyone knows the tuple, and so on. If  $S$  is a finite set, then the game is finite. Define  $\Delta(X)$  as the set of probability measures over  $X$ .  $\Delta(S_i)$  is the set of mixed strategies for player  $i$ .  $\sigma \in \Delta(S_1) \times \Delta(S_2) \dots \times \Delta(S_n)$  is a mixed strategy profile for all the players.

Each player  $i$  has Von Neumann-Morgenstern preferences, i.e., expected utility is defined as  $u_i(\sigma) = \sum_{s \in S} u_i(s)\sigma(s)$ . Next, we try to characterize ways in which a game is played in terms of solution concepts. We start with the example of a very simple game, prisoner's dilemma. There are two persons arrested. Each person has the option of taking one of the following two actions- cooperate (C) and defect (D). If both defect, then both are accused of a minor crime and sent to one year in prison. If both cooperate, then both are accused of a

major crime and sent to prison for 2 years. If Player 1(2) cooperates and Player 2(1) defects, then Player 2(1) is accused and sent to 3 years in prison and Player 1(2) is set free. Observe that defecting is strictly beneficial for each player, i.e., under both strategies for Player 2(1) it is strictly better for Player 1(2) to defect. We give the actions and payoffs in Table 2.1.

Table 2.1: Utility matrix for prisoner's dilemma.

	C	D
C	(-2,-2)	(-3,0)
D	(0,-3)	(-1,-1)

Next we define Nash Equilibrium, which is a solution concept central to non-cooperative game theory.

**Definition 11** *Nash Equilibrium:* A mixed strategy  $\sigma$  is a Nash Equilibrium if for every  $i \in N$ ,

$$u_i(\sigma_i, \sigma_{-i}) \geq u_i(s_i, \sigma_{-i}) \forall s_i \in S_i$$

For the game that we defined above, (D,D) is the Nash Equilibrium (as defect is the dominant strategy for both the players). The reader can refer to [Mih16] and references in there for further exposure to game theory.

## 2.3 Background for Dynamic Resource Allocation Planning

In this section, we give a brief background on Markov decision processes, which would be relevant for understanding Chapter 6. This section is heavily based on [Kri16].

### 2.3.1 Fully Observed Markov Decision Processes

Consider a discrete time stochastic system with state  $x_k \in \mathcal{X}$ , where  $k$  is the index of the time and  $k \in \{0, 1, \dots, N - 1\}$ .

$$x_{k+1} = A_k(x_k, u_k, w_k) \tag{2.11}$$

The initial  $x_0$  has a distribution  $\pi_0$ .  $\{w_k\}$  is an i.i.d. process with probability density  $p_w$  that is statistically independent of the initial state  $x_0$ . Also,  $u_k \in \mathcal{U}$  is the action taken by a decision-maker, where  $\mathcal{U}$  denotes the set of actions. We assume that the decision maker observes the state at each time  $k$ . If action  $u_k$  is taken at the time instance  $k$ , an instantaneous cost  $c(u_k, x_k, k)$  is incurred at  $k$ . We define the Markov transition probabilities

$$\mathbb{P}(x_{k+1} \in B \mid x_k = x, u_k = u) = \int \mathbb{P}(I(A_k(x, u, w) \in B)) p_w(w) dw \quad (2.12)$$

where  $B$  denotes a measurable set and  $I$  is an indicator function, which takes the value one when the condition inside is true and zero otherwise. We use the Dirac Delta function to obtain the following

$$p(x_{k+1} | x_k = x, u_k = u) = \int \delta(x_{k+1} - A_k(x, u, w)) p_w(w) dw \quad (2.13)$$

In summary, a discrete-time Fully Observable Markov Decision Process is characterized using the following tuple

$$(\mathcal{X}, \mathcal{U}, p(x_{k+1} | x_k = x, u_k = u), c(u_k, x_k, k)) \quad (2.14)$$

The decision maker chooses a sequence of the actions  $u_0, u_1, \dots, u_N$  and as a result, the random process moves into different states  $x_0, x_1, \dots, x_N$ . Suppose  $\mathcal{H}_k$  is the set of observations made by the decision maker up to time  $k$ . We define the policy at time  $k$ ,  $\pi_k$ , as a mapping from  $\mathcal{H}_k$  to  $\mathcal{U}$ . We define the policy  $\pi$  as  $\pi = (\pi_0, \dots, \pi_{N-1})$ . We define a finite horizon objective function for the decision maker as

$$V_\pi(x) = \mathbb{E}_\pi \left[ \sum_{k=0}^{N-1} c_k(x_k, \pi(\mathcal{H}_k), k) + c_N(x_N) \mid x_0 = x \right] \quad (2.15)$$

where  $\mathbb{E}_\pi$  is the expectation with respect to (w.r.t) the joint probability distribution on the histories in  $\mathcal{H}_N$ . We refer to  $V_\pi$  as the value function w.r.t. to policy  $\pi$ . The goal of the decision-maker is to solve for the optimal policy given as

$$\pi^* = \operatorname{argmin}_\pi V_\pi(x) \quad (2.16)$$

A general policy  $\pi_k$  maps the entire history up to time  $k$  to action  $u_k$ . If a policy makes a decision only based on the most recent state  $x_k$ , i.e.,  $\pi_k(x_k) = u_k$ , then the policy is Markovian. It is sufficient to search for the optima in the space of Markov policies to achieve the minimum in (2.16) (See [Kri16]).

### 2.3.1.1 Bellman's Stochastic Dynamic Programming Algorithm

Bellman's stochastic dynamic programming algorithm relies on backward recursion. Initialize  $V_N(x) = c_N(x)$ . For  $k = N - 1, \dots, 0$ , the algorithm works as follows.

$$\begin{aligned} V_k(x) &= \min_{u \in \mathcal{U}} \left\{ c(x, u, k) + \int V_{k+1}(A_k(x, u, w)) p_w(w) dw \right\} \\ \pi_k^*(x) &= \operatorname{argmin}_{u \in \mathcal{U}} \left\{ c(x, u, k) + \int V_{k+1}(A_k(x, u, w)) p_w(w) dw \right\} \end{aligned} \quad (2.17)$$

Finally, the optimal policy is  $\pi^* = (\pi_0^*, \pi_1^*, \dots, \pi_{N-1}^*)$ . The total cost associated with the optimal policy  $\pi^*$  denoted as  $V_{\pi^*}$  is computed based on (2.16).

**Theorem 1** *The output of the stochastic dynamic programming algorithm  $\pi^*$  achieves the minimum in (2.16)*

In this section, we discussed fully observed Markov decision processes. In the special case, where the number of possible states is finite, the objective and the recursion in (2.16) and (2.17) can be simplified (See [Kri16] for more details).

### 2.3.2 Partially Observable Markov Decision Process

We first define the building blocks of a Partially Observable Markov Decision Process (POMDP). Time is discrete and the planning horizon is finite. Each time index is  $k \in \{0, \dots, N - 1\}$ .  $\mathcal{X} = \{1, \dots, X\}$  is a finite state space.  $\mathcal{U} = \{1, \dots, U\}$  is a finite action space.  $\mathcal{Y}$  denotes the observation space which can either be finite or a subset of  $\mathbb{R}$ .  $x_k$  and  $u_k$  are the states and actions respectively in time slot  $k$ . For each  $u \in \mathcal{U}$ ,  $P(u)$  denotes a  $X \times X$  transition matrix with each element defined as  $P_{ij}(u) = \mathbb{P}(x_{k+1} = j | x_k = i, u_k = u)$ .

For each  $u \in \mathcal{U}$ ,  $B(u)$  denotes the observation distribution with each element defined as  $B_{iy}(u) = \mathbb{P}(y_{k+1} = y|x_{k+1} = i, u_k = u)$ . For state  $x_k$  and action  $u_k$  pair, the cost  $c(x_k, u_k)$  and at terminal time  $N$  the cost incurred is  $c_N(x_N)$ . Hence, a POMDP is characterized by the tuple given below. Define the prior distribution over the initial state  $x_0$  as  $b_0 = \{\mathbb{P}(x_0 = i), \forall i \in \mathcal{X}\}$ . The sequence of policies for time slots  $\{0, \dots, N-1\}$  are given as  $\pi = (\pi_0, \dots, \pi_{N-1})$ . The total expected cost of the decision-maker is given as

$$V_\pi(b_0) = \mathbb{E}_\pi \left[ \sum_{k=0}^{N-1} c(x_k, u_k) + c_N(x_N) | b_0 \right] \quad (2.18)$$

where  $\mathbb{E}_\pi$  is the expectation w.r.t to the history induced by the policy  $\pi$ . The goal of the decision-maker is to find the optimal policy such that the objective defined above is minimized, i.e.,

$$\pi^* = \operatorname{argmin} V_\pi(b_0) \quad (2.19)$$

In the case of fully observed MDPs described in the previous section, we discussed how the most recent state  $x_k$  is a sufficient statistic and hence, it is sufficient to search the space of Markovian policies. In POMDPs since we don't necessarily observe the state  $x_k$  we can only form a belief over the states conditioned on the observations. This belief as we discuss next forms a sufficient statistic for the POMDPs. We define the belief distribution as

$$b_k(i) = \mathbb{P}(x_k = i|h_k) \quad (2.20)$$

where  $h_k = (\mu_0, y_0, u_0, \dots, y_k, u_k)$  and  $i \in \mathcal{X}$ . We write the belief vector as

$$b_k = [b_k(0), \dots, b_k(N-1)]$$

We use the observation distribution defined above to define a  $X \times X$  diagonal matrix given as  $B_y(u) = \operatorname{diag}(\mathbb{P}(y_{k+1} = y|x_{k+1} = 0, u_k = u), \dots, \mathbb{P}(y_{k+1} = y|x_{k+1} = N-1, u_k = u))$ . The belief is updated based on the observation in time slot  $k$  as follows.

$$b_{k+1} = T(b_k, y_{k+1}, u_k) = \frac{B_{y_{k+1}}(u_k) P^t(u_k) b_k}{\sigma(b_k, y_{k+1}, u_k)} \quad (2.21)$$

where  $\sigma(b_k, y_{k+1}, u_k) = \mathbf{1}^t P^t(u_k)b_k$ , where  $\mathbf{1}$  is a  $N$  dimensional column vector.

$$\begin{aligned}
V_\pi(b_0) &= \mathbb{E}_\pi \left[ \sum_{k=0}^{N-1} c(x_k, u_k) + c_N(x_N) | b_0 \right] \\
&= \mathbb{E}_\pi \left[ \sum_{k=0}^{N-1} \sum_{i=1}^X c(i, u_k) b_k(i) + \sum_{i=1}^X c_N(i) b_k(i) | b_0 \right] \\
&= \mathbb{E}_\pi \left[ \sum_{k=0}^{N-1} c_{u_k}^t b_k + c_N^t b_k | b_0 \right]
\end{aligned} \tag{2.22}$$

where  $c_{u_k} = [c(1, u_k), \dots, c(N, u_k)]^t$  and  $c_N = [c_N(1), \dots, c_N(X)]^t$ . Based on the above equation (2.22), we realize that the belief vector  $b_k$  can be understood as the state vector and then this equation can be analyzed similarly to the Fully Observed Markov Decision Process in (2.15). This also leads us to realize that we only need to search in the space of policies that map from the belief vector to the actions. The optimal policy  $\pi^* = (\pi_0^*, \dots, \pi_{N-1}^*)$  for a POMDP can be obtained as a solution to the following backward recursion. Initialize  $V_N(b) = c_N^t b$  and then for  $k = N - 1, \dots, 0$ .

$$\begin{aligned}
V_k(b) &= \min_{u \in \mathcal{U}} \left\{ c_u^t b + \sum_{y \in \mathcal{Y}} V_{k+1}(T(b, y, u) \sigma(b, y, u)) \right\} \\
\pi_k^*(b) &= \operatorname{argmin}_{u \in \mathcal{U}} \left\{ c_u^t b + \sum_{y \in \mathcal{Y}} V_{k+1}(T(b, y, u) \sigma(b, y, u)) \right\}
\end{aligned} \tag{2.23}$$

The total number of belief vectors forms an uncountable set. Hence, the above recursive algorithm is not tractable. There are certain standard approaches described in [Kri16] that help overcome these limitations. For further discussion on POMDPs refer to [Kri16].

In Chapter 7, we rely on principles of dynamic programming, which we briefly discussed for the stochastic settings in 2.3.1.1, to construct our Algorithm. For further discussion on dynamic programming refer to [BBB95]. We did not cover the relevant background for Chapter 8 as it is quite involved and is out of the scope of this chapter. Instead, we refer the reader to [SS01].

## **2.4 Conclusion**

In this chapter, we gave a brief background for a reader new to these areas. This background would be useful in understanding the next chapters. A reader further interested should explore the references we cited in the chapter.

# CHAPTER 3

## Centralized Large Scale Multi-Agent Resource Sharing

### 3.1 Introduction

In this chapter, we describe a large scale resource sharing method for multi-agent systems. We describe the proposed method in the context of wireless networks. At the end of the chapter, we show that the proposed method is general and applies to many scenarios. Our objective is to solve an optimization problem that determines how should a resource be shared among multiple agents over a long period of time. We show that computing the exactly optimal solution to this problem is computationally intractable. We develop a solution that runs in polynomial time and in many cases is able to achieve a constant factor approximation in terms of the performance. This chapter is based on [AXS15b].

**Motivation.** As more and more devices are connecting to cellular networks, the demand for wireless spectrum is exploding. Dealing with this increased demand is especially difficult because most of the traffic comes from bandwidth-intensive and delay-sensitive applications such as multimedia streaming, video surveillance, video conferencing, gaming etc. These demands make it increasingly challenging for the cellular operators to provide sufficient quality of service (QoS). Dense deployment of distributed low-cost femtocells (or small cells in general, such as microcells and picocells) has been viewed as one of the most promising solutions for enhancing access to the radio spectrum [GMR12], [ACD12]. Femtocells are attractive because they can both extend the service coverage and boost the network capacity by shortening the access distance (cell splitting gain) and offloading traffic from the cellular network (offloading gain). However, in a closed access network when only registered mobile users can connect to the femtocell base station, dense deployment of femtocells operating in

the same frequency band leads to strong co-tier interference. In addition, since the macrocell users usually operate in the same frequency, the problem of interference (to both femtocells and macrocells) is further exacerbated due to cross-tier interference across macrocells and femtocells. In this chapter, we study a closed access network. Hence, it is crucial to design interference management policies to deal with both co-tier and cross-tier interference.

Interference management policies specify the transmission scheduling and transmit power levels of femtocell user equipments (FUEs) and macrocell user equipments (MUEs) in uplink transmissions, and specify the transmission scheduling and power levels of femtocell base stations (FBSs) and macrocell base stations (MBSs) in downlink transmissions. We focus on uplink transmissions in this chapter, but our framework can be easily applied to downlink transmissions. An efficient (interference management) policy should fulfill the following important requirements (as we will discuss in details in Section 3.2, state-of-the-art policies do not fulfill one or more of the following requirements):

- *Interference management based on network topology:* Effective interference management policies must take into account that uplink transmissions from neighboring UEs create strong mutual interference, but must also recognize and take advantage of the fact that non-neighboring UEs do not. Hence, the network topology (i.e. locations of femtocells/macrocells) must play a crucial role.
- *Limited signaling for interference coordination:* In dense, large-scale femtocell deployments, the UEs cannot coordinate their transmissions by sending a large amount of control signals across the network. Hence, effective interference management policies should not rely on heavy signaling and/or message exchanges across the UEs in the network.
- *Scalability (in terms of performance and complexity) in large networks:* Femtocell networks are often deployed on a large scale (e.g. in a city). Effective interference management policies should scale in large networks, namely achieve efficient network performance while maintaining low computational complexity.

- *Support for delay-sensitive applications:* Effective interference management policies must support delay-sensitive applications, which constitute the majority of wireless traffic.
- *Versatility in optimizing various network performance criteria:* The appropriate network performance criterion (e.g. weighted sum throughput, max-min fairness, etc.) may be different for different networks. Effective interference management policies should be able to optimize a variety of network performance criteria while ensuring performance guarantees for each MUE and each FUE.

In this chapter, we propose a novel, systematic, and practical methodology for designing and implementing interference management policies that fulfill *all* of the above requirements. Specifically, our proposed policies aim to optimize a given network performance criterion, such as weighted sum throughput and max-min fairness, subject to each UE's minimum throughput requirements. Our proposed policies can efficiently manage a wide range of interference. We manage strong interference between neighboring UEs by using time-division multiple access (TDMA) among them. We take advantage of weak interference between non-neighboring UEs by finding maximal sets of UEs that do not interfere with each other and allowing all the UEs in those sets to transmit at the same time. More specifically, we find the maximal independent sets (MISs)<sup>1</sup> of the interference graph<sup>2</sup>, and schedule different MISs to transmit in different time slots. The scheduling of MISs in our proposed policy is particularly designed for delay-sensitive applications: the schedule of MISs across time is not cyclic (i.e. the policies do not allocate transmission times to MISs in a fixed (weighted) round-robin manner), but rather follows a carefully designed non-stationary schedule, in which the MIS to transmit is determined adaptively online. For delay-sensitive applications, cyclic policies are inefficient because transmission opportunities (TXOPs) earlier in the cycle are more valuable than TXOPs later in the cycle (earlier TXOPs enhances the chances of

---

<sup>1</sup>A set of vertices in which no pair is connected by an edge is independent (IS) and if it is not a subset of another IS then it is MIS.

<sup>2</sup>Each vertex in the interference graph corresponds to a UE-BS pair, where the pair constitutes the BS and the UE it serves. An edge represents high interference from/to a neighboring vertex.

transmission before delay deadlines). The cyclic polices are unfair to UEs allocated to later TXOPs.

Another distinctive feature of our work is that we do not take the interference graph as given as in most existing works; instead, in our work we show how to choose the interference graph that maximizes the network performance. Specifically, in our construction of interference graphs, we determine how to choose the threshold on the distance between two cells, based on which we determine if there is an edge between them, in order to maximize the network performance. Moreover, we prove that under certain conditions, the proposed policy, computed based on the optimal threshold, can achieve the optimal network performance (weighted sum throughput) within a desired small gap. Note that for large networks, in general it is computationally intractable to find all the MISs of the interference graph [JYP88]. We propose efficient polynomial-time algorithms to find a subset of MISs, and prove that under a wide range of deployment scenarios, the proposed policy, computed based on the constructed subset of MISs, can achieve a constant competitive ratio (with respect to optimal weighted sum throughput) that is independent of the network size.

Finally, we summarize the main contributions of our work:

1. We propose interference management policies that are based on scheduling the MISs of the interference graph. The schedule of MISs is constructed in order to maximize the network performance criterion subject to minimum throughput requirements of the UEs. In addition, the schedule adapts to the delay sensitivity requirements of the UEs by scheduling transmissions in a non-stationary manner.
2. We construct the interference graph by comparing the distances between the BSs with a threshold (i.e. there is an edge between two cells if the distance between their BSs is smaller than the threshold). We develop a procedure to choose the optimal threshold such that the proposed scheduling of MISs leads to a high network performance. Importantly, we prove that under certain conditions, the proposed scheduling of MISs based on the optimal threshold achieves within a desired small gap of the optimal network performance (weighted sum throughput).

3. Since it is computationally intractable to find all the MISs in large networks, we propose an approximate algorithm that computes a subset of MISs within polynomial time. We prove that under a wide range of deployment scenarios, the proposed policy based on this subset of MISs has a constant competitive ratio (with respect to the optimal weighted sum throughput) that is independent of the network size.

The rest of the chapter is organized as follows. In Section 3.2, we discuss the related works and their limitations. We describe the system model followed by the problem formulation in Section 3.3 and 3.4, respectively. The design framework and its low-complexity variant for large networks are discussed in Section 3.5 and Section 3.6, respectively. In Section 3.7, we use simulations to compare the proposed policy with state-of-the-art policies. In Section 3.8, we discuss how the proposed framework is general and can be applied to other applications. Finally we conclude the chapter in Section 3.9.

## 3.2 Related Works

In this section we provide a comparison of state-of-the-art policies with the proposed policy. The interference management policies in the existing works can be categorized in two classes: 1) policies based on constant power control, and 2) policies based on spatial time/frequency reuse.

### 3.2.1 Interference Management Policies Based on Constant Power Control

The first and most widely-used interference management policies [HYC09, GI10, LLJ11, BPB13a, JMM09, CAM09] are based on constant power control. In these policies, all the UEs in the network transmit at a *constant* power at all time (provided that the system parameters remain the same) in the entire frequency band.<sup>3</sup> When the cross channel gains among BSs and UEs are high, simultaneous transmissions at the same time and in the same

---

<sup>3</sup>Although some power control policies [HYC09], [CAM09] go through a transient period of adjusting the power levels before converging to the optimal power levels, the users maintain the constant power levels after the convergence.

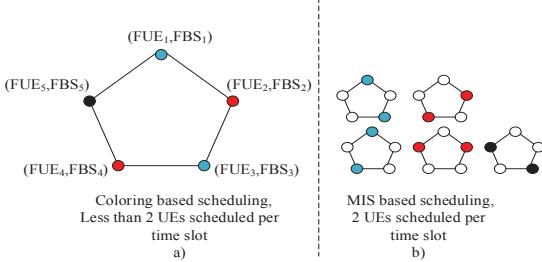


Figure 3.1: Coloring based scheduling in a) schedules less than two UEs per time slot on an average, while MIS based scheduling in b) is more efficient and schedules two UEs per time slot.

frequency band will cause significant interference among cells. Such strong interference is common in macrocells underlaid with femtocells. For example, in [CMK08] it is shown that interference from MUEs near the FBS severely affects the uplink transmissions of FUEs. Also, in offices and apartments, where FBSs are installed close to each other, inter-cell interference is particularly strong [LCV09]. In contrast, our proposed solutions mitigate the strong interference by letting only a subset of UEs (who do not interfere with each other much) to transmit at the same time (i.e. use time division multiplexing (TDM)).

### 3.2.2 Interference Management Based on Spatial Time/Frequency Reuse

Some existing works mitigate strong interference by letting different subsets of UEs to transmit in different time slots (spatial time reuse) [RL93, HS95, RP89, CS89, PST12, BBS06, ET90, AAS11, JPP05] or in different frequency channels (spatial frequency reuse) [LXH10, UAB11, LLJ10, LVD09, KL08, SHL12, Nec08]. Specifically, they partition UEs into disjoint subsets such that the UEs in the same subset do not interfere with each other [LXH10, UAB11, LLJ10, LVD09, KL08, SHL12, Nec08, BBS06, ET90, JPP05, AAS11, RL93, HS95, RP89, CS89, PST12].

Given the same partition of the UEs, the policies based on spatial time reuse and those based on spatial frequency reuse are equivalent. Hence, we focus on policies based on spatial time reuse hereafter.

Some policies based on spatial time reuse partition the UEs based on the coloring of the

interference graph [RL93, HS95, PST12] which is not efficient. In general, a set of UEs with the same color (i.e. the UEs who can transmit simultaneously) may not be maximal (See Fig. 3.1), in the sense that there may be UEs who do not interfere but have different colors (we will also show this in the motivating example in Section 3.4.2). In this case, it is more efficient to also let those non-interfering UEs to transmit simultaneously, although they have different colors. In other words, the partitioning based on coloring the interference graph is not efficient, because the average number of active UEs (i.e. the average cardinality of the subsets of UEs with the same color) can be low.

Some policies based on spatial time reuse [RP89, CS89, BBS06, ET90, JPP05, AAS11] partition the UEs based on the MISs of the interference graph, which is more efficient, because we cannot add any more UEs to an MIS without creating strong interference. However, they are still inefficient compared to our proposed policies for delay-sensitive applications. Specifically, they schedule different MISs in a cyclic and (weighted) round-robin manner, in which each UE transmits at a fixed position in each cycle. For delay-sensitive applications, earlier positions in the cycle are more desirable because they enhance the chances of transmitting prior to delay deadlines. Hence, a cyclic schedule is not fair to the UEs allocated to later positions. In contrast, our proposed policies schedule the MISs in an efficient, non-stationary manner for delay-sensitive applications.

Another notable difference from the existing works based on spatial time/frequency reuse is that they usually take the interference graph as given. On the contrary, our work discusses how to construct the interference graph optimally such that the network performance is maximized.

### 3.2.3 Other Interference Management Policies

Besides the above two categories, there are several other related works. For instance in [N BG10], [BPB13b], the authors propose reinforcement learning and evolutionary learning techniques for the femtocells to learn efficient interference management policies. In [N BG10], the femtocells learn the fixed transmit power levels, while in [BPB13b], the femtocells learn

to randomize over different transmit power levels. However, the interference management policies in [NBG10] and [BPB13b] cannot provide minimum throughput guarantees for the UEs. In contrast, we provide rigorous minimum throughput guarantees for the UEs. In both [NBG10], [BPB13b] the femtocell UEs need to limit their transmission powers in every time slot such that the signal to interference and noise ratio (SINR) of the macrocell UE is sufficiently high. If there is strong interference between some femtocells and the macrocell, the femtocell UEs will always transmit at lower power levels, leading to a low sum throughput for them.

Another method to mitigate interference is to use coordinated beam scheduling [CHH12], [YCC11]. In [CHH12] and [YCC11], the authors schedule a subset of beams to maximize the total reward associated with the scheduled subset, where the reward per beam reflects the channel quality and traffic. The first difference from our work is that the approach in [CHH12], [YCC11] schedules a fixed subset of beams and leaves the other UEs inactive. Hence, some UEs have no throughput, which means the minimum throughput as well as the delay-sensitivity of the UEs is not satisfied. Second, we rigorously prove that our proposed policy achieves good performance with low (polynomial-time) complexity, while [CHH12], [YCC11] do not. Third, the schemes in [CHH12], [YCC11] are proposed for a specific network performance criterion and may not be flexible enough for other network performance criteria (such as the minimum throughput). Finally, [CHH12], [YCC11] do not consider the delay sensitivity of the UEs.

### 3.3 System Model

#### 3.3.1 Heterogeneous Network of Macrocells and Femtocells

We consider a heterogeneous network of  $N$  femtocells (indexed by  $\{1, 2, \dots, N\}$ ) and  $M$  macrocells (indexed by  $\{N+1, \dots, N+M\}$ ) operating in the same frequency band, a common deployment scenario considered in practice [HYC09], [JMM09], [CAM09]. We assume that each FBS/MBS serves only one FUE/MUE as in [CAM09]. Our model can be easily

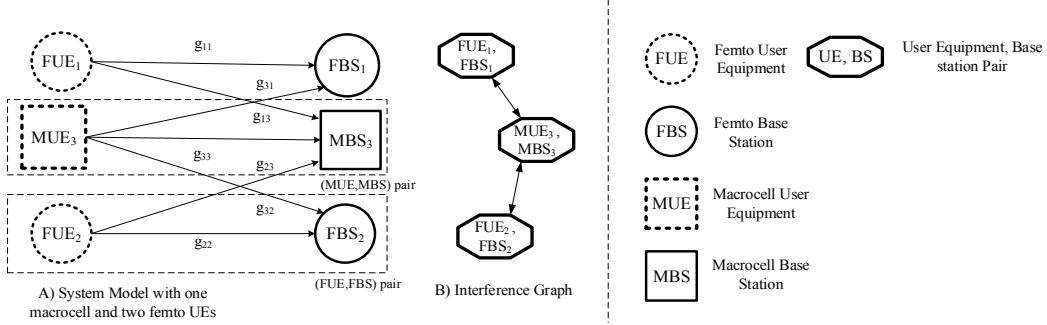


Figure 3.2: A) System model for the three-cell network, B) The interference graph for the three-cell network.

generalized to the setting where each BS serves multiple UEs, at the expense of complicated notations to denote the association among UEs and BSs. For notational clarity, we focus on the case where each BS serves one UE, and will demonstrate the applicability of our work to the setting where one BS serves multiple UEs in Section 3.7.

Since there is only one FUE or MUE in a femtocell or a macrocell, the index of each UE and that of each BS are the same as the index of the cell they belong to. We focus on the uplink transmissions. The proposed framework can be applied directly to the downlink scenarios in which each BS serves one UE at a time. See Fig. 3.2 for an illustration of a 3-cell network with  $N = 2$  femtocells and  $M = 1$  macrocell. Each UE  $i$  chooses its transmit power  $p_i$  from a compact set  $\mathcal{P}_i \subseteq \mathbb{R}_+$ . We assume that  $0 \in \mathcal{P}_i, \forall i \in \{1, \dots, N+M\}$ , namely a UE can choose not to transmit. The joint power profile of all the UEs is denoted by  $\mathbf{p} = (p_1, \dots, p_{N+M}) \in \mathcal{P}$ , where  $\mathcal{P} = \prod_{i=1}^{N+M} \mathcal{P}_i$ . The power profile of all the UEs other than  $i$  is denoted by  $\mathbf{p}_{-i}$ . When a UE  $i$  chooses a transmit power  $p_i$ , the signal to interference and noise ratio (SINR) experienced at BS  $i$  is  $\gamma_i(\mathbf{p}) = \frac{g_{ii}p_i}{\sum_{j \neq i} g_{ji}p_j + \sigma_i^2}$ , where  $g_{ji}$  is the channel gain from UE  $j$  to BS  $i$ , and  $\sigma_i^2$  is the noise power at BS  $i$ . Since the BSs cannot cooperate to decode their messages, each BS  $i$  treats the interference as white noise, and gets the following throughput [BPB13a], [CAM09] at the power profile  $\mathbf{p}$ ,  $r_i(\mathbf{p}) = \log_2(1 + \gamma_i(\mathbf{p}))$ .

### 3.3.2 Interference Management Policies

The system is time slotted at  $t = 0, 1, 2, \dots$ , and the UEs are assumed to be synchronized as in [XS12], [EPT07], [WWL09]. At the beginning of time slot  $t$ , each UE  $i$  decides its transmit power  $p_i^t$  and obtains a throughput of  $r_i(\mathbf{p}^t)$ . Each UE  $i$ 's strategy, denoted by  $\pi_i : \mathbb{Z}_+ = \{0, 1, \dots\} \rightarrow \mathcal{P}_i$ , is a mapping from time  $t$  to a transmission power level  $p_i \in \mathcal{P}_i$ . The interference management policy is then the collection of all the UEs' strategies, denoted by  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{N+M})$ . Each UE is *delay sensitive* and hence discounts the future throughput as in [EPT07, XS12, WWL09, XS14]. The average discounted throughput for UE  $i$  is given as  $R_i(\boldsymbol{\pi}) = (1 - \delta) \sum_{t=0}^{\infty} \delta^t r_i(\mathbf{p}^t)$ , where  $\mathbf{p}^t = (\pi_1(t), \dots, \pi_{N+M}(t))$  is the power profile at time  $t$ , and  $\delta \in [0, 1]$  is the discount factor assumed to be the same for all the UEs as in [EPT07, WWL09, XS12, XS14]. We also assume the channel gain to be fixed over the considered time horizon as in [BBS06, ET90, JPP05, AAS11, LXH10, UAB11, LLJ10] [EPT07, WWL09, XS12, TFL11, XS14]. However, we will illustrate in Section 3.7.3 that the proposed framework can be adapted to the scenarios in which the channel conditions are time-varying.

An interference management policy  $\boldsymbol{\pi}^{const}$  is a policy based on constant power control [HYC09, GI10, CAM09, LLJ11, BPB13a, JMM09], if  $\boldsymbol{\pi}(t) = \mathbf{p}$  for all  $t$ . Write the joint throughput profile of all the UEs as  $\mathbf{r}(\mathbf{p}) = (r_1(\mathbf{p}), \dots, r_{N+M}(\mathbf{p}))$ . Then the set of all joint throughput profiles achievable by policies based on constant power control can be written as  $\mathcal{R}^{const} = \{\mathbf{r}(\mathbf{p}), \mathbf{p} \in \mathcal{P}\}$ . As we have discussed before, our proposed policy is based on MISs of the interference graph. The interference graph  $G$  has  $M + N$  vertices, which are the  $M + N$  UE-BS pairs. Each pair constitutes the BS and the UE it serves. There is an edge between two vertices if their cross interference is high. We will describe in detail how to construct the interference graph later. Given an interference graph, we write  $\mathbf{I}^G = \{I_1^G, \dots, I_{s(G)}^G\}$  as the set of all the MISs of the interference graph. Let  $\mathbf{p}^{I_j^G}$  be a power profile in which the UEs in the MIS  $I_j^G$  transmit at their maximum power levels, namely  $p_k = p_k^{max} \triangleq \max \mathcal{P}_k$  if  $k \in I_j^G$  and  $p_k = 0$  otherwise. Let  $\mathcal{P}^{MIS(G)} = \{\mathbf{p}^{I_1^G}, \dots, \mathbf{p}^{I_{s(G)}^G}\}$  be the set of all such power profiles. Then  $\boldsymbol{\pi}$  is a policy based on MIS if  $\boldsymbol{\pi}(t) \in \mathcal{P}^{MIS(G)}$  for all  $t$ . We denote the set of policies based

on MISs by  $\Pi^{MIS(G)} = \{\boldsymbol{\pi} : \mathbb{Z}_+ \rightarrow \mathcal{P}^{MIS(G)}\}$ . The set of joint *instantaneous* throughput profiles achievable by policies based on MIS is then  $\mathcal{R}^{MIS(G)} = \{\mathbf{r}(\mathbf{p}) : \mathbf{p} \in \mathcal{P}^{MIS(G)}\}$ . We will prove in Theorem 2 that the set of joint discounted throughput profiles achievable by policies based on MIS is  $\mathcal{V}^{MIS(G)} = \text{conv}\{\mathcal{R}^{MIS(G)}\}$ , where  $\text{conv}\{X\}$  representing the convex hull of set  $X$ .

### 3.4 Problem Formulation

In this section, we formalize the interference management policy design problem, and subsequently give a motivating example to highlight the advantages of the proposed policy over existing policies in solving this problem.

#### 3.4.1 Policy Design Problem

The designer of the network (e.g. the network operator) aims to design an optimal interference management policy  $\boldsymbol{\pi}$  that fulfills each UE  $i$ 's minimum throughput requirement  $R_i^{min}$  and optimizes a chosen network performance criterion  $W(R_1(\boldsymbol{\pi}), \dots, R_{N+M}(\boldsymbol{\pi}))$ . The network performance criterion  $W$  is an increasing function in each  $R_i$ . For instance,  $W$  can be the weighted sum of all the UEs' throughput, i.e.  $\sum_{i=1}^N w_i^{FUE} R_i(\boldsymbol{\pi}) + \sum_{j=1}^M w_j^{MUE} R_{N+j}(\boldsymbol{\pi})$  with  $\sum_{i=1}^N w_i^{FUE} + \sum_{j=1}^M w_j^{MUE} = 1$  and  $w_i^{MUE}, w_j^{FUE} \geq 0$ . We emphasize that the higher-priority of MUEs can be reflected by setting higher weights for the MUEs (i.e.  $w_i^{MUE} \geq w_j^{FUE}, \forall i = 1, \dots, N, \forall j = 1, \dots, M$ ), and by setting higher minimum throughput requirements for MUEs. Another example of performance criterion  $W$  is the max-min fairness (i.e. the worst UE's throughput), i.e.  $\min_i R_i(\boldsymbol{\pi})$ . The policy design problem is given as follows.

#### Design Problem

$$\begin{aligned} \max_{\boldsymbol{\pi}} \quad & W(R_1(\boldsymbol{\pi}), \dots, R_{N+M}(\boldsymbol{\pi})) \\ \text{s.t.} \quad & R_i(\boldsymbol{\pi}) \geq R_i^{min}, \forall i \in \{1, \dots, N+M\} \end{aligned} \tag{3.1}$$

The key steps and the challenges in solving the design problem are as follows: 1) How

to determine the set of achievable throughput profiles? Note that the set depends on the discount factor  $\delta$ . It is an open problem to determine the set of achievable throughput profiles, even for the special case of  $\delta = 0$  (i.e. the set of throughput profiles achievable by policies based on constant power control). 2) How to construct the optimal policy that achieves the optimal target throughput profile? The optimal policy again depends on  $\delta$ . It is much more challenging to determine the policy for delay-sensitive applications (i.e.  $\delta < 1$ ) than for delay-insensitive applications (i.e.  $\delta \rightarrow 1$ ), because the optimal policy is not cyclic. 3) How to construct a policy that requires minimum communication overhead among the UEs?

### 3.4.2 Motivating Example

We consider a network of 5 femtocells. On the left plot of Fig. 3.1, we have portrayed the interference graph of this network. Each vertex denotes a pair of FBS and its FUE. Each edge denotes strong local interference between the connected vertices (i.e. the distance between the FBSs is below some threshold). The interference graph is a pentagon, where each UE interferes only with two neighbors. We show the partitioning of the UEs by coloring the interference graph. There are three colors, and there is one color (i.e. black) to which only one UE belongs. On the right plot of Fig. 3.1, we show the 5 MIS's, each of which consists of two UEs. Note that the MIS are not disjoint. For illustrative purposes, suppose that the 5 femtocells and their UEs are symmetric, in the sense that all the UEs have maximum transmit power of 30mW, direct channel gain of 1, cross channel gain of 0.25 between the neighbors, noise power at the receiver of 2mW, minimum throughput requirement of 1.2 bits/s/Hz, and discount factor of 0.8 representing delay sensitivity. For simplicity, we set the cross channel gain between non-neighbors to be 0.

We compare our proposed policy against the following policies discussed in Section 3.2:

- Policies based on constant power control [HYC09, GI10, CAM09, LLJ11, BPB13a], in which each UE chooses a constant (time-invariant) power level all the time.
- Coloring-based TDMA policies [RL93, HS95, PST12], in which the UEs are partitioned

into mutually exclusive subsets by coloring the interference graph; in each time slot, all the UEs of one color are chosen to transmit. In this example, 3 colors are required and there exists a color to which only one UE belongs. Hence, the average number of active UEs in each time slot is less than 2. Note that the optimal performance of coloring based frequency reuse policies is the same as the optimal performance that can be attained by any coloring based TDMA of any arbitrary cycle length. This is due to the fact that FDM and TDM are equivalent provided the frequency/time can be divided arbitrarily.

- Cyclic MIS-based TDMA policies [RP89, CS89, BBS06, ET90, JPP05, AAS11], in which different MISs of UEs are scheduled in a cyclic manner. In this example, there are 5 MISs, each of which consists of 2 UEs. Hence, the average number of active UEs in each time slot is 2. This is the major reason why MIS-based TDMA policies are more efficient than coloring-based TDMA policies. To completely specify the policy we must also specify a cycle length and order of transmissions; note that the efficiency of the policy will depend on the cycle length due to delay sensitivity.

We illustrate the performance of the above policies vs the proposed policy in Table 3.1. The performance criterion is max-min fairness, i.e. we aim to maximize the worst UE's throughput. Constant policies are inefficient, because simultaneous transmission results in strong mutual interference. Coloring-based TDMA policies eliminate the interference but they do so inefficiently, because there are slots in which only one UE is transmitting; this is wasteful (the average number of UEs transmitting in each time slot is less than 2). MIS-based cyclic TDMA policies improve on the coloring-based schemes because 2 UEs transmit in every slot but they are still inefficient due to delay-sensitivity. The inefficiency of cyclic MIS-based policies for delay-sensitive applications comes from the fact that not all the transmission opportunities (TXOPs) (i.e. positions) in a cycle are created equal: the earlier TXOPs guarantee higher chances to deliver packets prior to their deadlines. The UEs that transmit in later TXOPs of a cycle suffer from delay.

Remarkably, the proposed policy is not only much more efficient than existing policies, it

Table 3.1: Comparisons against spatial reuse TDMA based policies.

Policies	Max-min throughput (bits/s/Hz)	Performance Gain %
Optimal constant power	1.32	21.2%
Optimal Coloring TDMA (arbitrary L)	1.33 (Upper Bound)	20.3 %
Optimal MIS TDMA (L=5)	1.36	17.6 %
Optimal MIS TDMA (L=7)	1.49	7.8 %
Optimal Proposed	1.60	–

is much easier to compute. To compare with constant policies, note simply that finding the optimal constant policy is NP-hard [TFL11] in general, because the optimization problem is non-convex due to the mutual interference. To compare with different classes of TDMA policies, note that for (coloring-based and MIS-based) cyclic TDMA policies, the complexity of finding the optimal cyclic policy of a given length grows exponentially with the cycle length (and exponentially with the number of MISs when the cycle length is large enough for reasonable performance). To get a hint of why this is so, note that in a cyclic policy, the UE’s performance is determined not only by the number of TXOPs in a cycle but also by the positions of the TXOPs since UEs are discounting their future utilities (due to delay sensitivity). Thus it is not only the length of the cycle that is important but also the ordering of transmissions within each cycle. For instance, for the 5-UE case above, achieving performance within 10% of the optimal proposed policy requires that the cycle length L be at least 7, and so requires searching among the thousands  $(16800)^4$ <sup>4</sup> of different nontrivial schedules (the schedules in which each UE transmits at least once in each cycle) of cycle length 7. Even this small problem is computationally intensive. For a moderate number of 10 femtocells, assuming a completely connected interference graph which has 10 MISs, and a cycle length of 20, we need to search more than ten billion (i.e.  $10^{10}$ ) non-trivial schedules – a completely intractable problem.

---

<sup>4</sup>We compute the number of nontrivial schedules by exhaustively searching among all the possible policies.

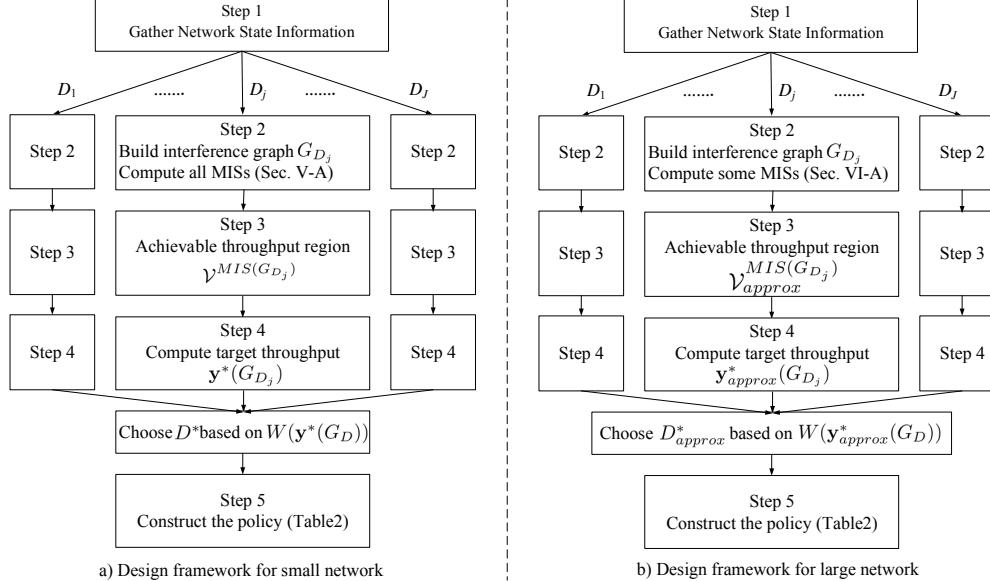


Figure 3.3: Steps in the design framework.

## 3.5 Design Framework

In this section, we develop a general design framework for solving the Design Problem. We will provide sufficient conditions under which our proposed framework is optimal, and demonstrate a wide variety of networks that fulfill the sufficient conditions.

### 3.5.1 Description of the Proposed Design Framework

The proposed methodology for solving the design problem consists of 5 steps which are illustrated in Fig. 3.3. We describe them in detail as follows.

#### 3.5.1.1 Step 1. The Designer Gathers Network Information

The designer is informed by each BS  $i$  of the minimum throughput requirement  $R_i^{min}$  of its UE, the channel gain from each UE  $j$  to its receiver  $g_{ji}$ , its UE's maximum transmit power level  $p_i^{max}$ , the noise power level at its receiver  $\sigma_i^2$ , and its location as in [LXH10], [UAB11], [LLJ10]. Such information is sent to the designer via the backhaul link. In some circumstances, the information about the location of FBSSs is available to the femtocell

gateways [LXH10], [LLJ10], who can send this information to the designer.

### 3.5.1.2 Step 2. The Designer Constructs the Interference Graph and Computes the MISs

The designer constructs the interference graph using the information of cell locations obtained in Step 1. Specifically, it uses a distance based threshold rule as in [HS95] [Hal80] to construct the graph: there is an edge between two cells if the distance between BSs in these two cells is smaller than a threshold  $D$ .<sup>5</sup> Given the threshold  $D$ , we denote the resulting graph by  $G_D$ , and the set of its MISs by  $\mathbf{I}^{G_D}$ , which can be calculated as in [JYP88]. We assume that the distance threshold  $D$  is fixed for now, and will discuss how to select the threshold in the next section.

### 3.5.1.3 Step 3. The Designer Characterizes Achievable the Throughput

Based on the MISs computed in Step 2, the designer identifies the set  $\mathcal{V}^{MIS(G_D)}(\delta)$  of throughput vectors achievable by MIS-based policies. Note that  $\mathcal{V}^{MIS(G_D)}(\delta)$  depends on the discount factor. Recall that  $\mathcal{R}^{MIS(G_D)} = \{\mathbf{r}(\mathbf{p}) : \mathbf{p} \in \mathcal{P}^{MIS(G_D)}\}$  is the set of *instantaneous* throughput profiles achievable by MIS-based policies in  $\Pi^{MIS(G_D)}$ . The theorem below proves that  $\mathcal{V}^{MIS(G_D)}(\delta)$  is a convex hull of  $\mathcal{R}^{MIS(G_D)}$ , i.e.  $\mathcal{V}^{MIS(G_D)}$  when the discount factor  $\delta \geq 1 - \frac{1}{s(G_D)}$ , where  $s(G_D)$  is the number of MISs in the interference graph  $G_D$ .

**Theorem 2** *Given the interference graph  $G_D$ , for any  $\delta \geq \bar{\delta} = 1 - \frac{1}{s(G_D)}$ , the set of throughput profiles achieved by MIS-based policies is  $\mathcal{V}^{MIS(G_D)}(\delta) = \mathcal{V}^{MIS(G_D)}$ .*

We provide the proof sketches here, while all the detailed proofs can be found in the Appendix Section at the end of this chapter.

**Proof Sketch 1** *The main step involved in proving the above is to derive the conditions on the discount factor such that each throughput vector in  $\mathcal{V}^{MIS(G_D)}$  can be decomposed into a*

---

<sup>5</sup>Note that the interference actually depends on the distance between a BS and a UE in another cell, instead of the distance between two BSs. When the distance from a BS to its UE is small, then the distance between BSs is an accurate representation of interference.

current throughput vector which belongs to  $\mathcal{R}^{MIS(G_D)}$  and a continuation throughput which belongs to  $\mathcal{V}^{MIS(G_D)}$ . To derive the conditions, we show that for any vector in  $\mathcal{V}^{MIS(G_D)}$  there exists at least one throughput vector in  $\mathcal{R}^{MIS(G_D)}$  to decompose the vector. Since the continuation throughput also belongs to  $\mathcal{V}^{MIS(G_D)}$ , it can be decomposed as well in a similar fashion. Hence, all the vectors in  $\mathcal{V}^{MIS(G_D)}$  are achievable.  $\blacksquare$

Theorem 2 is important because it analytically characterizes the set of throughput profiles achievable by MIS-based policies, and gives us the requirements that need to be fulfilled by the discount factor.

### 3.5.1.4 Step 4. The Designer Determines the Optimal Target Weights

Among all the achievable throughput profiles identified in Step 3, the designer selects the target throughput profile in order to optimize the network performance. Note that each UE  $i$ 's average throughput  $R_i$  can be expressed as a convex combination of the instantaneous throughput vectors achieved by MIS-based policies (i.e. the throughput vectors in  $\mathcal{R}^{MIS(G_D)}$ ). Thus determining the optimal target vector and its corresponding coefficients in the convex combination can be formulated as the following optimization problem:

$$\begin{aligned} & \max_{\mathbf{y}, \boldsymbol{\alpha}} W(y_1(G_D), \dots, y_{N+M}(G_D)) \\ & \text{s.t. } y_i(G_D) \geq R_i^{\min}, \forall i \in \{1, \dots, N+M\} \\ & y_i(G_D) = \sum_{j=1}^{s(G_D)} \alpha_j r_i(\mathbf{p}^{I_j^{G_D}}), \forall i \in \{1, \dots, N+M\} \\ & \sum_{j=1}^{s(G_D)} \alpha_j = 1, \alpha_j \geq 0, \forall j \in \{1, \dots, s(G_D)\} \end{aligned} \quad (3.2)$$

The above optimization problem is a convex optimization problem and is easy to solve if  $W$  is concave (e.g. weighted sum throughput or max-min fairness). The resulting optimal target vector and its corresponding coefficient is given as  $\mathbf{y}^*(G_D) = [y_1^*(G_D), \dots, y_{N+M}^*(G_D)]$  and  $\boldsymbol{\alpha}^*(G_D) = [\alpha_1^*(G_D), \dots, \alpha_{s(G_D)}^*(G_D)]$  respectively. Note that the optimal value depends on the interference graph  $G_D$  which we assume to be fixed in this section. The optimal

coefficient for the  $i^{th}$  MIS  $I_i^{G_D}$ , i.e.,  $\alpha_i^*(G_D)$  can be interpreted as the fraction of time for which  $I_i^{G_D}$  transmits.

### 3.5.1.5 Step 5. Each UE Implements the Policy Distributedly to Achieve the Target

The designer informs each UE  $i$  of the optimal coefficients, i.e.  $\boldsymbol{\alpha}^*(G_D)$  and the indices of MIs that UE  $i$  belongs to. The designer can send the above information to each BS  $i$ , who will forward the information to its UE. Each UE  $i$  executes the policy in Table 3.2. The policy in Table 3.2 leads to a non-stationary scheduling of the MIs. Note that each UE  $i$  computes its own policy online *without* information exchange. Hence, the computed policy is implemented in a decentralized manner by the UEs. Next we state the condition under which the policy indeed converges to the target vector  $\mathbf{y}^*(G_D)$ .

**Theorem 3** *For any  $\delta \geq \bar{\delta} = 1 - \frac{1}{s(G_D)}$ , the policy computed in Table 3.2 achieves the target throughput profile  $\mathbf{y}^*(G_D)$ .*

**Proof Sketch 2** *We show that when  $\delta \geq \bar{\delta} = 1 - \frac{1}{s(G_D)}$ , the policy developed in Table 3.2 ensures that the decomposition property given in Proof Sketch of Theorem 2 is satisfied in each time slot. This is used to show that the distance from the target,  $\mathbf{y}^*(G_D)$  strictly decreases in each time slot.* ■

We briefly discuss the intuition behind our proposed policy. We determine which MI to transmit based on a metric that can be interpreted as the “fraction of time slots allocated to an MI in the future”: the MI that has the maximum fraction of time slots in the future, i.e. the highest metric, will transmit at the current time slot. The metric is updated in each time slot as follows: the fraction of time slots for the MI who has just transmitted will decrease, and those of the other MIs will increase. Hence, the resulting schedule is non-stationary and does not necessarily follow a cyclic pattern.

### 3.5.2 Constructing Optimal Interference Graphs

In Step 2 of the design framework, we construct the interference graph by comparing the distances between two BSs with a threshold  $D$ . Here we show how to choose the optimal threshold  $D^*$  and hence the optimal interference graph  $G_{D^*}$ , based on which the proposed policy achieves the highest network performance achievable by any MIS based policy in  $\Pi^{MIS(G_D)}$ . Formally, the designer chooses the optimal threshold  $D^*$  that results in the optimal interference graph  $G_{D^*} = \arg \max_{G_D \in \mathcal{G}} W(\mathbf{y}^*(G_D))$ , where  $\mathcal{G}$  is the set of all possible interference graphs constructed based on the distance rule. The designer solves the above optimization problem by performing Steps 2-4 for each of the  $|\mathcal{G}| = J$  interference graphs as shown in Fig. 3.3 and chooses the optimal one. Note that the number  $|\mathcal{G}|$  of all such interference graphs is finite and upper bounded by  $\frac{(M+N) \cdot (M+N-1)}{2} + 1$ , because the number of different distances between BSs is finite and upper bounded by  $\frac{(M+N) \cdot (M+N-1)}{2} + 1$ . Note that the Steps 3-5 of our design framework can be used for any given interference graph, which is not necessarily constructed based on the distance based threshold rule. We assume a distance based threshold rule as a concrete example, in order to describe how to choose the optimal interference graph.

### 3.5.3 Optimality of the Proposed Design Framework

Our proposed design framework first constructs the interference graph based on the distances between BSs, and then schedules the MISs of the constructed interference graph. Then our proposed policy let the UEs in the scheduled MIS to transmit at their maximum power levels. To some extent, the interference graph is a binary quantization of the actual interference (i.e. “no interference” among non-neighbors and “strong interference” among neighbors). Hence, the performance of the proposed policy depends crucially on how close the interference graph is to the actual interference pattern. If we choose a smaller threshold  $D$ , the interference graph will have fewer edges, the non-neighboring UEs will have higher cross channel gains. Hence, the UEs in a MIS may experience high accumulative interference from the non-neighbors. If we choose a higher threshold  $D$ , the interference graph is more conservative

Table 3.2: The algorithm run by each UE  $i$ .

<b>Require:</b> Target weights $\alpha^*(G_D) = [\alpha_1^*(G_D), \dots, \alpha_{s(G_D)}^*(G_D)]$
<b>Initialization:</b> Sets $t = 0$ , $\alpha_j = \alpha_j^*(G_D)$ for all $j \in \{1, \dots, s(G_D)\}$ .
<b>repeat</b>
Finds the MIS with the maximum weight: $r^* = \arg \max_{j \in \{1, \dots, s(G_D)\}} \alpha_j$
<b>if</b> $i \in I_{r^*}^{G_D}$ <b>then</b>
Transmits at power level $p_i^t = p_i^{max}$
<b>end if</b>
Updates $\alpha_j$ for all $j \in \{1, \dots, s(G_D)\}$ as follows
$\alpha_{r^*} = \frac{\alpha_{r^*} - (1-\delta)}{\delta},$
$\alpha_j = \frac{\alpha_j}{\delta} \forall j \neq r^*$
$t \leftarrow t + 1$
<b>until</b> $\emptyset$

and will have more edges. Hence, some UEs outside a MIS may cause low interference and should be scheduled together with the UEs in the MIS. Our proposed policy will achieve performance close to optimal, if the interference graph is well constructed such that: 1) neighbors have strong interference, and 2) non-neighbors have weak interference. Next, we analytically quantify the above intuition and provide rigorous conditions for the optimality of the proposed design framework.

Let  $W^*$  denote the optimal network performance, namely the optimal value of the design problem (1) with the performance criterion being the weighted sum throughput. We give conditions under which the proposed policy can achieve within  $\epsilon$  of the optimal performance  $W^*$ . We first quantify strong local interference among neighbors as follows. Define  $r'_i(\mathbf{p}) = \log_2(1 + \frac{g_{ii}p_i}{\sum_{j \in \mathcal{N}_i(G_D)} g_{ji}p_j + \sigma_i^2})$ , where  $\mathcal{N}_i(G_D)$  is the set of neighbors of  $i$  in  $G_D$  and let  $\mathcal{R}_a^{const} = \{r'_i(\mathbf{p}), \mathbf{p} \in \mathcal{P}\}$ ,  $\mathcal{R}_a^{MIS(G_D)} = \{r'_i(\mathbf{p}), \mathbf{p} \in \mathcal{P}^{MIS(G_D)}\}$  and  $\mathcal{V}_a^{MIS(G_D)} = \text{conv}\{\mathcal{R}_a^{MIS(G_D)}\}$ . Note that  $r'_i(\mathbf{p})$  is not the actual throughput  $r_i(\mathbf{p})$ , because we do not count the interference from non-neighbors in  $r'_i(\mathbf{p})$ .

**Definition 1 (Strong Local Interference):** The interference graph  $G_D$  exhibits *Strong Local Interference* (SLI) if  $\mathcal{V}_a^{MIS(G_D)}$  dominates  $\mathcal{R}_a^{const}$ , in the sense that every throughput profile in  $\mathcal{R}_a^{const}$  is weakly Pareto dominated [MS06b] by a throughput profile in  $\mathcal{V}_a^{MIS(G_D)}$ .

Definition 1 states that for an interference graph with SLI, it is more efficient to use MIS-based policies than constant power control policies. Next, we quantify the weak interference among non-neighbors.

**Definition 2 (Weak Non-neighboring Interference):** The interference graph  $G_D$  has  $\epsilon - \text{Weak Non-neighboring Interference}$  ( $\epsilon$ -WNI) if each UE  $i$ 's maximum interference from its non-neighbors is below some threshold, namely  $Int_i^{max}(G_D) = \sum_{j \notin \mathcal{N}_i(G_D), j \neq i} g_{ji} p_j^{max} \leq (2^\epsilon - 1)\sigma_i^2$ ,  $\forall i \in \{1, \dots, N + M\}$ .

The two definitions above combined in a general sense ensure that there is negative externalities locally and no negative externalities from the rest. Hence, these two definitions combined are referred to as *strong negative local externalities*. Now we state Theorem 4 which uses the above two definitions to ensure optimality.

**Theorem 4** *If the constructed interference graph  $G_{D^*}$  exhibits SLI and  $\epsilon$ -WNI, then the proposed policy computed through Steps 1-5 of Section 3.5.1 achieves within  $\epsilon$  of the optimal network performance  $W^*$ .*

**Proof Sketch 3** *The set of throughput vectors achievable by any policy is  $\text{conv}\{\mathcal{R}^{const}\}$ . Denote the optimal throughput vector by  $\mathbf{v}^* \in \text{conv}\{\mathcal{R}^{const}\}$ , namely  $W(\mathbf{v}^*) = W^*$ . There must exist a vector  $\tilde{\mathbf{v}} \in \text{conv}\{\mathcal{R}_a^{const}\}$  such that  $\tilde{\mathbf{v}} \geq \mathbf{v}^*$ , because we do not count the interference from non-neighbors when we calculate  $r'_i(\mathbf{p}) \in \mathcal{R}_a^{const}$ . SLI indicates that there exists a vector  $\mathbf{v}' \in \mathcal{V}_a^{MIS(G_{D^*})}$  such that  $\mathbf{v}' \geq \tilde{\mathbf{v}} \geq \mathbf{v}^*$ . This condition implies that if hypothetically there was zero interference from non-neighbors, then MIS based policies will achieve the optimal throughput vector. However, since there is interference from non-neighbors, we use  $\epsilon$ -WNI to bound the loss in throughput caused by the interference from non-neighbors. Using  $\epsilon$ -WNI we can find a throughput profile  $\mathbf{v} \in \mathcal{V}^{MIS(G_{D^*})}$  which is within  $\epsilon$  from  $\mathbf{v}' \in \mathcal{V}_a^{MIS(G_{D^*})}$ . Hence, we have  $\mathbf{v}' \geq \mathbf{v} \geq \mathbf{v}' - \epsilon$  and  $v_i \geq R_i^{min} - \epsilon$ . This shows that we can achieve a throughput vector that is  $\epsilon$  close to the optimal one, i.e.  $\mathbf{v} \geq \mathbf{v}^* - \epsilon$ .* ■

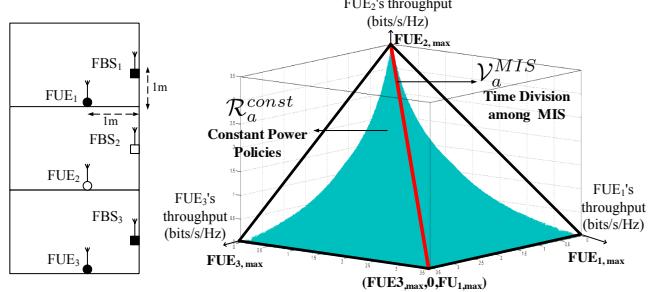


Figure 3.4: An example to illustrate the optimality of proposed framework.

*Example:* Consider 3 UEs and their corresponding FBS located on 3 different floors as shown in Fig. 3.4. Each UE can transmit at a maximum power of 100 mW. The channel model for determining the gain from a UE  $i$  to BS  $j$ , which includes the attenuation from the floor, is set based on [SR92]. Specifically, we have  $G_{ii} = 0.5$ ,  $G_{ji} = 0.25$  for  $|j - i| = 1$ ,  $G_{ji} = 0.0032$  for  $|j - i| = 2$ , and the noise power of 2 mW. We aim to maximize the average throughput while fulfilling a minimum throughput requirement of 1.2 bits/s/Hz for each FUE. Under three different thresholds  $D$ , we have the following three interference graphs (there are only three interference graphs because there are only three different values of distance between the BSs): 1) the triangle graph  $\{D \geq 4m\}$ , 2) the chain graph  $\{2m \leq D < 4m\}$  and 3) the edge-free graph  $\{0m \leq D < 2m\}$ . For each of these graphs, we apply the design framework described in Section 3.5.1 to obtain the corresponding policy, and achieve the following average throughput: 1) 1.56 bits/s/Hz 2) 2.7 bits/s/Hz and 3) 1.5 bits/s/Hz. Hence, the chain graph is the optimal choice among the three graphs. Also the chain graph exhibits SLI as illustrated in Fig. 3.4. and also exhibits  $\epsilon$ -WNI for  $\epsilon = 0.2$ . Hence, the proposed policy calculated based on the chain graph yields an average throughput within  $\epsilon = 0.2$  of the optimal solution  $W^*$  to the design problem in (1) (i.e.  $W^* \leq 2.9$  bits/s/Hz).

### 3.5.4 Complexity for Computing the Policy

We only compare the computational complexity of the proposed policies against cyclic MIS-based TDMA policies, since determining the optimal constant power based policy is a non-convex problem and has been shown to be NP-hard [TFL11]. We compare the two for a given

interference graph  $G_D$ . Both the optimal cyclic MIS TDMA and the proposed policy need to compute the set of MISs. Determining all the MISs is in general computationally expensive [JYP88]. However, the computational complexity is acceptable if the network is small, or if the number of MIS,  $s(G_D) = \mathcal{O}((N + M)^c)$ ,  $c > 1$  is bounded by a polynomial function in the number of vertices in  $G_D$ . We will develop an approximate algorithm to compute only a subset of MISs within polynomial time and with performance guarantees in Section 3.6. In our framework, the remaining amount of computation (other than computing MISs) is dominated by the amount of computation performed in Step 4, because in Step 5, the policy is computed online with a small amount  $\mathcal{O}(s(G_D))$  of computations per time slot. In Step 4, we solve the optimization problem in (5) with the objective function  $W$  and linear constraints. When  $W$  is linear (e.g. weighted sum throughput) or is the minimum throughput of any UE (in which case the problem can be transformed into a linear programming), the worst-case computational complexity for solving (5) is  $\mathcal{O}((s(G_D) + N + M)^{3.5} B^2)$  [Kar84] where  $B$  is the number of bits to encode a variable. In contrast, the complexity of computing the optimal cyclic MIS-based TDMA policy of cycle length  $L$  scales by  $[s(G_D)]^L$ . The complexity quickly becomes intractable when cycle lengths are moderately higher than  $N + M$ , which is usually needed for acceptable performance. In summary, the complexity of computing our policies is much lower than that of computing cyclic MIS-based TDMA policies.

### 3.5.5 Impact of the Density of Femtocells and Macrocells

The density of the network is defined as the average number of neighbors of a UE in the interference graph. To obtain sharp analytical results, we restrict our attention to a class of interference graphs with  $N + M$  vertices and  $H$  cliques of the same size. Note that a clique is a subset of vertices, where any two vertices are connected. Assuming that no two cliques are connected, we can compute the density as  $\frac{N+M}{H} - 1$ . When the total number  $N + M$  of UEs remains the same and the density increases, the number  $H$  of cliques will decrease. Since the vertices in a MIS can only come from different cliques, the number of MISs decreases as  $H$  decreases. As a result, the complexity of the policy will decrease. When the density increases, the multi-user interference increases, leading to a decrease in the throughput and

in the network performance.

## 3.6 Efficient Interference Management for Large-Scale Networks

### 3.6.1 Efficient Computation of a Subset of MISs

In our design framework proposed in Section 3.5, we require the designer to compute all the MISs in Step 2. However, computing all the MISs is computationally prohibitive for large networks. We propose an approximate algorithm to compute a subset of MISs for a given interference graph  $G_D$  in polynomial time and provide performance guarantees for our algorithm. Note that the graph  $G_D$  belongs to the class of unit-disk graphs [MBH95].

The subset of MISs are computed as follows.

i). *Approximate Vertex Coloring:* The designer first colors the vertices<sup>6</sup> of interference graph  $G_D$  using the approximate minimum vertex coloring scheme in [MBH95]. Let  $\mathcal{C}_1 = \{1, \dots, C(G_D)\}$  be the indices of the colors. It is proven in [MBH95] that the number of colors used is bounded by  $C^*(G_D) \leq C(G_D) \leq 3C^*(G_D)$  where  $C^*(G_D)$  is the minimum number of colors that can be used to color the vertices of  $G_D$ .

ii). *Generating MISs in a Greedy Manner:* The set of vertices with color  $i$  corresponds to an independent set  $I_i'^{G_D}$ . For each independent set  $I_i'^{G_D}$ , the designer adds vertices in a greedy fashion until the set is maximally independent. The procedure is described in Table 3.3. Let the output MIS obtained from Table 3.3 be  $I_{k(i)}^{G_D}$ , where  $k(i)$  is the index of the MIS in the original set of MISs  $\mathbf{I}^{G_D}$ . Hence, the set of ISs which are input to this step are  $\{I_1'^{G_D}, \dots, I_{C(G_D)}'^{G_D}\}$ . and the set of MISs that are output are  $\{I_{k(1)}^{G_D}, \dots, I_{k(C(G_D))}^{G_D}\}$ .

iii). *Generating the Approximate Maximum Weighted MIS:* Define a weight corresponding to each UE/vertex  $i$  as  $\bar{w}_i = r_i^{max}$ , where  $r_i^{max}$  is the maximum throughput achievable by UE  $i$  when all the other UEs do not transmit. Given these weights, the designer ideally will like to find the maximum weighted MIS, namely the MIS with the maximum sum weight of

---

<sup>6</sup>In minimum vertex coloring the objective is to use minimum number of colors and each vertex has to be assigned at least one color and no two neighbors are assigned the same color.

its vertices. However, finding the maximum weighted MIS is NP-hard [Rob86]. Hence, the designer will find the  $\eta$ -approximate maximum weighted MIS, denoted  $I_{k(C(G_D)+1)}^{G_D}$ , using the algorithm in [NHK05].

The set of MISs computed from the above steps is then  $\mathbf{I}_{approx}^{G_D} = \{I_{k(1)}^{G_D}, \dots, I_{k(C(G_D)+1)}^{G_D}\}$ . Note that  $\{I_{k(1)}^{G_D}, \dots, I_{k(C(G_D))}^{G_D}\}$  ensure that all the UEs are included in the scheduled MISs, and  $I_{k(C(G_D)+1)}^{G_D}$  is included for performance improvement. Given this subset of MISs, we can define  $\mathcal{P}_{approx}^{MIS(G_D)} = [\mathbf{p}^{I_{k(1)}^{G_D}}, \dots, \mathbf{p}^{I_{k(C(G_D)+1)}^{G_D}}]$ ,  $\mathcal{R}_{approx}^{MIS(G_D)} = \{\mathbf{r}(\mathbf{p}), \mathbf{p} \in \mathcal{P}_{approx}^{MIS(G_D)}\}$  and  $\mathcal{V}_{approx}^{MIS(G_D)} = \text{conv}\{\mathcal{R}_{approx}^{MIS(G_D)}\}$ . Let  $\Pi_{approx}(G_D) = \{\boldsymbol{\pi} : \mathbb{Z}_+ \rightarrow \mathcal{P}_{approx}^{MIS(G_D)}\}$  be the set of policies in which only the subset of MISs are scheduled. Steps 3,4 and 5 of the design framework in Section 3.5 are performed given this subset (See Fig. 3.3). The results of Theorem 2 and 3 still apply to the policies in  $\Pi_{approx}(G_D)$  and the set of achievable throughput profiles is  $\mathcal{V}_{approx}^{MIS(G_D)}$  given the  $\delta \geq 1 - \frac{1}{C(G_D)+1}$ . The target vector in  $\mathcal{V}_{approx}^{MIS(G_D)}$  and the corresponding coefficient is computed as in Step 4 of Section 3.5 and is denoted as  $\mathbf{y}_{approx}^*(G_D)$ ,  $\boldsymbol{\alpha}_{approx}^*(G_D)$  respectively. The coefficient vector  $\boldsymbol{\alpha}_{approx}^*(G_D)$  along with the indices of the MISs that UE  $i$  belongs to is transmitted to the BS  $i$  as in the Step 5 of Section 3.5.

The main intuition for the procedure developed above is as follows. Steps i) and ii) find MISs that contain all the UEs, and hence ensure that the minimum throughput requirements are satisfied. Step iii) finds the MIS that contains UEs with higher weights to optimize performance. Given the MISs obtained in Steps i)-iii) the Steps 3-5 of the design framework are performed.

### 3.6.2 Performance Guarantees for Large Networks

In this section, we consider the network performance criterion as the weighted sum throughput, and give performance guarantees for the policy when we compute the subset of MISs by Steps i)-iii) in the Section 3.6.1. Note that as we will show in the Section 3.7, the subset of MISs perform well in large networks for other network performance metrics as well. In particular the performance guarantee implies that the performance scales with the optimum  $W(\mathbf{y}^*(G_D))$  (the optimal network performance achieved by the policy proposed in Section

Table 3.3: Algorithm run by the designer.

<b>Require:</b> $V = 1, \dots, N + M$ set of vertices, $\bar{w}$ vector of weights of vertices, Independent set $I_i'^{G_D}$ and $\text{Adj}(I_i'^{G_D})$ where $\text{Adj}(X)$ is the set of neighbors of $X$
<b>Initialization:</b> $I_{k(i)}^{G_D} = I_i'^{G_D}$ , $\mathcal{N}'_i = V \cap (I_i'^{G_D} \cup \text{Adj}(I_i'^{G_D}))^c$ , here $(X)^c$ is the complement of $X$
While( $\mathcal{N}'_i \neq \phi$ ) $\mathcal{N}'_i = \text{sort}(\mathcal{N}'_i)$ , sort the vertices in $\mathcal{N}'_i$ in the decreasing order of the weights $\bar{w}_j$ $v' = \mathcal{N}'_{i,1}$ , here $\mathcal{N}'_{i,1}$ is the first vertex in $\mathcal{N}'_i$ $I_{k(i)}^{G_D} = I_{k(i)}^{G_D} \cup v'$ $\mathcal{N}'_i = \mathcal{N}'_i \cap (\{v'\} \cup \text{Adj}(\{v'\}))^c$ end

3.5.1) as the network size  $N + M$  increases. Define  $D_{ij}^{UE}$  as the distance from UE- $i$  to BS- $j$ . We make the following homogeneity assumption,  $p_i^{max} = p^{max}$ ,  $\sigma_i^2 = \sigma^2$ ,  $R_i^{min} = R^{min}$ ,  $\max_i D_{ii}^{UE} \leq \Delta$  and  $w_i = \frac{1}{N+M}$ .<sup>7</sup> Here  $\Delta$  is fixed and does not depend on the size of the network. We fix these parameters in order to understand the performance guarantee as a function of the network size. Let the channel gain  $g_{ij} = \frac{1}{(D_{ij}^{UE})^{np}}$ , where  $np$  is the path loss coefficient.

We choose the trade-off variables  $\rho, \zeta, \kappa$  that satisfy

$$\rho + 1 < \min\left\{\frac{\log_2\left(1 + \frac{p^{max}}{\Delta^{np}2\zeta\sigma^2}\right)}{3R^{min}}, \frac{\kappa}{\zeta(1+\eta)} \log_2\left(1 + \frac{p^{max}}{\Delta^{np}\sigma^2}\right)\right\}$$

and  $0 < \kappa < 1$ . Any eligible triplet  $\rho, \zeta, \kappa$  will define a class of interference graphs that exhibit  $\zeta$ -WNI and have maximum degrees upper bounded by  $\rho$ . Note that such interference graphs can have arbitrarily large sizes (see the example at the end of this section). Then the following theorem provides performance guarantees for the policy described in Section 3.6.1 for this class of interference graphs.

---

<sup>7</sup>We can extend our result to a heterogeneous network with  $p_i^{max} \geq p^{max}$ ,  $\sigma_i^2 \leq \sigma^2$ ,  $R_i^{min} \leq R^{min}$ ,  $\max_i D_{ii}^{UE} \leq \Delta$  and  $w_i \geq \frac{c}{N+M}$  with  $c$  as a constant. But we do not show this general result to avoid overly complicated notations.

**Theorem 5** For any interference graph that has a maximum degree no larger than  $\rho$  and exhibits  $\zeta$ -WNI with  $\rho + 1 < \min\{\frac{\log_2(1+\frac{p^{max}}{\Delta^{np}2\zeta\sigma^2})}{3R^{min}}, \frac{\kappa}{\zeta(1+\eta)} \log_2(1 + \frac{p^{max}}{\Delta^{np}\sigma^2})\}$  the policy in Section 3.6.1 achieves a performance  $\mathbf{W}(\mathbf{y}_{approx}^*(G_D))$  with a guarantee that  $\mathbf{W}(\mathbf{y}_{approx}^*(G_D)) \geq \frac{(1-\gamma)(1-\kappa)}{(1+\eta)} \cdot \mathbf{W}(\mathbf{y}^*(G_D))$ , where  $\gamma = (3(\rho+1)) \frac{R^{min}}{\log_2(1+\frac{p^{max}}{\Delta^{np}2\zeta\sigma^2})}$ .

**Proof Sketch 4** The condition that the graph does not have a degree more than the given threshold and the  $\zeta$ -WNI condition ensure that the algorithm proposed in Section 3.6.1 yields a feasible solution satisfying each UE's minimum throughput constraint. Also, it is shown that the minimum coefficient/fraction of time allocated to  $I_{k(C(G_D)+1)}^{G_D}$  is  $\geq (1 - \gamma)$ . Then it is shown that if UEs in  $I_{k(C(G_D)+1)}^{G_D}$  were to transmit all the time then the competitive ratio achieved is no smaller than  $\frac{1-\kappa}{(1+\eta)}$ . This combined with minimum coefficient of  $I_{k(C(G_D)+1)}^{G_D}$  leads to the competitive ratio guarantee of no less than  $\frac{(1-\gamma)(1-\kappa)}{(1+\eta)}$ . ■

The trade-off variables  $\rho, \zeta, \kappa$  as their name suggests provide trade-offs between how large is the class of interference graphs for which we can provide performance guarantees, and how good are the competitive ratio guarantees. On one hand, a higher  $\kappa$  allows higher  $\rho$ , and higher  $\rho$  and  $\zeta$  allow a larger class of graphs. On the other hand, as we can see from Theorem 5, higher  $\rho$  and  $\zeta$ , provided that they are eligible (higher  $\zeta$  decrease the maximum eligible  $\rho$ ), result in higher  $\gamma$ , and higher  $\gamma$  and  $\kappa$  give lower competitive ratio guarantees. Hence, we can tune the design parameters to provide different levels of competitive ratio guarantees for different classes of interference graphs.

Next, we give an example to illustrate Theorem 5.

Example: Consider a layout of FBSs in a  $K \times K$  square grid, i.e.  $K^2$  FBSs with a distance of 5m between the nearest FBSs, and assume that each FUE is located vertically below its FBS at a distance of 1 m. Fix the parameters  $p^{max} = 100$  mW,  $\sigma^2 = 3$  mW,  $R^{min} = 0.1$  bits/s/Hz,  $\eta = 0.1$ ,  $np = 4$  and the threshold  $D = 7$  m, which gives us the upper bound  $\rho = 4$  on the maximum degrees. We can also verify that the interference graphs under any number  $K^2$  of FBSs exhibit  $\zeta$ -WNI with  $\zeta = 0.15$ . Given  $\rho = 4$  and  $\zeta = 0.15$ , we choose the minimum  $\kappa = 0.17$ , which provides the highest competitive ratio guarantee of 0.53. This performance guarantee holds for any interference graph of any size  $K$ .

We now discuss the low complexity construction of efficient interference graph for large networks, which is useful especially when the procedure proposed in Section 3.5.2 is computationally prohibitive. In this case, the designer computes the subset of MISs as described in Section 3.6.1 and compares the optimal solution obtained to decide the best distance threshold for computing the policy. Formally stated, the designer computes  $G_{D_{approx}^*} = \arg \max_{G \in \mathcal{G}} W(\mathbf{y}_{approx}^*(G))$ . See Fig. 3.3 for a comparison of the design framework in Section 3.6.1 for large networks with that in Section 3.5.1 for small networks.

### 3.6.3 Complexity for Computing the Subset of MISs

We show that the proposed approximation method for computing the subset of MISs described in Section 3.6.1 has a complexity bounded by a polynomial in the number of vertices, i.e.,  $\mathcal{O}((N + M)^c)$ ,  $c > 1$ . This is because Steps i) and iii) use the algorithms developed in [MBH95] and [NHK05] for which the complexity has been proven to be polynomial and Step ii) uses a greedy strategy in which there can be a maximum of  $N + M$  iterations since at least one vertex is always removed from  $\mathcal{N}'_i$  in each iteration. The worst possible number of computations in an iteration is bounded by  $(N + M)^2$ . Hence, the upper bound of the complexity of Step ii) is  $\mathcal{O}((N + M)^3)$ . Hence, the subsets of the MISs can be computed within polynomial time, and the policy computed using this subset can guarantee a constant competitive ratio as shown in Section 3.6.2.

### 3.6.4 Extensions

#### 3.6.4.1 Construction of Interference Graphs Based on Other Rules

Our design frameworks in Section 3.5 and Section 3.6 do not rely on a specific method for constructing the interference graph. In Step 2 of the design frameworks (i.e., the step in which the interference graph is constructed), we can replace our distance-based construction of the interference graph with construction based on other criteria, such as SINR, interference levels [UAB11], etc. Then we can use the resulting interference graph as the input to Step 3. For construction rules based on other criteria, we can also use the procedure described

in Section 3.5.2 to optimize the construction rule (e.g., to choose the optimal threshold of SINR or the interference level, above which an edge is drawn between two nodes).

Note that in the design framework in Section 3.6, we find a subset of MISs, instead of all the MISs, because the network is large. To find this subset, we use the coloring algorithm in [NHK05], which is known to have polynomial-time complexity for unit-disk graphs. This is where we used the fact that the interference graph is constructed based on distances (such that the resulting graph is a unit-disk graph). However, we can use other polynomial-time coloring algorithms if the interference graph is generated based on other criteria. We can use a standard greedy coloring algorithm as in [Erc13]. In the next step we extend the ISs obtained by coloring to MISs. We can do this based on Step ii) in Section 3.6.1. The target weights and the corresponding schedule for these MISs can be generated based on Section 3.6.1. Results about the performance guarantees in terms of competitive ratio (See Theorem 5) can also be extended to this case.

#### 3.6.4.2 Incorporating Uncertainty in Channel Gains

Our design frameworks in Section 3.5 and Section 3.6 can be extended to the deployment scenarios in which the channel gains are not static. For fast fading, we can replace the instantaneous throughput with the expected instantaneous throughput in our design frameworks. For slow fading, we can track the fading by regularly re-computing the policy. Re-computing the entire policy every time may be costly. In Section 3.7.3 we show that the designer does not need to re-compute the entire policy to get considerable gains compared to the state-of-the-art. Specifically, the designer fixes the interference graph that is selected in the beginning, and only re-computes the target weights rather than re-compute the optimal interference graph and the corresponding target weights. We also show that the performance loss incurred with respect to the latter approach, which is based on an entire re-computation is limited (8%).

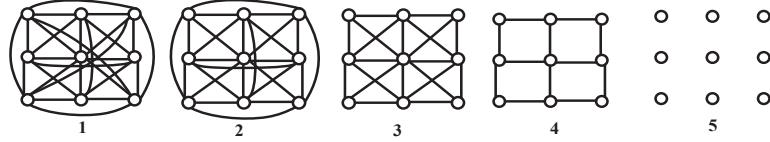


Figure 3.5: Different interference graphs for the  $3 \times 3$  BS grid.

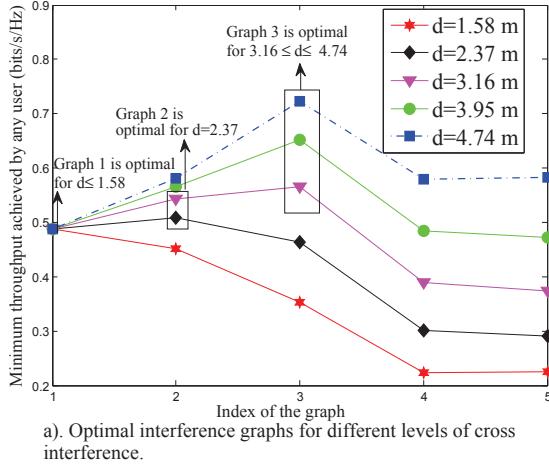
### 3.6.4.3 Incorporating Beamforming

We focus on the case where each UE has one antenna. When UEs have multiple antennas, we can easily incorporate beamforming in our framework. Beamforming mitigates the interference among the UEs served by the same BS. Hence, we can remove the edges between UEs in the same cell from the interference graph. Then we can use the new interference graph as the input to Step 3 of our design framework.

## 3.7 Illustrative Results

In this section, we show via simulations that our proposed policy significantly outperforms existing interference management policies under different performance criteria. These performance gains are obtained under varying interference levels for both small and large networks. We also evaluate the proposed policy when the channel conditions are time-varying due to fading. In this case, the designer ideally needs to recompute the optimal interference graph each time the channels change at the cost of a higher complexity. We show the robustness of the proposed policy when we choose a fixed interference graph regardless of the time-varying fading.

In each setting, we compare with the state-of-the-art policies described in Section 3.2, namely the constant power control based policies and the cyclic MIS TDMA based policies. We do not compare with coloring based TDMA/Frequency reuse policies as it was already shown in Section 3.4.2 that the MIS based TDMA policies will always lead to better network performance. Throughout this section, we will set the discount factor as the minimum one required when we use our original design framework in Section 3.5 (namely  $\delta = 1 - \frac{1}{s(G_D)}$  according to Theorems 2, 3), and the minimum one required when we use the approximate



a). Optimal interference graphs for different levels of cross interference.

Figure 3.6: Optimal interference graph selection for  $3 \times 3$  grid.

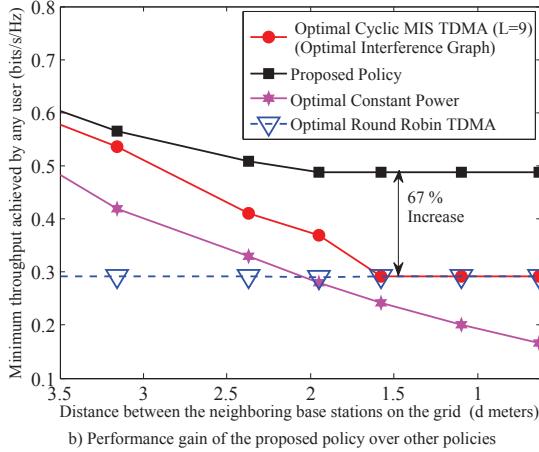


Figure 3.7: Performance comparison of the proposed policy for different size of the grid.

design framework for large networks in Section 3.6 (namely  $\delta = 1 - \frac{1}{C(G_D)+1}$ ). In this way, we evaluate the performance of our proposed policies under the most delay-sensitive applications.

### 3.7.1 Performance Gains Under Varying Interference Levels

Consider a  $3 \times 3$  square grid of 9 BSs (see Fig. 3.5) and corresponding UEs with the minimum distance between any two BSs given as  $d$ . Each UE  $i$  has  $\delta = 0.89$  and a maximum power of 200 mW and the noise power at the base station is 1 mW. Assume that the UEs and the

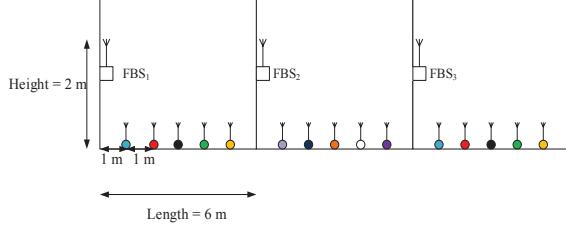


Figure 3.8: Illustration of the setup with 3 rooms.

BSs are in two parallel horizontal hyperplanes separated by a distance 3.16m. Each BS is vertically above its UE with a distance of 3.16m. Then the distance from UE  $i$  to another BS  $j$  is  $D_{ij}^{UE} = \sqrt{3.16^2 + (D_{ij}^{BS})^2}$ , where  $D_{ij}^{BS}$  is the distance between BSs  $i$  and  $j$ . The channel gain from UE  $i$  to BS  $j$  is  $g_{ij} = \frac{1}{(D_{ij}^{UE})^2}$ . The performance criterion is the max-min fairness. Under different thresholds  $D$  chosen by the designer, there are 5 possible topologies of the interference graph, as shown in the Fig. 3.5. For each grid size  $d$ , the optimal solution to (2) is computed for each interference graph as described in Section 3.5.2. Fig. 3.6. shows that under different grid sizes (i.e. different interference levels), the optimal interference graph (i.e. the optimal threshold) changes. As the interference level increases, the corresponding optimal interference graph has more edges. Fig 3.7 compare the performances of different policies under different grid sizes  $d$  (i.e. different interference levels). For a fixed grid size the optimal interference graph (computed as discussed above) is used as the input to each policy that we compare with. We can see that the proposed policy achieves up to 67% performance gain over the second best policy. Through the above results, we see that 1) it is important to construct different interference graphs based on the interference level, and 2) the proposed non-stationary schedule of MISs outperform the cyclic schedules.

### 3.7.2 Performance Scaling in Large Networks

We study a dense deployment scenario to evaluate the performance gain of our proposed scheme over the state-of-the-art. We allow more than one UE to transmit to a single BS, and will increase the number of UEs associated with a BS. Consider the uplink of a femtocell network in a building with 12 rooms adjacent to each other. Fig. 3.8 illustrates 3 of the 12

rooms with 5 UEs in each room. For simplicity, we consider a 2-dimensional geometry, in which the rooms and the FUEs are located on a line. Each room has a length of 6 meters. In each room, there are  $P$  uniformly spaced FUEs, and one FBS installed on the left wall of the room at a height of 2m. The distance from the left wall to the first FUE, as well as the distance between two adjacent FUEs in a room, is  $\frac{6}{(1+P)}$ m. Based on the path loss model in [SR92], the channel gain from each FBS  $i$  to a FUE  $j$  is  $\frac{1}{(D_{ij}^{UE})^2 \Delta^{n_{ij}}}$ , where  $\Delta = 10^{0.25}$  is the coefficient representing the loss from the wall, and  $n_{ij}$  is the number of walls between FUE  $i$  and FBS  $j$ . Each UE has a maximum transmit power level of 1000 mW and a minimum throughput requirement of  $R_i^{min} = 0.05$  bits/s/Hz. The noise power at the base station is 1 mW. For each  $P$ , the designer chooses the optimal threshold to construct the optimal interference graph. Note that the UEs in the same room accessing the same BSs are all connected to each other in the interference graph, since the distances between their receiving BSs is 0.

We vary the number  $P$  of FUEs in each room from 5 to 15. We fix the  $\delta = 0.97$ , i.e. the least value it can take based on the largest number of UEs per room, i.e.  $P = 15$ . For each  $P$ , the designer constructs the optimal graph  $G$  as described in Section 3.6.1 using the low complexity method as the number of UEs is large. Under all the considered values of  $P$ , the optimal interference graph connects all the UEs in adjacent rooms with edges and does not connect the UEs in non-adjacent rooms. We use the same optimal graph to compute the optimal cyclic MIS TDMA of cycle length  $L$ . The cycle length is varied from 12 to 58 depending upon the number of UEs (we try to choose as large cycle lengths as possible to maximize performance within a feasible computational complexity). The number of non-trivial cyclic policies under different  $P$  may vary from  $10^8$  to even more than  $10^{50}$  which renders exhaustive search to be intractable. Hence, for each  $P$  we do a randomized search in 4 million policies to search for the optimal one. Fig. 3.9 compares the performance of different policies in terms of both the max-min fairness and the sum throughput. The constant power policy cannot satisfy the feasibility conditions for any number of UEs in each room. The performance gain over cyclic MIS TDMA policies increases as the network becomes larger. When there are 15 UEs in each room, we can improve the worst UE's

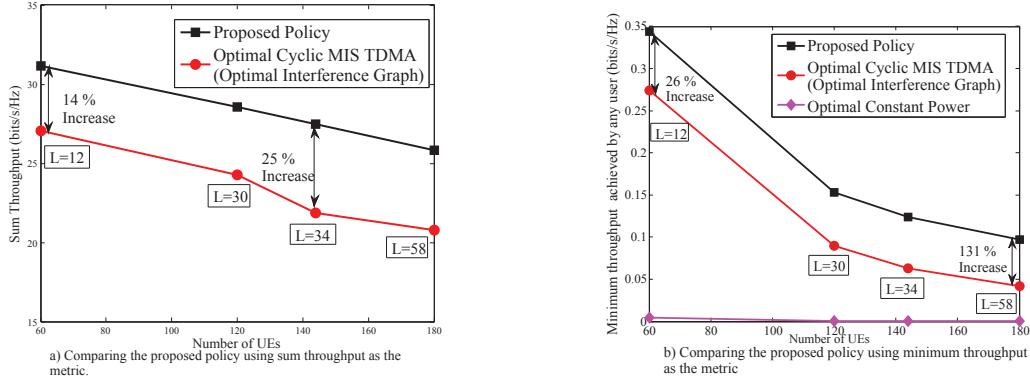


Figure 3.9: Comparing the proposed policy against others for a) sum throughput as the metric, b) minimum throughput as the metric.

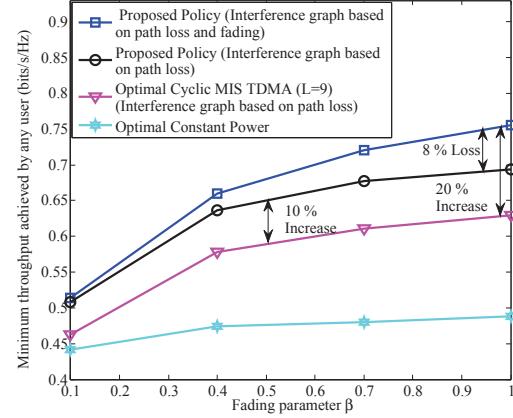


Figure 3.10: Illustrating the robustness of interference graph selection based on path loss.

throughput by 131% compared to cyclic MIS TDMA policies.

### 3.7.3 Performance under Dynamic Channel Conditions

We consider a 9-cell network with a grid size  $d = 4.74$  m, where each BS is vertically above its UE at a distance of 3.16m as in Section 3.7.1. Each UE has a maximum power level of 1000 mW,  $\delta = 0.89$  and the noise power at the base station is 1 mW. The channel gain is the product of path loss as in Section 3.7.1 and fading component,  $f_{ij} \sim Rayleigh(\beta)$ . Here, we assume that the fading component changes every 50 time slots independently and the new channel conditions are reported to the designer by each FBS as in Step-1 in Section 3.5.1. The designer has the choice of recomputing the optimal interference graph and thereby the

optimal target every 50 time slots at the cost of a higher complexity, or choosing a fixed optimal interference graph based on the channel gains computed from the path loss model (which will be graph 3 in Fig. 3.5) and selecting the optimal target every 50 time slots based on it. In Fig. 3.10 we compare the loss due to choosing a fixed interference graph with choosing the optimal interference graph every 50 time slots. We average the performance for a duration over a total of 10000 time slots for a fixed  $\beta$ . In Fig. 3.10, we see that for a low  $\beta$ , i.e.  $\beta = 0.1$  which implies a lower variance in fading the loss is only 1% and even when  $\beta$  is large, i.e.  $\beta = 1$  then as well the loss is 8%. We also compare with Cyclic MIS TDMA, cycle length  $L = 9$  and optimal constant power policy, the performance gain with the proposed policy using a fixed interference graph is consistently 10 % for varying fading conditions, while choosing the optimal interference graph leads to a maximum gain of 20%.

### 3.8 Discussion on the Generality of the Framework

The framework described in the previous sections focused on the application of interference management in wireless networks. In the framework that we described, each UE's utility for one time slot  $t$  was defined as the Shannon capacity  $r_i(\mathbf{p}^t)$ . In this section, we make no restrictions on the functional form of  $r_i(\mathbf{p}^t)$ . For a general framework, we consider a problem with  $N$  users and each user's utility for taking an action  $p_i^t \in \mathcal{P}_i$  in time slot  $t$  is defined as  $r_i(\mathbf{p}^t)$  and the long-term utility is defined as  $R_i(\boldsymbol{\pi})$ , where  $\boldsymbol{\pi}$  is the joint resource sharing policy. We define a general design problem as follows.

#### General Design Problem (GDP)

$$\begin{aligned} \max_{\boldsymbol{\pi}} \quad & W(R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi})) \\ \text{subject to} \quad & R_i(\boldsymbol{\pi}) \geq R_i^{\min}, \forall i \in \{1, \dots, N\} \end{aligned}$$

There are different problems where resource sharing is useful such as task scheduling when different tasks compete for limited computational resources. Next, we describe the design framework for the above GDP. We comment on how different steps of the design framework can be adapted to this setting. In Step 1 and 2 the designer constructs the interference graph

and computes the MISs. In this case as well the designer can construct the interference graph based on the externalities between the different users as follows. The interfering neighbors of a user  $i$  depends on the utility function for each user  $i$ . In the design framework described earlier, the identification was carried out by a centralized entity using certain protocols. In a general case, we can analyze the utility functions  $r_i(\mathbf{p})$  of all the pairs of users. If the worst case utility of a user when the other user is using the resource to the maximum is below a certain threshold, then we assume that there is an edge between the two users in the interference graph. In Theorem 2, we characterized the set of achievable rates. The same theorem also directly carries over to this case and we can characterize the set of achievable payoffs in the same way (because the proof of the previous theorem did not rely on the functional form of  $r_i(\mathbf{p})$ ). We follow the exact same steps as in Step 4 and Step 5 for the general problem as well. In Theorem 4, the optimality of the proposed scheme was shown. We adapt the definition of weak non-neighboring interference to the general case of as follows. If the the maximum difference between the utility achieved any user when all the non-neighbors use the resource to the maximum extent possible and when all the non-neighbors do not use the resource at all is bounded by a small value  $\epsilon$ , then the system exhibits weak non-neighboring interference. We can invoke the assumption of SLI directly and do not need to change it. As long as this assumption and SLI holds, the Theorem 4 also holds in this more general case.

### 3.9 Conclusion

In this chapter, we proposed a novel and systematic method for centralized resource sharing. We mainly focused on the problem of interference management but we showed that the framework's results are general and extend to other scenarios. The proposed framework relies on constructing optimal interference graphs and optimally scheduling the MIS of the constructed graph to maximize the network performance given the minimum throughput requirements. Importantly, the proposed policy is non-stationary and can address the requirements of delay sensitive users. We prove the optimality of the proposed policies under

various deployment scenarios. The proposed policy can be implemented in a decentralized manner with low overhead of information exchange between BSs and UEs. For large networks, we develop a low-complexity design framework that is provably efficient. Our proposed policies achieve significant (up to 130 %) performance improvement over existing policies, especially for dense and large-scale deployments of femtocells.

### 3.10 Appendix

We would begin by defining a self-generating set similar to what is defined in repeated game theory [MS06a]. However, our definition is less restrictive since it does not involve incentive compatibility as required in strategic user setting in repeated games.

**Definition 1. Decomposability:** A throughput vector  $\mathbf{v} \in \mathbb{R}^{N+M}$  is decomposable on a set  $\mathcal{W}$  with respect to a discount factor  $\delta$ , if there exists a power profile  $\mathbf{p} \in \mathcal{P}^{MIS(G_D)}$  and a mapping  $\gamma : \mathcal{P}^{MIS(G_D)} \rightarrow \mathcal{W}$  such that  $\forall i \in \{1, \dots, N+M\}$

$$v_i = (1 - \delta)r_i(\mathbf{p}) + \delta\gamma_i(\mathbf{p}) \quad (3.3)$$

Define  $\mathcal{D}(\mathcal{W}, \delta) = \{\mathbf{v} : \mathbf{v} \text{ is decomposable on } \mathcal{W}\}$ .

**Definition 2. Self-Generation:**  $\mathcal{W}$  is self-generating with respect to discount factor  $\delta$ , if each throughput vector  $\mathbf{v} \in \mathcal{W}$  is decomposable on  $\mathcal{W}$ , thus  $\mathcal{W} \subseteq \mathcal{D}(\mathcal{W}, \delta)$

#### 3.10.1 Appendix A

**Proof of Theorem 2.**  $\mathcal{V}^{MIS(G_D)} = \text{conv}\{\mathcal{R}^{MIS(G_D)}\}$ , where  $\mathcal{R}^{MIS(G_D)} = \{\mathbf{r}(\mathbf{p}^{I_j^{G_D}}), \forall j \in \{1, \dots, s(G_D)\}\}$  and  $\text{conv}\{X\}$  is the convex hull of the set  $X$ . We will show that  $\mathcal{V}^{MIS(G_D)}$  is self-generating for  $\delta \geq \bar{\delta}$ . If we can show that the set is self-generating then we can construct an MIS-based policy which achieves any given target vector in that set. This is explained as follows. Let us assume that  $\mathcal{V}^{MIS(G_D)}$  is self-generating with respect to certain  $\delta$ , then given a vector  $\mathbf{v} \in \mathcal{V}^{MIS(G_D)}$  it can be decomposed as,  $\mathbf{v} = (1 - \delta)\mathbf{r}(\mathbf{p}^{I_j^{G_D}}) + \delta\gamma(\mathbf{p}^{I_j^{G_D}})$ . The vector  $\gamma(\mathbf{p}^{I_j^{G_D}})$  obtained on decomposition is treated as the target vector for the transmissions starting the next period. We know that  $\gamma(\mathbf{p}^{I_j^{G_D}}) \in \mathcal{V}^{MIS(G_D)}$  and  $\mathcal{V}^{MIS(G_D)}$  is assumed to

be self-generating, which means  $\gamma(\mathbf{p}^{I_j^{G_D}})$  can also be decomposed in the same manner by a certain MIS based power profile  $\mathbf{p}^{I_j^{G_D}} \in \mathcal{R}^{MIS(G_D)}$ . This step can be recursively followed for all the future periods to generate a policy and hence, the target  $\mathbf{v}$  is achieved. Next, we show the conditions under which the set  $\mathcal{V}^{MIS(G_D)}$  is indeed self-generating.

Let  $\mathbf{v} \in \mathcal{V}^{MIS(G_D)}$  and we can express

$$\mathbf{v} = \sum_{k=1}^{s(G_D)} \alpha_k \mathbf{r}(\mathbf{p}^{I_k^{G_D}}), \quad \alpha_k \geq 0, \quad \forall k \in \{1, \dots, s(G_D)\}$$

and  $\sum_{k=1}^{s(G_D)} \alpha_k = 1$ . If  $\mathcal{V}^{MIS(G_D)}$  is self-generating with respect to certain discount factor  $\delta$  then  $\exists \mathbf{p}^{I_j^{G_D}}$  and  $\gamma(\mathbf{p}^{I_j}) \in \mathcal{V}^{MIS(G_D)}$  about which  $\mathbf{v}$  can be decomposed as follows

$$\mathbf{v} = (1 - \delta) \mathbf{r}(\mathbf{p}^{I_j^{G_D}}) + \delta \gamma(\mathbf{p}^{I_j^{G_D}}) \quad (3.4)$$

Next, we come up with sufficient conditions for  $\gamma(\mathbf{p}^{I_j^{G_D}}) \in \mathcal{V}^{MIS(G_D)}$

$$\begin{aligned} \gamma(\mathbf{p}^{I_j^{G_D}}) &= \frac{\mathbf{v} - (1 - \delta) \mathbf{r}(\mathbf{p}^{I_j^{G_D}})}{\delta} \\ &= \sum_{k=1, k \neq j}^s \frac{\alpha_k}{\delta} \mathbf{r}(\mathbf{p}^{I_k^{G_D}}) + \frac{\alpha_j - (1 - \delta)}{\delta} \mathbf{r}(\mathbf{p}^{I_j^{G_D}}) \end{aligned}$$

Observe that  $\sum_{k=1}^s \frac{\alpha_k}{\delta} + \frac{\alpha_j - (1 - \delta)}{\delta} = 1$ . If  $0 \leq \frac{\alpha_k}{\delta} \leq 1, \forall k \in \{1, \dots, s(G_D)\}, k \neq j$  and  $0 \leq \frac{\alpha_j - (1 - \delta)}{\delta} \leq 1$  then  $\gamma(\mathbf{p}^{I_j^{G_D}}) \in \mathcal{V}^{MIS(G_D)}$ . This can be combined into one condition on  $\delta$  given as

$$\delta \geq \left\{ \max_{k \neq j} \{\alpha_k\}, 1 - \alpha_j \right\}$$

Note that decomposition condition requires the existence of at least one profile  $\mathbf{p}^{I_j^{G_D}}$ . This means we can choose the least possible bound on  $\delta$ , which is sufficient to ensure that there will exist at least one profile  $\mathbf{p}^{I_j^{G_D}}$  for decomposition.

$$\begin{aligned} \delta &\geq \min_{j \in \{1, \dots, s(G_D)\}} \left\{ \max_{k \neq j} \{\alpha_k\}, 1 - \alpha_j \right\}, \\ &= 1 - \alpha_{[s(G_D)]}, \end{aligned}$$

where  $\alpha_{[s(G_D)]} = \max_{j \in \{1, \dots, s(G_D)\}} \{\alpha_1, \dots, \alpha_{s(G_D)}\}$ . Also,  $\sum_{k=1}^{s(G_D)} \alpha_k = 1, \alpha_k \geq 0, \forall k \in \{1, \dots, s(G_D)\}$  yields that  $\alpha_{[s(G_D)]} \geq \frac{1}{s(G_D)}$ . Hence, the condition  $\delta \geq 1 - \frac{1}{s(G_D)}$  is sufficient to

ensure decomposition of every vector in the convex hull. Thus  $\mathcal{V}^{MIS(G_D)}$  is self-generating for all the discount factors  $\delta \geq \bar{\delta} = 1 - \frac{1}{s(G_D)}$ . Hence, we have been able to show that for a discount factor  $\delta \geq 1 - \frac{1}{s(G_D)}$  each vector in  $\mathcal{V}^{MIS(G_D)}$  can be achieved by an MIS based policy, i.e.  $\mathcal{V}^{MIS(G_D)} \subseteq \mathcal{V}^{MIS(G_D)}(\delta)$ . The throughput achieved by any MIS based policy is  $\mathbf{R}(\boldsymbol{\pi}) = (1 - \delta) \sum_{t=0}^{\infty} \delta^t \mathbf{r}(\mathbf{p}^t)$ , here  $\mathbf{r}(\mathbf{p}^t) \in \mathcal{R}^{MIS(G_D)}$ . Since the coefficients  $(1 - \delta)\delta^t \geq 0$  and the sum of the coefficients of the throughput vector sum to 1, i.e.  $\sum_{t=0}^{\infty} (1 - \delta)\delta^t = 1$ , this implies  $\mathbf{R}(\boldsymbol{\pi}) \in \mathcal{V}^{MIS(G_D)}$ . Hence,  $\mathcal{V}^{MIS(G_D)}(\delta) \subseteq \mathcal{V}^{MIS(G_D)}$ . Therefore,  $\mathcal{V}^{MIS(G_D)}(\delta) = \mathcal{V}^{MIS(G_D)}$  for  $\delta \geq 1 - \frac{1}{s(G_D)}$ . Next, we give a corollary of the above Theorem 2, which states the restriction on the discount factor and the corresponding achievable set for the policy based on the subset of MISs in Section 3.6.  $\blacksquare$

**Corollary 1** *Given the interference graph  $G_D$ , if the  $\delta \geq 1 - \frac{1}{C(G_D)+1}$  then the throughput vectors achieved by the policies in  $\Pi_{approx}^{MIS(G_D)}$  is  $\mathcal{V}_{approx}^{MIS(G_D)}$ .*

**Proof:** This follows on the same lines as Theorem 2.  $\mathcal{V}_{approx}^{MIS(G_D)} = \text{conv}\{\mathcal{R}_{approx}^{MIS(G_D)}\}$ , where  $\mathcal{R}_{approx}^{MIS(G_D)} = \{\mathbf{r}(\mathbf{p}^{I_j^{G_D}}), \forall j \in \{k(1), \dots, k(C(G_D) + 1)\}\}$ . It can be shown on the same lines that  $\mathcal{V}_{approx}^{MIS(G_D)}$  is self-generating if  $\delta \geq 1 - \frac{1}{C(G_D)+1}$ . This is due to the fact that  $\mathcal{V}_{approx}^{MIS(G_D)}$  has  $C(G_D) + 1$  extreme points.  $\blacksquare$

### 3.10.2 Appendix B

**Proof of Theorem 3.** The policy in Table 3.2 is based on the decomposition property of  $\mathcal{V}^{MIS(G_D)}$  (explained in the proof of Theorem 2). Define  $\boldsymbol{\gamma}(t) = \sum_{k=1}^{s(G_D)} \alpha_k(t) \mathbf{r}(\mathbf{p}^{I_k^{G_D}})$ , where  $\alpha_k(t)$ ,  $\forall k \in \{1, \dots, s(G_D)\}$ ,  $\forall t \geq 0$  correspond to the coefficient  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{N+M})$  at the beginning of time slot  $t$  in the policy in Table 3.2. Also, let  $\mathbf{p}^{I_{r_t}^{G_D}}$  correspond to the power vector used for transmission at time  $t$  for  $t \geq 0$ . First, we show that if  $\delta \geq \bar{\delta}$  then  $\boldsymbol{\gamma}(t+1) \in \mathcal{V}^{MIS(G_D)}$ ,  $\forall t \geq 0$ . Expressing  $\boldsymbol{\gamma}(t+1)$  in terms of coefficients of  $\alpha_k(t)$ ,  $\forall k \in \{1, \dots, s(G_D)\}$  as in Table 3.2.

$$\boldsymbol{\gamma}(t+1) = \sum_{k=1}^{s(G_D)} \alpha_k(t+1) \mathbf{r}(\mathbf{p}^{I_k^{G_D}}) = \sum_{k=1, k \neq r_t}^{s(G_D)} \frac{\alpha_k(t)}{\delta} \mathbf{r}(\mathbf{p}^{I_k^{G_D}}) + \frac{\alpha_{r_t}(t) - (1 - \delta)}{\delta} \mathbf{r}(\mathbf{p}^{I_{r_t}^{G_D}}) \quad (3.5)$$

where  $\alpha_{r_t}(t) = \max_k \alpha_k(t)$ . If  $0 \leq \frac{\alpha_k(t)}{\delta} \leq 1, \forall k \in \{1, \dots, s(G_D)\}, k \neq r_t$  and  $0 \leq \frac{\alpha_{r_t}(t)-(1-\delta)}{\delta} \leq 1$  then  $\gamma(t+1) \in \mathcal{V}^{MIS(G_D)}$ . If  $\delta \geq \max_{t \geq 0} 1 - \alpha_{r_t}(t)$  then  $\gamma(t+1) \in \mathcal{V}^{MIS(G_D)}, \forall t$ . We know that  $\alpha_{r_t}(t) \geq \frac{1}{s(G_D)}, \forall t$ , this is true because  $\alpha_{r_t}(t) = \max_k \alpha_k(t)$ . The condition on  $\delta \geq 1 - \frac{1}{s(G_D)}$  implies that  $\gamma(t+1) \in \mathcal{V}^{MIS(G_D)}, \forall t \geq 0$ . Therefore, we can express

$$\gamma(t) = (1 - \delta)\mathbf{r}(\mathbf{p}^{I_{r_t}^{G_D}}) + \delta\gamma(t+1)$$

here  $\gamma(t+1) \in \mathcal{V}^{MIS(G_D)}$ . Hence using this recursively we get,

$$\gamma(0) - (1 - \delta) \sum_{\tau=0}^t \delta^\tau \mathbf{r}(\mathbf{p}^{I_{r_\tau}^{G_D}}) = \delta^{t+1} \gamma(t+1) \quad (3.6)$$

Here,  $\gamma(t+1) \in \mathcal{V}^{MIS(G_D)}$  and  $\gamma(0) = \mathbf{y}^*(G_D)$ . The above expression in (3.6) specifies the difference from the target vector and the vector achieved till time  $t$ . Next, we find the time  $T$  after which the norm of difference in (3.6) is below  $\epsilon$ .  $\gamma(t+1)$  can be bounded above (since  $\mathcal{V}^{MIS(G_D)}$  is closed and bounded),  $\|\gamma(t+1)\| \leq \theta^{bd}, \forall t \geq 0$ . Hence,  $T = \frac{\log(\frac{\epsilon}{\theta^{bd}})}{\log(\delta)} - 1$  is sufficiently high to ensure the norm of difference in 3.6 is below  $\epsilon$ . Hence as  $\epsilon$  is chosen small, the policy would converge to the target payoff.

Next, we give a corollary, which states the condition on the policy based on the subset of the MISs computed in Section 3.6.1. ■

**Corollary 2** For any  $\delta \geq 1 - \frac{1}{C(G_D)+1}$  the policy computed in Table 3.2 based on the subset of MISs computed in Section 3.6.1 achieves the target throughput  $\mathbf{y}_{approx}^*(G_D)$

**Proof:** The policy in Table 3.2 using the target weights  $\alpha_{approx}^*(G_D)$  is based on the decomposition property of  $\mathcal{V}_{approx}^{MIS(G_D)}$ . Here also using the decomposition property repeatedly it can be shown that the difference between the target and the target vector achieved till time  $t$  decreases exponentially which proves convergence.

Next, we state two lemmas which will be used to prove Theorem 4 and 5. Also, note that any inequality between two vectors would represent a component-wise inequality.

**Lemma 1** In a bounded degree graph with  $N+M$  vertices and bound on the maximum degree  $\rho$  there exists a maximal independent set with size  $\frac{N+M}{\rho+1}$

**Proof of Lemma 1.** In a bounded degree graph the maximum chromatic number for the conventional vertex coloring is  $\rho + 1$  [Erc13]. Let the independent sets corresponding to each color class obtained by using the minimum coloring with at most  $\rho + 1$  colors be given as  $\{I'_1, \dots, I'_{\rho+1}\}$  and let the sets of the sizes corresponding to each of these independent sets be given as  $\{n_1, \dots, n_{\rho+1}\}$ . Let the set of maximum size be  $I'_{[1]}$  and the corresponding size be given as  $n_{[1]}$ . We know that  $\sum_{i=1}^{\rho+1} n_i = N + M$  and  $n_{[1]} \geq n_i, \forall i \in \{1, \dots, \rho + 1\}$ . From these conditions we can obtain  $(\rho + 1)n_{[1]} \geq N + M$ . Hence,  $n_{[1]} \geq \frac{N+M}{\rho+1}$ . ■

**Lemma 2** *If the interference graph exhibits  $\epsilon$ -WNI then the expression  $r'_i(\mathbf{p})$  (as defined in Section 3.5.3) is an  $\epsilon$  approximation to the actual rate  $r_i(\mathbf{p})$  and this holds for all  $i \in \{1, \dots, N + M\}$  and  $\forall \mathbf{p} \in \mathcal{P}$ .*

**Proof of Lemma 2.** To show that  $r'_i(\mathbf{p})$  is an  $\epsilon$  approximation, it is sufficient to show that  $\max_{\mathbf{p} \in \mathcal{P}} |e_i(\mathbf{p})| \leq \epsilon$ , where  $e_i(\mathbf{p}) = r'_i(\mathbf{p}) - r_i(\mathbf{p})$ . Solving for  $\arg \max_{\mathbf{p} \in \mathcal{P}} |e_i(\mathbf{p})|$ , we get  $p_i = p_i^{max}$ ,  $p_j = 0$ ,  $\forall j \in \mathcal{N}_i(G_D)$  and  $p_k = p_k^{max}$ ,  $\forall k \in (\mathcal{N}_i(G_D) \cup \{i\})^c$ . This is explained as follows. Since the accumulative interference from non-neighbors is not accounted for in  $r'_i(\mathbf{p})$  this means  $r'_i(\mathbf{p}) \geq r_i(\mathbf{p}) \implies e_i(\mathbf{p}) \geq 0, \forall \mathbf{p} \in \mathcal{P}$ . Note that  $e_i(\mathbf{p}) = \log_2\left(\frac{1 + \frac{g_{ii}p_i}{1 + \sum_{j \in \mathcal{N}_i(G_D)} \frac{g_{ji}p_j + \sigma_i^2}{g_{ii}p_i}}}{1 + \frac{g_{ii}p_i}{1 + \sum_{j \neq i} \frac{g_{ji}p_j + \sigma_i^2}{g_{ii}p_i}}}\right)$  is an increasing function in  $p_i$  and  $p_k$ ,  $\forall k \in (\mathcal{N}_i(G_D) \cup \{i\})^c$  and a decreasing function of  $p_j$ ,  $\forall j \in \mathcal{N}_i(G_D)$ . From this we get that  $e_i(\mathbf{p})$  takes its maximum value when  $p_i = p_i^{max}$  and  $p_k = p_k^{max}$ ,  $\forall k \in (\mathcal{N}_i(G_D) \cup \{i\})^c$  and  $p_j = 0$ ,  $\forall j \in \mathcal{N}_i(G_D)$ . The corresponding maximum value is  $e_i^{max} = \log_2\left(\frac{1 + \frac{g_{ii}p_i^{max}}{\sigma_i^2}}{1 + \frac{g_{ii}p_i^{max}}{\sum_{k \in (\mathcal{N}_i(G_D) \cup \{i\})^c} g_{ki}p_k^{max} + \sigma_i^2}}\right)$ . This combined with  $\epsilon$ -WNI, i.e.  $Int_i^{max}(G_D) = \sum_{k \in (\mathcal{N}_i(G_D) \cup \{i\})^c} g_{ki}p_j^{max} \leq (2^\epsilon - 1)\sigma_i^2$  gives that  $e_i^{max} \leq \epsilon$ . The same argument holds for any  $i$ . ■

### 3.10.3 Appendix C

**Proof of Theorem 4.** With constraint tolerance of  $\epsilon$ , the optimization problem in Step 5 of the design framework is stated as follows.

$$\begin{aligned}
& \max_{\mathbf{y}, \boldsymbol{\alpha}} W(y_1(G_D), \dots, y_{N+M}(G_D)) && (3.7) \\
& y_i(G_D) \geq R_i^{\min} - \epsilon, \quad \forall i \in \{1, \dots, N+M\} \\
& y_i(G_D) = \sum_{k=1}^{s(G_D)} \alpha_k r_i(\mathbf{p}^{I_k^{G_D}}), \quad \forall i \in \{1, \dots, N+M\} \\
& \alpha_k \geq 0, \quad \forall k \in \{1, \dots, s(G_D)\} \\
& \sum_{k=1}^{s(G_D)} \alpha_k = 1
\end{aligned}$$

The optimal argument of the above problem is  $\mathbf{y}^*(G_D)$ . Define

$$\mathbf{y}^*(G_{D^*}) = \arg \max_{D \leq D_{\max}^{BS}} W(\mathbf{y}^*(G_D))$$

Consider the design problem in Section 3.4.1. The feasible region for average throughput vectors for the design problem is a subset of  $\text{conv}\{\mathcal{R}^{\text{const}}\}$ . This is explained as follows. The throughput vector achieved by an interference management policy can be written as:  $\mathbf{R}(\boldsymbol{\pi}) = (1 - \delta) \sum_{t=0}^{\infty} \delta^t \mathbf{r}(\mathbf{p}^t)$ , here  $\mathbf{R}(\boldsymbol{\pi}) = (R_1(\boldsymbol{\pi}), \dots, R_{N+M}(\boldsymbol{\pi}))$ . The coefficients of  $\mathbf{r}(\mathbf{p}^t)$  are positive and the sum of these coefficients is 1 and  $\mathbf{r}(\mathbf{p}^t) \in \mathcal{R}^{\text{const}}$ , which implies that  $\mathbf{R}(\boldsymbol{\pi}) \in \text{conv}\{\mathcal{R}^{\text{const}}\}$ . Let  $\mathbf{v}^* \in \text{conv}\{\mathcal{R}^{\text{const}}\}$  be the optimal solution to the design problem in Section 3.4.1 with weighted sum throughput as the objective and the corresponding optimal value of the objective is  $W^* = \sum_{i=1}^{N+M} w_i v_i^*$ , where  $w_i \geq 0 \quad \forall i \in \{1, \dots, N+M\}$  and  $\sum_{i=1}^{N+M} w_i = 1$ . Note that we have assumed that the design problem in Section 3.4.1 is feasible, otherwise if it is not feasible then clearly the proposed framework will be infeasible as well. Expressing  $\mathbf{v}^*$  in terms of throughput vectors in  $\mathcal{R}^{\text{const}}$  as follows,  $\mathbf{v}^* = \sum_{j=1}^q \theta_j \mathbf{r}^{j,*}$  where  $\mathbf{r}^{j,*} \in \mathcal{R}^{\text{const}}$ ,  $\theta_j \geq 0$ ,  $\forall j \in \{1, \dots, N+M\}$  and  $\sum_{j=1}^q \theta_j = 1$ . Here,  $\mathbf{v}^* \geq \mathbf{R}^{\min}$  where  $\mathbf{R}^{\min} = [R_1^{\min}, \dots, R_{N+M}^{\min}]$  and the inequality between the vectors is a component-wise inequality. Our aim is to show that there exists  $\mathbf{v} \in \mathcal{V}^{\text{MIS}(G_{D^*})}$  which is  $\epsilon$  close to the optimal and satisfies the minimum throughput constraint within a tolerance of  $\epsilon$ . Let  $\mathbf{r}(\mathbf{p}^{j,*}) = \mathbf{r}^{j,*}$  and let  $\mathbf{r}'(\mathbf{p}^{j,*}) \in \mathcal{R}_a^{\text{const}}$  be the corresponding throughput taking only the interference from neighbors into account (given in Section 3.5.3). Let  $\tilde{\mathbf{v}} \in \text{conv}\{\mathcal{R}_a^{\text{const}}\}$  defined as follows  $\tilde{\mathbf{v}} = \sum_{j=1}^q \theta_j \mathbf{r}'(\mathbf{p}^{j,*})$ . Since  $\mathbf{r}'$  is computed only from the interference contribution of the

neighbors we have  $\mathbf{r}'(\mathbf{p}^{j,*}) \geq \mathbf{r}(\mathbf{p}^{j,*})$  and from SLI we know that there exists  $\mathbf{v}'^j \in \mathcal{V}_a^{MIS(G_{D^*})}$  which satisfies

$$\mathbf{v}'^j \geq \mathbf{r}'(\mathbf{p}^{j,*}) \geq \mathbf{r}(\mathbf{p}^{j,*}), \forall j \in \{1, \dots, q\} \quad (3.8)$$

Let  $\mathbf{v}' = \sum_{j=1}^q \theta_j \mathbf{v}'^j$ . Using above (3.8) we get,

$$\mathbf{v}' \geq \tilde{\mathbf{v}} \geq \mathbf{v}^* \quad (3.9)$$

Expressing  $\mathbf{v}'^j$  in terms of the throughput vectors in  $\mathcal{R}_a^{MIS(G_{D^*})}$  we get

$$\mathbf{v}'^j = \sum_{k=1}^{s(G_{D^*})} \beta_k^j \mathbf{r}'(\mathbf{p}^{I_k^{G_{D^*}}})$$

and the corresponding actual throughput is given as  $\mathbf{v}^j = \sum_{k=1}^{s(G_{D^*})} \beta_k^j \mathbf{r}(\mathbf{p}^{I_k^{G_{D^*}}})$ . From the condition in the Theorem we know  $G_{D^*}$  exhibits  $\epsilon$ -WNI which means  $Int_i^{max}(G_{D^*}) \leq (2^\epsilon - 1)\sigma_i^2 \forall i \in \{1, \dots, N+M\}$ . Hence using Lemma 2, we have  $r'_i(\mathbf{p})$  is an  $\epsilon$  approximation to  $r_i(\mathbf{p})$  and this holds for all  $i \in \{1, \dots, N+M\}$ . Using  $\mathbf{r}(\mathbf{p}^{I_k^{G_{D^*}}}) \geq \mathbf{r}'(\mathbf{p}^{I_k^{G_{D^*}}}) - \epsilon, \forall k \in \{1, \dots, s(G_{D^*})\}$  we get

$$\mathbf{v}^j \geq \mathbf{v}'^j - \epsilon. \quad (3.10)$$

Hence, using the lower bound on  $\mathbf{v}'^j \geq \mathbf{r}(\mathbf{p}^{j,*})$  we get  $\mathbf{v}^j \geq \mathbf{r}(\mathbf{p}^{j,*}) - \epsilon$ . Also, the same can be done in general  $\forall j \in \{1, \dots, q\}$ . Using this we get

$$\sum_{j=1}^q \theta_j \mathbf{v}^j = \sum_{k=1}^{s(G_{D^*})} \sum_{j=1}^q \theta_j \beta_k^j \mathbf{r}(\mathbf{p}^{I_k^{G_{D^*}}}) \geq \mathbf{v}^* - \epsilon$$

Let  $\mathbf{v} = \sum_{j=1}^q \theta_j \mathbf{v}^j$  and let  $\mathbf{v} = [v_1, \dots, v_{N+M}]$ . We can get the following relationship.

$$\mathbf{v} \geq \mathbf{v}^* - \epsilon \quad (3.11)$$

It can be seen that we can write  $\mathbf{v} = \sum_{k=1}^{s(G_{D^*})} \alpha_k \mathbf{r}(\mathbf{p}^{I_k^{G_{D^*}}})$  with  $\alpha_k = \sum_{j=1}^q \theta_j \beta_k^j$ . Since  $\alpha_k \geq 0, \forall k \in \{1, \dots, s(G_{D^*})\}$  and  $\sum_{k=1}^{s(G_{D^*})} \alpha_k = 1$ , which means  $\mathbf{v} \in \mathcal{V}^{MIS(G_{D^*})}$ . This gives  $\sum_{i=1}^{N+M} w_i v_i \geq \sum_{i=1}^{N+M} w_i v_i^* - \epsilon$  and  $\mathbf{v} \geq \mathbf{R}^{min} - \epsilon$ . Hence,  $\mathbf{v}$  is a feasible throughput vector

for (3.7). Since  $\mathbf{y}^*(G_{D^*})$  is the optimal solution to the above problem in (3.7), we can state the following,

$$\sum_{i=1}^{N+M} w_i y_i^*(G_{D^*}) \geq \sum_{i=1}^{N+M} w_i v_i \geq \sum_{i=1}^{N+M} w_i v_i^* - \epsilon' \quad (3.12)$$

This proves the result.  $\blacksquare$

### 3.10.4 Appendix D

**Proof of Theorem 5.** We make the following homogeneity assumption  $p_i^{max} = p^{max}, R_i^{min} = R^{min}, \max(D^{UE})_{ii} \leq \Delta, w_i = \frac{1}{N+M}$ . These quantities are fixed to understand the effect of scaling of network size, i.e.  $N + M$ . Let  $r_i^{max} = r_i(p_i = p^{max}, \mathbf{p}_{-i} = \mathbf{0})$ . The optimization in Step 4 in Section 3.5.1.4 is stated with weighted sum throughput as the objective.

$$\begin{aligned} & \max_{\mathbf{y}, \boldsymbol{\alpha}} \quad \sum_{i=1}^{N+M} w_i y_i \\ \text{subject to} \quad & y_i = \sum_{j=1}^{s(G_D)} \alpha_j r_i(\mathbf{p}^{I_j^{G_D}}), \quad \forall i \in \{1, \dots, N+M\} \\ & y_i \geq R^{min} \\ & \sum_{j=1}^{s(G_D)} \alpha_j = 1 \\ & \alpha_j \geq 0, \quad \forall j \in \{1, \dots, s(G_D)\} \end{aligned} \quad (3.13)$$

We formulate another optimization problem whose solution is an upper bound to the above.

$$\begin{aligned} & \max_{\mathbf{y}, \boldsymbol{\alpha}} \quad \sum_{i=1}^{N+M} w_i y_i \\ \text{subject to} \quad & y_i = \sum_{j=1}^{s(G_D)} \alpha_j r_i^{max} \mathbf{1}_{i \in I_j^{G_D}}, \quad \forall i \in \{1, \dots, N+M\} \\ & \sum_{j=1}^{s(G_D)} \alpha_j = 1 \\ & \alpha_j \geq 0, \quad \forall j \in \{1, \dots, s(G_D)\} \end{aligned} \quad (3.14)$$

Here  $\mathbf{1}_{i \in I_j^{G_D}}$  is an indicator function which takes value 1 if  $i \in I_j^{G_D}$  and 0 otherwise. Note that the solving the above optimization problem in (3.14) is equivalent to finding

the maximum weighted independent set [NHK05] with the weights assigned to each vertex  $i$  given as  $\bar{w}_i = \frac{1}{N+M} r_i^{\max}$ . Let  $\mathbf{y}^*(G_D)$  be the optimal solution to (3.13) and the corresponding optimal value is  $\sum_{i=1}^{N+M} w_i y_i^*(G_D)$ . Let the maximum weighted independent set which is a solution to (3.14) be denoted as  $I_{p^*}^{G_D}$  and hence the optimal value of the objective in (3.14) is  $\sum_{i=1}^N w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}^{G_D}}$ . The solution to the problem in (3.14) is an upper bound to the solution of (3.13), this is formally stated as

$$\sum_{i=1}^N w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}^{G_D}} \geq \sum_{i=1}^{N+M} w_i y_i^*(G_D) \quad (3.15)$$

This is explained as follows. Let the feasible sets of problem (3.13) and (3.14) be given as  $\mathcal{F}_1, \mathcal{F}_2$  respectively. If  $[\boldsymbol{\alpha}', \mathbf{y}'] \in \mathcal{F}_1$  then for the same  $\boldsymbol{\alpha}'$  the corresponding  $\mathbf{y}'' = \sum_{j=1}^s \alpha'_j \mathbf{r}^{\max} \mathbf{1}_{i \in I_j^{G_D}}$  satisfies  $\mathbf{y}'' \geq \mathbf{y}'$  since  $r_i^{\max} \geq r_i(\mathbf{p}^{I_j^{G_D}})$ .

Next we use the fact that if the interference graph exhibits  $\zeta$ -WNI, i.e.  $\{Int_i^{\max}(G_D) = \sum_{j \notin \mathcal{N}_i, j \neq i} g_{ji} p^{\max} \leq (2^\zeta - 1)\sigma^2\}$  then  $r_i(\mathbf{p})' = \log_2(1 + \frac{g_{ii} p^{\max}}{\sum_{j \in \mathcal{N}_i(G_D)} g_{ji} p_j + \sigma^2})$  is an  $\zeta$  approximation of  $r_i(\mathbf{p})$ . This follows from Lemma 2. Thus we have

$$r_i^{\max} \mathbf{1}_{i \in I_j^{G_D}} - r_i(\mathbf{p}^{I_j^{G_D}}) \leq \zeta \quad (3.16)$$

The weight of approximate weighted maximum independent set  $I_{k(C(G_D)+1)}^{G_D}$  is given as  $\sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{k(C(G_D)+1)}^{G_D}}$ .  $I_{k(C(G_D)+1)}^{G_D}$  is  $\eta$  approximate independent set computed using the algorithm in [NHK05], hence, we can write

$$\sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{k(C(G_D)+1)}^{G_D}} \geq \frac{1}{1+\eta} \left( \sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}^{G_D}} \right) \quad (3.17)$$

If  $\alpha_{k(C(G_D)+1)} = 1$  and  $\alpha_j = 0, \forall j \neq k(C(G_D) + 1)$  then the resulting value of the objective function is  $\sum_{i=1}^{N+M} w_i r_i(\mathbf{p}^{I_{k(C(G_D)+1)}^{G_D}})$ . From (3.16) and (3.17) we have,

$$\sum_{i=1}^{N+M} w_i r_i(\mathbf{p}^{I_{k(C(G_D)+1)}^{G_D}}) \geq \sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{k(C(G_D)+1)}^{G_D}} - \zeta \geq \frac{1}{1+\eta} \left( \sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}^{G_D}} \right) - \zeta \quad (3.18)$$

The minimum value that the expression  $(\frac{1}{1+\eta} \sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}^{G_D}})$  can take is given as  $\frac{1}{(\rho+1)(\eta+1)} \log_2(1 + \frac{1}{\Delta^{np}} \frac{p^{\max}}{\sigma^2})$ . To derive this, first substitute the value of  $w_i = \frac{1}{N+M}$ .

From Lemma 1 we know that there exists a maximal independent set with size no less than  $\frac{N+M}{\rho+1}$ . Also, using the minimum value of the direct channel gain  $g_{ii} \geq \frac{1}{\Delta^{np}}$  we get  $r_i^{\max} \geq \log_2(1 + \frac{1}{\Delta^{np}} \frac{p^{\max}}{\sigma^2})$ . Combining the fact that maximal independent set has a size no less than  $\frac{N+M}{\rho+1}$  and  $r_i^{\max} \geq \log_2(1 + \frac{1}{\Delta^{np}} \frac{p^{\max}}{\sigma^2})$  we get the minimum value of the expression.

From the condition in the Theorem we have  $\zeta < \frac{\kappa}{(\rho+1)(\eta+1)} \log_2(1 + \frac{1}{\Delta^{np}} \frac{p^{\max}}{\sigma^2})$ , where  $0 < \kappa < 1$  determines the distance from the optimal solution. If  $\zeta$  is selected based on this threshold then,  $\sum_{i=1}^{N+M} w_i r_i(\mathbf{p}^{I_k(C(G_D)+1)}) \geq \frac{1-\kappa}{1+\eta} (\sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{v_i \in I_{p^*}^{G_D}})$ . Using the fact that solution to (3.14) is an upper bound to (3.13) as stated in (3.15), we get the following.

$$\sum_{i=1}^{N+M} w_i r_i(\mathbf{p}^{I_k(C(G_D)+1)}) \geq \frac{1-\kappa}{1+\eta} \left( \sum_{i=1}^{N+M} w_i r_i^{\max} \mathbf{1}_{i \in I_{p^*}} \right) \geq \frac{1-\kappa}{1+\eta} \sum_{i=1}^{N+M} w_i y_i^*(G_D) \quad (3.19)$$

Next, we state the optimization problem to compute  $W(\mathbf{y}_{approx}^*(G_D))$  which is similar to the problem in Step-4 in the Section 3.5 but uses the subset of the MISs computed in Section 3.6.1, here  $\mathbf{y}_{approx}^*(G_D)$  is the corresponding optimal argument. Note that  $W(\mathbf{y}_{approx}^*(G_D))$  is the optimal value that can be attained by the policy based on the subset of MISs computed in Section 3.6.1.

$$\begin{aligned} \max_{\mathbf{y}, \boldsymbol{\alpha}} \quad & \sum_{i=1}^{N+M} w_i y_i \\ \text{subject to} \quad & y_i = \sum_{j=1}^{C(G_D)+1} \alpha_j r_i(\mathbf{p}^{I_{k(j)}^{G_D}}), \quad \forall i \in \{1, \dots, N+M\} \\ & y_i \geq R^{\min}, \quad \forall i \in \{1, \dots, N+M\} \\ & \sum_{j=1}^{C(G_D)+1} \alpha_j = 1 \\ & \alpha_j \geq 0, \quad \forall j \in \{1, \dots, C(G_D) + 1\} \end{aligned} \quad (3.20)$$

Using  $g_{ii} \geq \frac{1}{\Delta^{np}}$  and the fact that  $Int_i^{\max}(G_D) \leq (2^\zeta - 1)\sigma^2$  ( $\zeta$ -WNI) ensures that following is true

$$r_i(\mathbf{p}^{I_{k(j)}^{G_D}}) \geq \log_2(1 + \frac{p^{\max}}{\Delta^{np} 2^\zeta \sigma^2}) \mathbf{1}_{i \in I_{k(j)}^{G_D}}, \quad \forall j \in \{1, \dots, C(G_D)\} \quad (3.21)$$

We now develop a feasible solution to the above problem in (3.20). Assign  $\beta_{approx,k(j)} =$

$\frac{R^{min}}{\log_2(1+\frac{p^{max}}{\Delta^{np_2\zeta\sigma^2}})}$ ,  $\forall j \in \{1, \dots, C(G_D)\}$ . Hence, we can write

$$\sum_{j=1}^{C(G_D)} \beta_{approx,k(j)} r_i(\mathbf{p}^{I_k^{G_D}}) \geq R^{min} \sum_{j=1}^{C(G_D)} \mathbf{1}_{i \in I_k^{G_D}}, \forall j \in \{1, \dots, C(G_D)\} \quad (3.22)$$

The maximal independent sets  $\{I_{k(1)}^{G_D}, \dots, I_{k(C(G_D))}^{G_D}\}$  are obtained by adding vertices in a greedy manner in ii) to the independent sets constituting color classes, as in Section 3.6.1. Hence, all these MISs together contain all the vertices in the graph, this ensures  $\sum_{j=1}^{C(G_D)} \mathbf{1}_{i \in I_k^{G_D}} \geq 1$ . Therefore, this assignment of  $\beta_{approx,k(j)}$  ensures the minimum throughput constraint in (3.20) is satisfied. However, it still needs to be checked if  $\sum_{j=1}^{C(G_D)} \beta_{approx,k(i)} \leq 1$ . Since  $C(G_D) \leq 3(\rho + 1)$  from [MBH95]. The condition on  $\rho$ , i.e.  $\rho < \frac{\log_2(1+\frac{p^{max}}{\Delta^{np_2\epsilon\sigma^2}})}{3R^{min}} - 1$  ensures  $\sum_{j=1}^{C(G_D)} \beta_{approx,k(i)} = \gamma = (3(\rho + 1)) \frac{R^{min}}{\log_2(1+\frac{p^{max}}{\Delta^{np_2\epsilon\sigma^2}})} < 1$ . Hence, the solution obtained is feasible. The remaining fraction  $1 - \gamma$  is assigned in such a way to ensure a constant competitive ratio.

Assign  $\beta_{approx,k(C(G_D)+1)} = (1 - \gamma)$  then the lower bound is given as  $W(\mathbf{y}_{approx}^*(G_D)) \geq (1 - \gamma) \sum_{i=1}^{N+M} w_i r_i(\mathbf{p}^{I_k^{G_D}})$ . This combined with the lower bound in (3.19) derived above gives:

$$W(\mathbf{y}_{approx}^*(G_D)) \geq \frac{(1 - \kappa)(1 - \gamma)}{1 + \eta} \sum_{i=1}^{N+M} w_i y_i^*(G_D) \quad (3.23)$$

■

# CHAPTER 4

## Distributed Large Scale Multi-Agent Resource Sharing

### 4.1 Introduction

In this chapter, we describe a method for distributed large scale resource sharing between multiple agents. We describe the proposed method in the context of wireless networks. At the end of the chapter, we show how the proposed method is general and applies to many scenarios. This chapter tries to solve the problem in Chapter 3 but with the additional constraints that all the agents cooperate and distributedly compute a resource sharing plan. The approach proposed in this chapter also extends the method and the results described in Chapter 3. This chapter is based on my work in [AXS15a].

**Motivation.** Dense deployment of low-cost heterogeneous small cells (e.g. picocells, femtocells) has become one of the most effective solutions to accommodate the exploding demand for wireless spectrum [HLN13] [GMR12] [ACD12]. On one hand, dense deployment of small cells significantly shortens the distances between small cell base stations (SBSs) and their corresponding user equipments (UEs), thereby boosting the network capacity. On the other hand, dense deployment also shortens the distances between neighboring SBSs, thereby potentially increasing the inter-cell interference. Hence, while the solution provided by the dense deployment of small cells is promising, its success depends crucially on interference management by the small cells. Efficient interference management is even more challenging in heterogeneous small cell networks, due to the lack of central coordinators, compared to that in traditional cellular networks.

In this chapter, we propose a novel framework for designing interference management policies in the uplink of small cell networks, which specify when and at what power level

each UE should transmit.<sup>1</sup> Our proposed design framework and the resulting interference management policies fulfill *all* the following important requirements:

- *Deployment of heterogeneous small cell networks:* Existing deployments of small cell networks exhibit significant heterogeneity such as different types of small cells (picocells and femtocells), different cell sizes, different number of UEs served, different UEs' throughput requirements etc.
- *Interference avoidance and spatial reuse:* Effective interference management policies should take into account the strong interference among neighboring UEs, as well as the weak interference among non-neighboring UEs. Hence, the policies should effectively avoid interference among neighboring UEs and use spatial reuse to take advantage of the weak interference among non-neighboring UEs.
- *Distributed implementation with local information and message exchange:* Since there is no central coordinator in small cell networks, interference management policies need to be computed and implemented by the UEs in a distributed manner, by exchanging only local information through local message exchanges among neighboring UE-SBS pairs.
- *Scalability to large networks:* Small cells are often deployed over a large scale (e.g., in a city). Effective interference management policies should scale in large networks, namely achieve efficient network performance while maintaining low computational complexity.
- *Ability to optimize different performance criteria:* Under different deployment scenarios the small cell networks may have different performance criteria, e.g., weighted sum throughput or max-min fairness. The design framework should be general and should prescribe different policies to optimize different network performance criteria.

---

<sup>1</sup>Although we focus on uplink transmissions in this chapter, our framework can be easily applied to downlink transmissions.

- *Performance guarantees for individual UEs:* Effective interference management should provide performance guarantees (e.g., minimum throughput guarantees) for individual UEs.

As we will discuss in detail in Section 4.2, existing state-of-the-art policies for interference management cannot simultaneously fulfill all of the above requirements.

Next, we describe our key results and major contributions:

1. We propose a general framework for designing distributed interference management policies that maximizes the given network performance criterion subject to each UE's minimum throughput requirements. The proposed policies schedule maximal independent sets (MISs)<sup>2</sup> of the interference graph to transmit in each time slot. In this way, they avoid strong interference among neighboring UEs (since neighboring UEs cannot be in the same MIS), and efficiently exploit the weak interference among UEs in a MIS by letting them to transmit at the same time.
2. We propose a distributed algorithm for the UEs to determine a subset of MISs. The subset of MISs generated ensures that each UE belongs to at least one MIS in this subset. Moreover, the subset of MISs can be generated in a distributed manner in logarithmic time (logarithmic in the number of UEs in the network) for bounded-degree interference graphs.<sup>3</sup> The logarithmic convergence time is significantly faster than the time (linear or quadratic in the number of UEs) required by the distributed algorithms for generating subsets of MISs in [RP89, CS89, ET90].
3. Given the computed subsets of MISs, we propose a distributed algorithm in which each UE determines the optimal fractions of time occupied by the MISs with only local message exchange. The message is exchanged only among the UE-SBS pairs that strongly

---

<sup>2</sup>Consider the interference graph of the network, where each vertex is a UE-SBS pair and each edge indicates strong interference between the two vertices. An independent set (IS) is a set of vertices in which no pair is connected by an edge. An IS is a MIS if it is not a proper subset of another IS.

<sup>3</sup>Bounded-degree graphs are the graphs whose maximum degree can be bounded by a constant independent of the size of the graph, i.e.,  $\Delta = \mathcal{O}(1)$ . As we will show in Theorem 10, for the interference graphs that are not bounded-degree graphs, even the centralized solution, given all the MISs, cannot satisfy the minimum throughput requirements.

interfere with each other, i.e. among neighbors in the interference graph. The distributed algorithm will output the optimal fractions of time for each MIS such that the given network performance criterion is maximized subject to the minimum throughput requirements.

4. Under a wide range of conditions, we analytically characterize the competitive ratio of the proposed distributed policy with respect to the optimal network performance. Importantly, we prove that the competitive ratio is independent of the network size, which demonstrates the scalability of our proposed policy in large networks. Remarkably, the constant competitive ratio is achieved even though our proposed policy requires only local information, is distributed, and can be computed fast, while the optimal network performance can only be obtained in a centralized manner with global information (e.g., all the UEs' channel gains, maximum transmit power levels, minimum throughput requirements).

5. Through simulations, we demonstrate significant (from 160% to 700 %) performance gains over state-of-the-art policies. Moreover, we show that our proposed policies can be easily adapted to a variety of heterogeneous deployment scenarios, with dynamic entry and exit of UEs.

The rest of the chapter is organized as follows. In Section 4.2, we discuss the related works and their limitations. We describe the system model in Section 4.3. Then we formulate the interference management problem and give a motivating example in Section 4.4. We propose the design framework in Section 4.5, and demonstrate the performance gain of our proposed policies in Section 4.6. In Section 4.7, we discuss how the proposed framework is general and can be applied to other applications. Finally, we conclude the chapter in Section 4.8.

## 4.2 Related Works

State-of-the-art interference management policies can be divided into three main categories: policies based on power control, policies based on spatial reuse, and policies based on joint power control and spatial reuse. In the following, we discuss their limitations for the considered distributed interference management problem in heterogeneous small cell networks.

#### 4.2.1 Distributed Interference Management Based on Power Control

Policies based on distributed power control (representative works [HBH06, SMG01, HYC09]) have been used for interference management in both cellular and ad-hoc networks. In these policies, all the UEs in the network transmit at *constant* power levels all the time (provided that the system parameters remain the same).<sup>4</sup> For this reason, we refer to them as constant power control policies in the rest of this chapter. The major limitation of constant power control policies is the difficulty in providing minimum throughput guarantees for each UE, especially in the presence of strong interference. Some works [HBH06, SMG01, HYC09] use pricing to mitigate the strong interference. However, they cannot strictly guarantee the UEs' minimum throughput requirements [HBH06, SMG01, HYC09]. Indeed, the low throughput experienced by some users, caused by strong interference, is the fundamental limitation of constant power control policies, even for the optimal constant power control policy obtained by a central controller<sup>5</sup> [CTP07]. Since strong interference is very common in dense small cell deployments (e.g., in offices and apartments where SBSs are installed close to each other [LCV09]), constant power control policies do not perform well in these scenarios. Note that there exist a different strand of works based on [FM93], which proposes distributed algorithms to achieve the desired minimum throughput requirement for each UE with the objective of minimizing transmit power levels. These works cannot optimize network performance criteria such as weighted sum throughput, max-min fairness etc., and hence are suboptimal under these performance criteria.

#### 4.2.2 Distributed Spatial Reuse Based on Maximal Independent Sets

An efficient solution to mitigate strong interference is spatial reuse, in which only a subset of UEs (those who do not significantly interfere with each other) transmit at the same time.

---

<sup>4</sup>Although some power control policies [HBH06, SMG01, HYC09] go through a transient period of adjusting the power levels before the convergence to the optimal power levels, the users maintain constant power levels after the convergence.

<sup>5</sup>In the case of average sum throughput maximization given the minimum average throughput constraints of the UEs, the power control policies are inefficient if the feasible rate region is non-convex [SWB09].

Spatial Time reuse based Time Division Multiple Access (STDMA) has been widely used in existing works on broadcast scheduling in multi-hop networks [RP89] [ET90].<sup>6</sup> Hence, we also compare with these works to have a comprehensive literature review. Specifically, these policies construct a cyclic schedule such that in each time slot an MIS of the interference graph is scheduled. The constructed schedule ensures that each UE is scheduled at least once in the cycle.

In terms of performance, STDMA policies [RP89, CS89, ET90] cannot guarantee the minimum throughput requirement of each UE, and usually adopt a fixed scheduling (i.e. follow a fixed order in which the MISs are scheduled), which may be very inefficient depending on the given network performance criteria. For example, the policies in [ET90] are inefficient in terms of fairness. In terms of complexity, for the distributed generation of the subsets of MISs, the STDMA policies in [RP89, CS89, ET90] require an ordering of all the UEs, and have a computational complexity (in terms of the number of steps executed by the algorithm) that scales as  $\mathcal{O}(|V|)$  (in [CS89] [ET90]) or  $\mathcal{O}(|V||E|)$  (in [RP89]), where  $|V|$  and  $|E|$  are the number of vertices/UEs and the number of edges in the interference graph, respectively. Hence, in large-scale dense deployments, the complexity grows superlinearly with the number of UEs, making the policies difficult to compute. By contrast, our proposed distributed algorithm for generating subsets of MISs does not require the ordering of all the UEs, and has a complexity that scales as  $\mathcal{O}(\log |V|)$ , namely logarithmically with the number of the UEs, for bounded-degree graphs.<sup>7</sup>

Finally, the STDMA policies in [RP89, CS89, ET90] are designed for the MAC layer and assume that all the UEs are homogeneous at the physical layer. In practice, different UEs are heterogeneous due to their different distances from their SBSs, their different maximum transmit power levels, etc. This heterogeneity is important, and will be considered in our design framework.

<sup>6</sup>These works [RP89, CS89, ET90] do not have exactly the same model as in our setting. However, these works can be adapted to our model.

<sup>7</sup>As we will show in Theorem 10, for graphs which do not have bounded degrees, even a centralized solution based on all the MISs cannot satisfy the minimum throughput requirements.

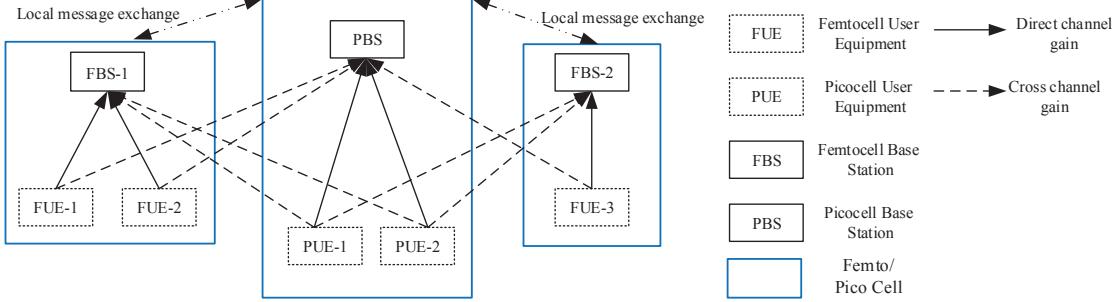


Figure 4.1: Illustration of a heterogeneous small cell network.

#### 4.2.3 Distributed Power Control and Spatial Reuse For Multi-Cell Networks

The works discussed in the above two sections either focus on distributed power control in the physical layer [HBH06, SMG01, HYC09] or focus on distributed spatial reuse in the MAC layer [RP89, CS89, ET90]. Similar to our chapter, some works (see [KG08] [GKG07] for representative works) adopted a cross-layer approach and proposed joint distributed power control and spatial reuse for multi-cell networks. Although these works schedule a subset of UEs to transmit at each time slot, the subset is not the MIS of the interference graph [KG08] [GKG07]. For example, the policies in [KG08] [GKG07], called power matched scheduling (PMS) policies, schedule one UE from each small cell at the same time, even if some UEs from different cells are very close to each other. In this case, these UEs will experience strong inter-cell interference. Hence, the works in [KG08] [GKG07] cannot perfectly eliminate strong interference from neighboring cells and exploit weak interference from non-neighboring cells. Moreover, the works in [KG08] [GKG07] cannot provide minimum throughput guarantees for the UEs.

### 4.3 System Model

#### 4.3.1 Heterogeneous Network of Small Cells

We consider a heterogeneous network of  $K$  small cells operating in the same frequency band (see Fig. 4.1), which represents a common deployment scenario considered in practice [GMR12] [HYC09] [lte]. Note that the small cells can be of different types (e.g. picocells,

femtocells, etc.) and thereby belong to different tiers in the heterogeneous network. Each small cell  $j$  has one SBS, (SBS  $j$ ), which serves a set of UEs under a closed access scenario [HYC09]. Denote the set of UEs by  $\mathcal{U} = \{1, \dots, N\}$ . We write the association of UEs to SBSs as a mapping  $T : \{1, \dots, N\} \rightarrow \{1, \dots, K\}$ , where each UE  $i$  is served by SBS  $T(i)$ . The interference graph  $G$  of the network has  $N$  vertices, each of which is a UE-SBS pair. There is an edge between two pairs/vertices if their cross interference is high (rules for deciding if interference is high will be discussed later).

We focus on the uplink transmissions; the extension to downlink transmissions is straightforward when each SBS serves one UE at a time (e.g., TDMA among the UEs connected to the same SBS). Each UE- $i$  chooses its transmit power  $p_i$  from a compact set  $\mathcal{P}_i \subseteq \mathbb{R}_+$ . We assume that  $0 \in \mathcal{P}_i$ ,  $\forall i \in \{1, \dots, N\}$ , namely any UE can choose not to transmit. The joint power profile of all the UEs is denoted by  $\mathbf{p} = (p_1, \dots, p_N) \in \mathcal{P} \triangleq \prod_{i=1}^N \mathcal{P}_i$ . Under the joint power profile  $\mathbf{p}$ , the signal to interference and noise ratio (SINR) of UE  $i$ 's signal, experienced at its serving SBS  $j = T(i)$ , can be calculated as  $\gamma_i(\mathbf{p}) = \frac{g_{ij}p_i}{\sum_{k=1, k \neq i}^N g_{kj}p_k + \sigma_j^2}$ , where  $g_{ij}$  is the channel gain from UE  $i$  to SBS  $j$ , and  $\sigma_j^2$  is the noise power at SBS  $j$ . Since the UEs do not cooperate to encode their signals to avoid interference, each UE-SBS pair treats the interference from other UEs as white noise. Hence, each UE  $i$  gets the following throughput [KG08],  $r_i(\mathbf{p}) = \log_2(1 + \gamma_i(\mathbf{p}))$ .<sup>8</sup>

### 4.3.2 Interference Management Policies

The system is time slotted at  $t = 0, 1, 2, \dots$ , and the UEs are assumed to be synchronized.<sup>9</sup> At the beginning of each time slot  $t$ , each UE  $i$  decides its transmit power  $p_i^t$  and obtains a throughput of  $r_i(\mathbf{p}^t)$ . Each UE  $i$ 's strategy, denoted by  $\pi_i : \mathbb{Z}_+ \triangleq \{0, 1, \dots\} \rightarrow \mathcal{P}_i$ , is a mapping from time  $t$  to a transmission power level  $p_i \in \mathcal{P}_i$ . The interference management policy is then the collection of all the UEs' strategies, denoted by  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_N)$ . The average throughput

---

<sup>8</sup>We use the Shannon capacity here. However, our analysis is general and applies to the throughput models that consider the modulation scheme used.

<sup>9</sup>Strict synchronization is required for inter-cell interference coordination (ICIC) in Release 10 of 3GPP [lte] and is widely assumed in the literature as well [GMR12] [RP89, CS89, ET90] [KG08] [GKG07].

for UE- $i$  is given by  $R_i(\boldsymbol{\pi}) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T r_i(\mathbf{p}^t)$ , where  $\mathbf{p}^t = (\pi_1(t), \dots, \pi_N(t))$  is the power profile at time  $t$ . We assume that the channel gains are fixed over the considered time horizon as in [KG08] [JPP05, LXH10, UAB11, LLJ10]. However, we will illustrate in Section 4.6 that our framework performs well under time-varying channel conditions (e.g., due to fading) as well.

An interference management policy  $\boldsymbol{\pi}^{const}$  is a constant power control policy [HBH06, SMG01, HYC09] if  $\boldsymbol{\pi}^{const}(t) = \mathbf{p}$  for all  $t$ . As we have discussed before, our proposed policy is based on MISs of the interference graph. Given an interference graph, we write  $\mathbf{I} = \{I_1, \dots, I_{N_{MIS}}\}$  as the set of all the MISs of the interference graph. Let  $\mathbf{p}^{I_j}$  be a power profile in which the UEs in the MIS  $I_j$  transmit at their maximum power levels and the other UEs do not transmit, namely  $p_k^{I_j} = p_k^{max} \triangleq \max \mathcal{P}_k$  if  $k \in I_j$  and  $p_k = 0$  otherwise. Let  $\mathcal{P}^{MIS} = \{\mathbf{p}^{I_1}, \dots, \mathbf{p}^{I_{N_{MIS}}}\}$  be the set of all such power profiles. Then  $\boldsymbol{\pi}$  is a policy based on MIS if  $\boldsymbol{\pi}(t) \in \mathcal{P}^{MIS}$  for all  $t$ . We denote the set of policies based on MISs by  $\Pi^{MIS} = \{\boldsymbol{\pi} : \mathbb{Z}_+ \rightarrow \mathcal{P}^{MIS}\}$ .

## 4.4 Problem Formulation

In this section, we formulate the interference management policy design problem.

### 4.4.1 The Interference Management Policy Design Problem

We aim to optimize a chosen network performance criterion  $W(R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi}))$ , defined as a function of the UEs' average throughput. We can choose any performance criterion that is concave in  $R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi})$ . For instance,  $W$  can be the weighted sum of all the UEs' throughput  $\sum_{i=1}^N w_i R_i(\boldsymbol{\pi})$  with  $\sum_{i=1}^N w_i = 1$  and  $w_i \geq 0$ . Alternatively, the network performance can be max-min fairness (i.e., the worst UE's throughput)  $\min_i R_i(\boldsymbol{\pi})$ . The policy design problem (PDP) can be then formalized as follows

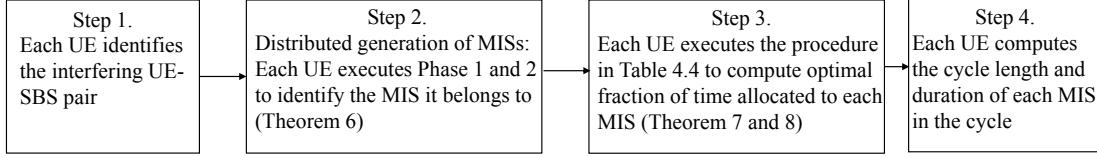


Figure 4.2: Steps in the design framework.

### Policy Design Problem (PDP)

$$\begin{aligned}
 \max_{\boldsymbol{\pi}} \quad & W(R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi})) \\
 \text{subject to} \quad & R_i(\boldsymbol{\pi}) \geq R_i^{\min}, \forall i \in \{1, \dots, N\}
 \end{aligned}$$

The above design problem is very challenging to solve even in a centralized manner (it is NP-hard [TFL11] when  $W$  is the sum throughput). Denote the optimal value of the PDP as  $\mathbf{W}_{\text{opt}}$ . Our goal is to develop distributed, fast algorithms to construct policies that achieve a constant competitive ratio with respect to  $\mathbf{W}_{\text{opt}}$ , with the competitive ratio independent of the network size. We achieve our goal by focusing on policies based on MISs  $\Pi^{\text{MIS}}$ , among other innovations that will be described later.

## 4.5 Design Framework for Distributed Interference Management

### 4.5.1 Proposed Design Framework

Our proposed design framework (see Fig. 4.2) consists of the following four steps.

**Step 1. Identification of the interfering neighbors:** In Step 1, each UE-SBS pair identifies the UE-SBS pairs that strongly interfere with it. Essentially, each pair obtains a *local* view (i.e., its neighbors) of the interference graph. Note that an edge exists between two pairs if at least one of them identifies the other as a strong interferer.

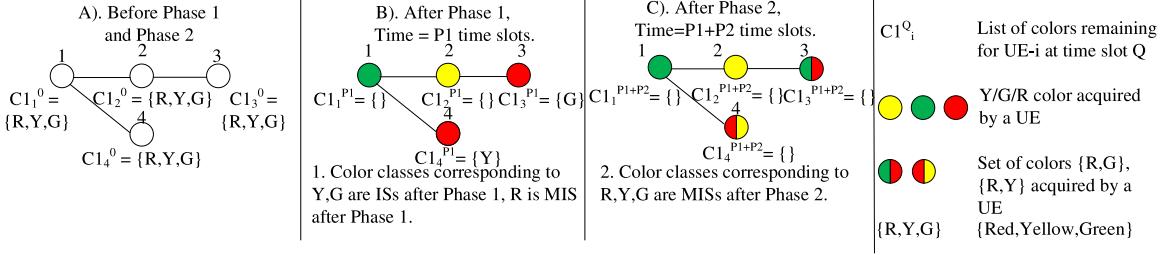


Figure 4.3: Illustration of the distributed generation of MISs in Step 2.

Specifically, each UE-SBS pair is first informed of other pairs in the geographical proximity by managing servers (e.g., femtocell controllers/gateways) [HWK10] [LGD11] [LXH10] [UAB11]. Then each pair can decide whether another pair is strongly interfering based on various rules, such as rules based on Received Signal Strength (RSS) in the *Physical Interference Model* [HWK10] [LXH10] [UAB11], and rules based on the locations in the *Protocol Model* [JPP05]. If one pair identifies another pair as strongly interfering, its decision can be relayed by the managing servers to the latter, such that any two pairs can reach consensus of whether there exists an edge between them.

**Step 2. Distributed generation of MISs that span all the UEs:** In Step 2, the UE-SBS pairs generate a subset of MISs in a distributed fashion. It is important that the generated subset spans all the UEs, namely every UE is contained in at least one MIS in the subset. Otherwise, some UEs will never be scheduled.

The key idea is that from a given list of colors, each UE has to choose a set of colors such that the choice does not conflict with its neighbors. We should ensure that each UE has at least one color. We call the set of UEs with the same color “a color class”. In addition, we should also ensure that every color class is a MIS. This step is composed of two phases: first, distributed coloring of the interference graph based on [Joh99], and second, extension of color classes to MISs. All the UEs are synchronized and carry out their computation simultaneously. We now explain the algorithm in detail. The pseudo-codes can be found in Table 4.2 and 4.3.

**Phase 1. Distributed coloring of the interference graph:** Let  $H^{10}$  be the maximum

---

<sup>10</sup>The maximum number of colors  $H$  should be set to be larger than the maximum number of UE-SBS

number of colors given to all SBSs at the installation and  $d_i$  be the degree (number of neighbors in the interference graph) of the  $i^{th}$  pair. The goal of this phase is to let each UE-SBS pair  $i$  choose *one* color from  $C_i^0 \triangleq \{1, \dots, H\} \cap \{1, \dots, d_i + 1\}$ , such that no neighbors choose the same color. The distributed coloring works as follows.

- i) At the beginning of each time slot  $t$ , each UE  $i$  chooses a color from the set of remaining colors  $C_i^t$  uniformly randomly, and informs its neighbors of its tentative choice. This information can be transmitted through the back-haul network/X2 interface that is used for ICIC [LGD11].
- ii) If the tentative choice of a UE does not conflict with any of its neighbors, then it fixes its color choice and informs the neighbors of its choice. This UE does not contend for colors any further in Phase 1. The neighbors delete the color chosen by  $i$  from their lists  $C_j^{t+1}, \forall j \in \mathcal{N}(i)$ , where  $\mathcal{N}(i)$  is the set of  $i$ 's neighbors.
- iii) Otherwise, if there is a conflict, then the UE does not choose that color and repeats i) and ii) in the next time slot.

There are  $\lceil c_1 \log_{\frac{4}{3}} N \rceil + 1$  time slots in Phase 1, where  $c_1$  is the parameter given by the protocol. The number of time slots is known to the SBSs at installation. Phase 1 is successful if all the UEs acquire a color, which implies that the set of color classes (i.e., the set of UE-SBS pairs with the same color) spans all the UEs.

**Phase 2. Extending color classes to the MISs:** Each color class obtained at the end of Phase 1 is an independent set (IS) of the graph. In Phase 2, we extend each of these ISs to MISs and possibly generate additional MISs. After Phase 1, each UE has chosen one color and deleted some colors from its list. But there may still be remaining colors in its list that are not acquired by any of its neighbors. If the UEs can acquire these remaining colors without conflicting with its neighbors, then each color class will be a MIS. Phase 2 works as follows.

---

pairs interfering with any UE-SBS pair. The SBSs can determine  $H$  according to the deployment scenario.  $H$  in general will also include the number of UEs that use the same SBS who interfere with each other along with the other neighboring UEs. For example,  $H$  can be 10-15 in an office building with dense deployment of SBSs, and can be 3-5 in a residential area.

- i) At each time slot in Phase 2, UE  $i$  chooses each color from the remaining colors in its list independently with probability  $c$ . Each UE  $i$  then sends the set of its tentative choices to its neighboring UEs, and receives their neighbors' choices.
- ii) For any tentative choice of color, if there is a conflict with at least one neighbor, then that color is not fixed; otherwise, it is fixed.
- iii) At the end of each time slot, each UE deletes its set of fixed colors from its list, and transmits this set of fixed colors to its neighbors, who will delete these fixed colors from their lists as well. Note that a UE deletes a particular color if and only if the UE itself or some of its neighbors have chosen this color. Based on this key observation, we can see that if a color is not in any UE's list, the set of UEs with this color is a MIS. If all the UEs have an empty list, then for any color in the set  $\{1, \dots, H\}$ , the set of UEs with this color is a MIS.

There are  $\lceil c_2 \log_x N \rceil + 1$  time slots in Phase 2, where  $x = \frac{1}{1-(c)^H(1-c)^{H^2}}$ , and  $c_2$  is the parameter given by the protocol. The number of time slots is known to the SBSs at installation. We say that Phase 2 is successful, if it finds  $H$  MISs, or equivalently if all the UEs have an empty list.

**Example:** We illustrate Step 2 in a network of 4 UE-SBS pairs, whose interference graph is shown in Fig. 4.3. At the start, each pair has a list of 3 colors {Red, Yellow, Green}. Phase 1 is run for  $P1 = \lceil c_1 \log_{\frac{4}{3}} 5 \rceil$  time slots. At the end of Phase 1, UE 1 and UE 2 acquire Green and Yellow respectively, while UEs 3-4 acquire Red. Hence, UE 1 (UE 2) has an empty list, as Green (Yellow) is acquired by itself and Red, Yellow (Green) by its neighbors. UE 3 (UE 4) has Green (Yellow) color in its list of remaining colors. At the end of Phase 1, the Red color class is a MIS, while the Yellow and Green color classes are not. Phase 2 is run for  $P2 = \lceil c_2 \log_x 5 \rceil + 1$  time slots. UE 3 (UE 4) acquires the remaining color Green (Yellow). At the end of Phase 2, the Green and Yellow color classes become MISs too.

The next theorem establishes the high success probability of Step 2.

**Theorem 6** *For any interference graph with the maximum degree  $\Delta \leq H - 1$ , the proposed algorithm in Table 4.2 and 4.3 outputs a set of  $H$  MISs that span all the UEs in  $(\lceil c_1 \log_{\frac{4}{3}} N \rceil + \lceil c_2 \log_x N \rceil + 2)$  time slots with a probability no smaller than  $(1 - \frac{1}{N^{c_1-1}})(1 - \frac{1}{N^{c_2-1}})$ , where*

$c_1$  and  $c_2$  are design parameters that trade-off the run time and the success probability.

The proofs to all the theorems in this chapter can be found in the Appendix given at the end of the chapter.

Theorem 6 characterizes the performance of our proposed algorithm, in terms of the run time of the algorithm and the lower bound of the success probability. When the parameters  $c_1$  and  $c_2$  are larger, the lower bound of the success probability increases at the expense of a longer run time. When the maximum degree of the interference graph is larger, we need to set a higher  $H$ , which results in a longer run time. This is reasonable, because it is harder to find coloring and MISs when the number of interfering neighbors is higher. Finally, we can see that the lower bound of the successful probability is very high even under smaller  $c_1$  and  $c_2$ , especially if the number of UEs is large. Note that the exact successful probability should depend on the probability  $c$  in Phase 2, while the lower bound in Theorem 6 does not. Hence, our lower bound is robust to different system parameters. Note also that the interference graph here is a bounded-degree graph since the maximum degree is bounded by a given constant,  $H - 1$ . The algorithms in [RP89] [ET90] (require ordering of the vertices, work sequentially and have a higher complexity) can be used to output the MISs spanning all the UEs for arbitrary graphs. However, we will show in Theorem 10, that the restriction to bounded-degree graphs is a must to ensure that the minimum throughput requirement of each UE is satisfied for any MIS based policy.

**Step 3. Distributed computation of the optimal fractions of time for each MIS:** Let the set of MISs generated in Step 2 be  $\{I'_1, \dots, I'_H\}$ . In Step 3, the UE-SBS pairs compute the fractions of time allocated to each MIS in a distributed manner.

When an MIS is scheduled, the UEs in this MIS transmit at their maximum power levels, and the other UEs do not transmit. Define  $R_i^k$  as the *instantaneous* throughput obtained by UE  $i$  in the MIS  $I'_k$ , which can be calculated as  $\log_2(1 + \frac{g_{iT(i)}p_i^{I'_k}}{\sum_{r=1, r \neq i}^N g_{rT(i)}p_r^{I'_k} + \sigma_{T(i)}^2})$ , where  $p_i^{I'_k} = p_i^{max}$  if  $i \in I'_k$  and  $p_i^{I'_k} = 0$  otherwise. To determine  $R_i^k$ , the UE needs to know the total interference it experiences when transmitting in  $I'_k$ . This can be measured by having an initial cycle of transmissions of UEs in each MIS in the order of the indices of MISs/colors.

From now on, we assume that the network performance criterion  $W(\mathbf{y})$  is concave in  $\mathbf{y}$  and is separable, namely  $W(y_1, \dots, y_N) = \sum_{i=1}^N W_i(y_i)$ . Examples of separable criteria include weighted sum throughput and proportional fairness. Our framework can also deal with max-min fairness, although it is not separable (see the discussion in the Appendix). The problem of computing the optimal fractions of time for the MISs is given as follows.

### Coupled Problem (CP)

$$\begin{aligned} & \max_{\alpha} \sum_{i=1}^N W_i \left( \sum_{k=1}^H \alpha^k R_i^k \right) \\ & \text{subject to } \sum_{k=1}^H \alpha^k R_i^k \geq R_i^{\min}, \forall i \in \{1, \dots, N\} \\ & \quad \sum_{k=1}^H \alpha^k = 1, \alpha^k \geq 0, \forall k \in \{1, \dots, H\} \end{aligned}$$

Each UE  $i$  knows only its own utility function  $W_i$  and minimum throughput requirement  $R_i^{\min}$ . Hence, it cannot solve the above problem by itself. We will first reformulate the above problem into a decoupled problem and then show that the reformulated problem can be solved in a distributed manner. Let each UE  $i$  have a local estimate  $\beta_i^k$  of the fractions of time allocated to each MIS  $I'_k$  (including those MISs that UE  $i$  does not belong to). We impose an additional constraint that all the UEs' local estimates are the same. Note that this constraint will be satisfied by our solution, and is not an assumption. Such a constraint is still global, because any two UEs, even if they are not neighbors, need to have the same local estimate. Hence, global message exchange among any pair of UEs is still needed to solve this problem with local estimates and global constraints.<sup>11</sup> To avoid global message exchange, we reformulate the CP into a decoupled problem (DP) that involves only local coupling among the neighbors and can be solved with local message exchange using Alternating Direction Method of Multipliers (ADMM) [WO13].

Write  $\boldsymbol{\beta}_i = (\beta_i^1, \dots, \beta_i^H)$  as UE  $i$ 's local estimates of the fractions of time allocated to each

---

<sup>11</sup>If the UEs could exchange messages globally, i.e. broadcast messages to all the UEs in the network, and if the network performance criterion is strictly concave, we could use standard dual decomposition with augmented Lagrangian in [BT89] to derive a distributed algorithm. However, in large networks, the UEs cannot exchange messages globally with other UEs, and the network performance criterion may not be strictly concave (e.g., the weighted sum throughput is linear).

MIS, and  $\mathbf{R}_i = (R_i^1, \dots, R_i^H)$  as UE  $i$ 's throughput when each MIS is scheduled. Each UE  $i$ 's local estimates should be in the polyhedron  $\mathcal{B}_i \triangleq \{\boldsymbol{\beta}_i : \mathbf{1}^T \boldsymbol{\beta}_i = 1, \boldsymbol{\beta}_i \geq \mathbf{0}, \boldsymbol{\beta}_i^T \mathbf{R}_i \geq R_i^{min}\}$ , where  $(\cdot)^T$  is the transpose. Let  $E$  be the set of edges, where each edge  $e = \{i, j\}$  is an ordered set of the vertices,  $i < j$  that are directly connected. As we will prove in Theorem 7, in a connected interference graph<sup>12</sup>, the requirement that all UEs' local estimates are the same can be reduced to the requirement that every UE has the same local estimate as its neighbors, namely  $\boldsymbol{\beta}_i = \boldsymbol{\beta}_j$  for  $i, j$  s.t.  $\{i, j\} = e$ , where  $e \in E$ . To make the problem solvable by ADMM, we rewrite the constraints by introducing auxiliary variables  $\theta_{ei}^k$ , where  $i \in e$  is one endpoint of the edge. Then the constraint for each edge  $e = \{i, j\}$  can be rewritten as  $\beta_i^k = \theta_{ei}^k$ ,  $-\beta_j^k = \theta_{ej}^k$ ,  $\theta_{ei}^k + \theta_{ej}^k = 0$ . Hence, the auxiliary variable  $\theta_{ei}^k$  can be interpreted as  $i$ 's estimate of its neighbor  $j$ 's estimate  $\beta_j^k$ . For  $e = \{i, j\}$  define the set of the auxiliary variables  $\Theta_e^k = \{(\theta_{ei}^k, \theta_{ej}^k) \in \mathbb{R}^2 : \theta_{ei}^k + \theta_{ej}^k = 0, -1 \leq \theta_{ei}^k \leq 1, -1 \leq \theta_{ej}^k \leq 1\}$  and let  $\boldsymbol{\Theta}^k = \Pi_{e \in E} \Theta_e^k$ . Also for each edge  $e = \{i, j\}$  and for each  $k \in \{1, \dots, H\}$  define  $D_{ei}^k = 1$  and  $D_{ej}^k = -1$ . Then the decoupled problem is given as follows.

### Decoupled Problem (DP)

$$\begin{aligned} & \min_{\{\boldsymbol{\beta}_i \in \mathcal{B}_i\}_{i=1}^N, \{\boldsymbol{\theta}^k \in \boldsymbol{\Theta}^k\}_{k=1}^H} - \sum_{i=1}^N W_i (\boldsymbol{\beta}_i^T \mathbf{R}_i) \\ & \text{subject to } D_{eq}^k \beta_q^k = \theta_{eq}^k, \forall q \in e, \forall e \in E, \forall k \in \{1, \dots, H\} \end{aligned}$$

**Theorem 7** *For any connected interference graph, the coupled problem (CP) is equivalent to the decoupled problem (DP).*

The above theorem shows that the original problem (CP), which requires global information and global message exchange to solve, is transformed into an equivalent problem (DP), which as we will show, can be solved in a distributed manner with local message exchange.

We denote the optimal solution to the DP by  $\mathbf{W}_{\text{distributed}}^G$ . We associate with each constraint  $D_{eq}^k \beta_q^k = \theta_{eq}^k$  a dual variable  $\lambda_{eq}^k$ . The augmented Lagrangian for DP is

---

<sup>12</sup>A graph is connected, if any two nodes are connected by a path of edges.

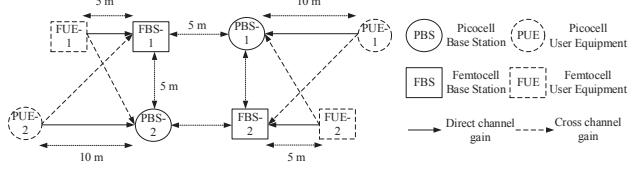


Figure 4.4: A heterogeneous network of 2 PBS and 2 FBS and their corresponding UEs.

$$L_y(\{\beta_i\}_i, \{\theta_{eq}^k\}_{k,e,q}, \{\lambda_{eq}^k\}_{k,e,q}) = - \sum_{i=1}^N W_i(\beta_i^T \mathbf{R}_i) + \sum_{k=1}^H \sum_{e \in E} \sum_{q \in e} [\lambda_{eq}^k (D_{eq}^k \beta_q^k - \theta_{eq}^k) + \frac{y}{2} (D_{eq}^k \beta_q^k - \theta_{eq}^k)^2] \quad (4.1)$$

In the ADMM procedure (see Table 4.4), each UE  $i$  solves for its optimal local estimates  $\beta_i(t)$  that maximizes the augmented Lagrangian given the previous dual variables  $\lambda_{ei}^k(t-1)$  and auxiliary variables  $\theta_{ei}^k(t-1)$ . Then it updates its dual variable  $\lambda_{ei}^k(t)$  and auxiliary variable  $\theta_{ei}^k(t)$  based on its local estimate  $\beta_i^k(t)$  and its neighbor  $j$ 's local estimate  $\beta_j^k(t)$ . This iteration of updating local estimates, dual variables, and auxiliary variables is repeated  $P$  times. Next, it is shown that this procedure will indeed converge.

**Theorem 8** *If DP is feasible<sup>13</sup>, then the ADMM algorithm in Table 4.4 converges to the optimal value  $\mathbf{W}_{distributed}^G$  with a rate of convergence  $\mathcal{O}(\frac{1}{P})$ .*

**Step 4. Determining the cycle length and transmission times:** At the end of Step 3, all the UEs have a consensus about the optimal fractions of time allocated to each MIS, namely  $\beta_i^* = \gamma^* = (\gamma_1^*, \dots, \gamma_H^*)$ ,  $\forall i \in \{1, \dots, N\}$ . The MISs transmit in the order of their indices (i.e.,  $\{1, \dots, H\}$ ) in cycles. In each cycle of transmission, MIS  $I'_k$  transmits for  $\left\lceil \frac{\gamma_k^*}{\min_{i \in 1, \dots, N} \gamma_i^*} \times 10^d \right\rceil$  slots, where we multiply by  $10^d$  such that the rounding error is reduced or eliminated in case that  $\frac{\gamma_k^*}{\min_{i \in 1, \dots, N} \gamma_i^*}$  is not an integer.

---

<sup>13</sup>If the feasible region resulting from the constraints in DP is non-empty, then DP is feasible.

#### 4.5.2 A Motivation Example

Consider a network of 2 picocell base stations (PBS) and 2 femtocell base stations (FBS), each serving one UE. The network topology is shown in Fig. 4.4. We assume a path loss model for channel gains, with path loss exponent 4. The maximum transmit power of each UE is 80 mW, and the noise power at each SBS is  $1.6 \times 10^{-3}$  mW. UEs in different tiers have different minimum throughput requirements: FUE (femtocell UE) 1 and FUE 2 in the femtocells require a minimum throughput 0.4 bits/s/Hz, and PUE (picocell UE) 1 and PUE 2 in the picocells require 0.2 bits/s/Hz. The interference graph is constructed according to a distance based threshold rule similar to [JPP05]. Specifically, an edge exists between two UE-BS pairs if the distance between any pair of SBSs is less than a threshold, which is set to be 1.2m here. There are two MISs. MIS 1 consists of FUE 1 and FUE 2, and MIS 2 consists of PUE 1 and PUE 2. We consider two performance criteria: the max-min fairness and the sum throughput. We will compare with the following state-of-the-art policies:

- 1. Distributed Constant Power Control Policies** [HBH06, SMG01, HYC09]: In these policies, all the UEs choose *constant* power levels determined by distributed algorithms utilizing information (e.g., power levels used by neighbors) made available through local/global message exchange.
- 2. Optimal Centralized Constant Power Policies:** In these policies, all the UEs choose *constant* power levels determined by a central controller utilizing global information.
- 3. Distributed MIS STDMA-1** [ET90] and **STDMA-2** [RP89]: These policies construct a subset of the MISs of the interference graph in a distributed manner and propose fixed schedules of the MISs. Different works adopt different schedules, and we differentiate them by referring to them as MIS STDMA-1 [ET90] and STDMA-2 [RP89].
- 4. Distributed Joint Power Control and Spatial Reuse** [KG08] [GKG07]: These policies choose one UE from each cell to form a subset, and schedule these subsets of UEs based on their channel gains to maximize the sum throughput. The policies are named power matched scheduling (PMS).

In Table 4.1, we compare the performance of our proposed policy with state-of-the-art

policies for the same setup as in Fig. 4.1. We compute the optimal centralized constant power control policy by exhaustive search, which serves as the performance upper bound of the distributed constant power control policies [HBH06,SMG01,HYC09] centralized constant power control policies [CTP07]. In PMS policies [KG08] [GKG07], UEs within the same cell are scheduled in a time-division multiple access (TDMA) fashion, and the active UEs in different cells transmit simultaneously. In this motivating example, there is one UE in each cell, which will be scheduled to transmit all the time. Therefore, the PMS policy reduces to a constant power control policy, and is worse than the optimal centralized constant power control policy. We can see that our proposed policy outperforms all constant power control policies and distributed PMS policies by at least 375% and 32.8%, in terms of max-min fairness and sum throughput, respectively. The significant performance improvement over the constant power control policies results from the elimination of the high interference among the users through scheduling MISs. Our proposed policy also outperforms distributed STDMA policies by 30%-40%. As we will see in Section 4.6, the performance gain is even higher (160%-700%) in realistic deployment scenarios. Finally, in this motivating example, the proposed policy achieves the optimal performance of the benchmark problem defined in Section 4.6, which is a close approximation of the original problem (PDP).

#### 4.5.3 Performance Guarantees for Large Networks and Properties of Interference Graphs

In this section, we provide performance guarantees for our proposed framework described in Section 4.5.1. Specifically, we prove that the network performance  $\mathbf{W}_{\text{distributed}}^G$  achieved by the proposed distributed algorithms has a constant competitive ratio with respect to the optimal value  $\mathbf{W}_{\text{opt}}$  of the PDP. Moreover, we prove that the competitive ratio does not depend on the network size. Our result is strong, because the solution to PDP needs to be computed by a centralized controller with global information, while our proposed framework allows the UEs to compute the policy fast in a distributed manner with local information and local message exchange.

Table 4.1: Comparisons in terms of max-min fairness & sum throughput criterion.

Policies	Max-min throughput (bits/s/Hz)	Performance Gain %	Sum throughput (bits/s/Hz)	Performance Gain %
Distributed constant power control [HBH06, SMG01, HYC09]	<0.28	>375 %	6.1	32.8 %
Distributed PMS [KG08, GKG07]	<0.28	>375%	6.1	32.8 %
Optimal centralized constant power control	0.28	375%	6.1	32.8 %
Distributed MIS STDMA-2/1 [RP89, ET90]	0.96	38.5%	6.25	30.0 %
Proposed (Section- 4.5)	1.33	-	8.12	-
Benchmark Problem (BP)	1.33	-	8.12	-

Before characterizing the competitive ratio analytically, we define some auxiliary variables. Define the upper and lower bounds on the UEs' maximum transmit power levels and throughput requirements as,  $0 < p_{lb}^{max} \leq p_i^{max} \leq p_{ub}^{max}, \forall i \in \{1, \dots, N\}$  and,  $0 < R_{lb}^{min} \leq R_i^{min} \leq R_{ub}^{min}, \forall i \in \{1, \dots, N\}$  respectively. Let  $D_{ij}$  be the distance between UE  $i$  and SBS  $j$ . Define upper and lower bounds on the distance between any UE and its serving SBS and the noise power at the SBSs as,  $0 < D^{lb} \leq D_{iT(i)} \leq D^{ub}, \forall i \in \{1, \dots, N\}$  and,  $\sigma_{lb}^2 \leq \sigma_j^2 \leq \sigma_{ub}^2, \forall j \in \{1, \dots, K\}$  respectively. We assume that the channel gain is  $g_{ij} = \frac{1}{(D_{ij})^{np}}$ , where  $np$  is the path loss exponent.

**Definition 1 (Weak Non-neighboring Interference):** The interference graph  $G$  exhibits  $\zeta$  Weak Non-neighboring Interference ( $\zeta$ -WNI) if for each UE  $i$  the maximum interference from its non-neighbors is bounded, namely  $\sum_{j \notin \mathcal{N}(i), j \neq i} g_{jT(i)} p_j^{max} \leq (2^\zeta - 1) \sigma_{ub}^2, \forall i \in \{1, \dots, N\}$ .

Define  $\Delta^{max} = \frac{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})}{R_{ub}^{min}} - 1$ . Then we have the following theorem for the network performance criterion, sum throughput.<sup>14</sup>

---

<sup>14</sup>We can extend this result for weighted sum throughput, with weights  $w_i = \Theta(\frac{1}{N})$ , it is not done to avoid complex notations.

**Theorem 9** For any connected interference graph, if the maximum degree  $\Delta \leq \Delta^{max}$  and it exhibits  $\zeta$ -WNI then, our proposed framework of interference management described in Section 4.5.1 achieves a performance  $\mathbf{W}_{distributed}^G \geq \Gamma \cdot \mathbf{W}_{opt}$  with a probability no smaller than  $(1 - \frac{1}{N^{c_1-1}})(1 - \frac{1}{N^{c_2-1}})$ . Moreover, the competitive ratio  $\Gamma = \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np}\sigma_{lb}^2})}$  is independent of the network size.

Note that the analytical expression of competitive ratio,  $\Gamma = \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np}\sigma_{lb}^2})}$ , does not depend on the size of the network. Our results are derived under the conditions that the interference graph has a maximum degree bounded by  $\Delta^{max}$ , and that the interference from non-neighbors is bounded (i.e.  $\zeta$ -WNI). These conditions do not restrict the size of the network. The next example illustrates this. In addition, our results hold for any interference graph that satisfy the conditions in Theorem 9, regardless of how the graph is constructed.

*Example:* Consider a layout of SBSs in a  $K \times K$  square grid, i.e.  $K^2$  SBSs with a distance of 5m between the nearest SBSs. Assume that each UE is located vertically below its SBS at a distance of 1m. Fix the parameters  $p_i^{max} = 100$  mW,  $\sigma_i^2 = 3$  mW,  $R_i^{min} = 0.1$  bits/s/Hz,  $\forall i \in \{1, \dots, K^2\}$ ,  $np = 4$ . We construct the interference graph based on the distance rule [JPP05], namely there is an edge between two pairs if the distance between their SBSs exceeds 6m, which gives us the maximum degree  $\Delta = 4$ . We can also verify that the interference graphs under any number  $K^2$  of SBSs exhibit  $\zeta$ -WNI with  $\zeta = 0.15$  and  $\Delta < \Delta^{max}$ , where  $\Delta^{max} = 48$ . Given  $\Delta = 4$  and  $\zeta = 0.15$ , from Theorem 9, we get the performance guarantee of 0.17 for any network size  $K^2$ . Note that the number 0.17 is a performance guarantee, and that the actual performance is much higher compared to the performance guarantee as well as those achieved by state-of-the-art policies (see Section 4.6).

Both Theorem 6 and 9 restricted the interference graph to be bounded-degree. We justify our restriction by showing that the bounded-degree property is necessary to fulfill the minimum throughput requirements of each UE. Specifically, we prove that if the maximum degree exceeds some threshold, then no MIS based policy in  $\Pi^{MIS}$  (which is a large policy space) is feasible. Suppose that the interference graph is constructed based on a distance based threshold rule similar to [JPP05]: an edge exists between two UE-SBS pairs if and

only if the distance between two SBSs is no greater than  $D^{th}$ . We define the threshold of the maximum degree as  $\Delta^*$  (See the Appendix for the expression).

**Theorem 10** *If the maximum degree of the interference graph  $\Delta \geq \Delta^*$ , then any MIS based policy in  $\Pi^{MIS}$  fails to satisfy the minimum throughput requirements of the UEs.*

The intuition behind Theorem 10 is that, if the degree of the interference graph is large then there must be a large number of UE-SBS pairs which interfere with each other strongly, which makes it impossible to allocate each UE enough transmission time to satisfy their minimum throughput requirements simultaneously.

#### 4.5.4 Self-Adjusting Mechanism for Dynamic Entry/Exit of UEs

We now describe how the proposed framework can adjust to dynamic entry/exit by the UEs in the network without recomputing all the four steps. We allow the UEs to enter and exit, but the number of SBSs is fixed. We only let one UE enter or leave the network in any time slot.

1. *UE leaves the network:* Suppose UE  $i$ , which was transmitting to SBS  $T(i)$ , leaves the network. If the UE  $i$  was transmitting in a set of colors  $C_i$ , then as soon as it leaves, these colors can be potentially used by some neighbors,  $\mathcal{N}(i)$ . SBS  $T(i)$  can still be serving other UEs which are still in the network and transmitting. Then for each color  $c' \in C_i$  it first searches among these UEs that are not already transmitting in  $c'$  and who also do not have a neighboring UE-SBS pair which is already transmitting in  $c'$ . Let the set of such UEs be  $UE_{i,left}^{c'}$ . SBS  $T(i)$  allocates color  $c'$  to the UE whose index is  $\arg \max_{j \in UE_{i,left}^{c'}} R_j^{c'}$ . In case  $UE_{i,left}^{c'}$  is empty then that color,  $c'$  is left unused.

2. *UE enters the network:* Suppose a UE  $i$  registered with SBS  $T(i)$  enters the network. SBS  $T(i)$  creates the list of colors  $C_{i,enter}^{valid}$ , which are either unused or the UEs transmitting in the colors are transmitting at more than their minimum throughput requirement. SBS  $T(i)$  allocates some portions from the fractions of time allocated to the colors in  $C_{i,enter}^{valid}$ , to satisfy UE- $i$ 's throughput requirement to the best possible extent, making sure that the minimum

throughput requirements of UEs transmitting to SBS  $T(i)$  in  $C_{i,enter}^{valid}$  are not violated. If the requirement of UE- $i$  is not satisfied then, SBS  $T(i)$  requests the neighboring UE-SBSs to announce the set of colors, which are either not being used or in which the UEs being served are operating at more than their throughput requirement. From the list of colors received,  $T(i)$  chooses those in which UE  $i$  can transmit without conflicting with neighbors. For each of these colors it sends the request (portion of time needed) to the neighbors. SBS  $T(i)$  and the neighbors go through a phase of communication (more detail in the Appendix), based on which SBS  $T(i)$  can decide how much time UE- $i$  can transmit in these colors.

#### 4.5.5 Extensions

In our model, UEs operate in the same frequency band. However, our methodology can be extended to scenarios where UEs operate in different frequency channels (frequency reuse) and transmit at the same time. In this case, the problem is to find the optimal frequency allocation with the same objective function and constraints as in PDP. To solve this problem, the first two steps of the framework remain the same. In Step 3, the UEs compute distributedly the optimal fractions of *bandwidth* to be allocated to each MIS. This step is equivalent to computing the optimal fraction of time allocated to each MIS as in our current formulation. In Step 4, the UEs compute the number of frequency channels allocated to each MIS based on the bandwidth allocation.

Note that we do not implement beamforming, although beamforming can be used in conjunction with our policy. If the UEs transmitting to the same SBS cooperate to do beamforming, we can delete the edge between them in the interference graph, and use the new interference graph in the scenario with beamforming.

## 4.6 Illustrative Results

In this section, we evaluate our proposed policy under a variety of scenarios with different levels of interference, large numbers of UEs, different performance criteria, time-varying channel conditions, and dynamic entry and exit of UEs.

We compare our policy with the optimal centralized constant power control policy, the distributed MIS STDMA-1 [ET90] and STDMA-2 [RP89], distributed PMS [KG08] [GKG07], in terms of sum throughput and max-min fairness. We do not separately compare with distributed/centralized constant power control policies in [HBH06,SMG01,HYC09] [CTP07], because their performance is upper bounded by the optimal centralized power control. Since it is difficult to compute the solution to the NP-hard PDP, we define a benchmark problem, where we restrict our search to policies in which a UE either transmits at its maximum power level or does not transmit. The space of such policies can be written as  $\Pi_{BC} = \{\boldsymbol{\pi} = (\pi_1, \dots, \pi_N) : \pi_i : \mathbb{Z}_+ \rightarrow \{0, p_i^{max}\} \forall i \in \{1, \dots, N\}\}$ . The policy space  $\Pi_{BC}$  is a subset of all policies  $\Pi$  and is a superset of MIS based policies  $\Pi^{MIS}$ . In other words, the benchmark problem has the same objective and constraints as PDP; the only difference is the policy space to search. Hence, the benchmark problem is a close approximation of the PDP. Note that the benchmark problem is also NP-hard (See the Appendix).

#### 4.6.1 Performance under Time-Varying Channel Conditions

Consider a 3x3 square grid of 9 SBSs with the minimum distance between any two SBSs being  $d = 4.7m$ . Each SBS serves one UE, who has a maximum power of 1000 mW and a minimum throughput requirement of 0.45 bits/s/Hz. The UEs and the SBSs are in two parallel horizontal hyperplanes, and each SBS is vertically above its UE with a distance of  $\sqrt{10}m$ . Then the distance from UE  $i$  to another SBS  $j$  is  $D_{ij} = \sqrt{10 + (D_{ij}^{BS})^2}$ , where  $D_{ij}^{BS}$  is the distance between SBSs  $i$  and  $j$ . The channel gain from UE  $i$  to SBS  $j$  is a product of path loss and Rayleigh fading  $f_{ij} \sim Rayleigh(\beta)$ , namely  $g_{ij} = \frac{1}{(D_{ij})^2} f_{ij}$ . The density function of  $Rayleigh(\beta)$  is  $v(z) = \frac{z}{\beta^2} e^{-\frac{z^2}{2\beta^2}}$  for  $z \geq 0$ , and  $v(z) = 0$  for  $z < 0$ . The SBSs identify neighbors using a distance based rule with the threshold distance as in Section 4.5.3 with  $D^{th} = 7m$ . Note that different thresholds lead to different interference graphs, and hence different performance, which will be discussed next. Although we use a distance based threshold rule, our framework is general and does not rely on a particular rule. The resulting interference graph for this setting is graph 3 shown in Fig. 4.7 a).

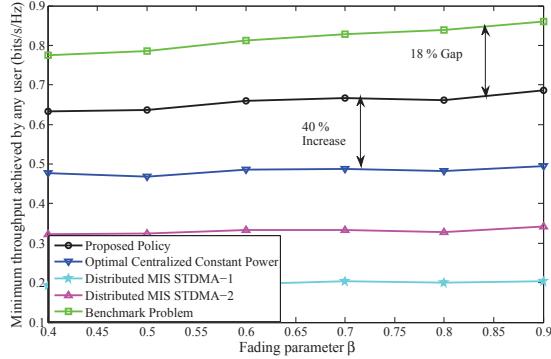


Fig. 4.5 a)

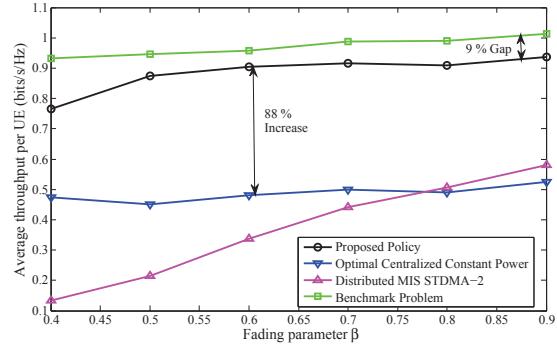


Fig. 4.5 b)

Figure 4.5: Comparison of the proposed policy with state of the art under different interference strength and time-varying channel conditions.

At the beginning, the UE-SBS pairs generate the set of MISs (Step 2 of the design framework in Section 4.5), and compute the optimal fractions of time allocated to each MIS (Step 3). In our simulation, we assume a block fading model [Gol05] and the fading changes every 100 time slots independently. To reduce complexity, the UEs do not recompute the interference graph and the MISs, but will recompute the optimal fractions of time under the new channel gains every 100 time slots. In Fig. 4.5, we compare the performance of the proposed policy with state of the art policies under different variances  $\beta$  of Rayleigh fading. We do not plot the performance of distributed PMS for this scenario since it is upper bounded by optimal centralized constant power control (because there is one UE per cell). We do not plot the distributed MIS STDMA -1 either, when the performance criterion is average throughput per UE (i.e.,  $\frac{\text{sum throughput}}{N}$ ), because it cannot satisfy the minimum throughput constraints. From Fig. 4.5, we can see that in terms of both average throughput and max-min fairness, our proposed policy achieves large performance gain (up to 88%) over existing policies, and achieves performance close to the benchmark (as close as 9%).

*Selecting the Optimal Interference Graph :* For different values of  $d$ , there can be five possible interference graphs, which are shown in Fig. 4.7 a). In Fig. 4.6 a) we show that as the grid size  $d$  decreases ( $d = 4.7\text{m}$ ,  $d = 3.7\text{m}$  and  $d = 2.5\text{m}$ ), the levels of interference from the adjacent UEs increases, and as a result, the interference graph with higher degrees perform better (as  $d$  decreases, the optimal graph changes from graph 3 to graph 1).

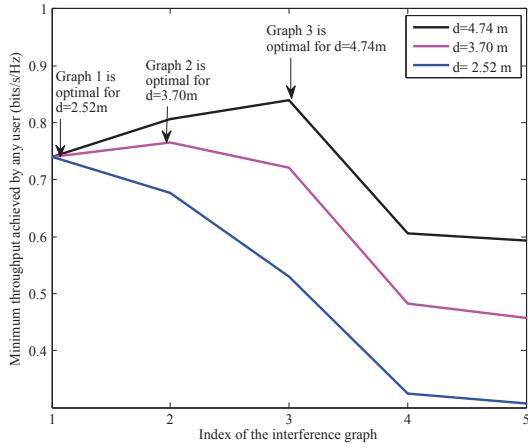


Fig. 4.6 a)

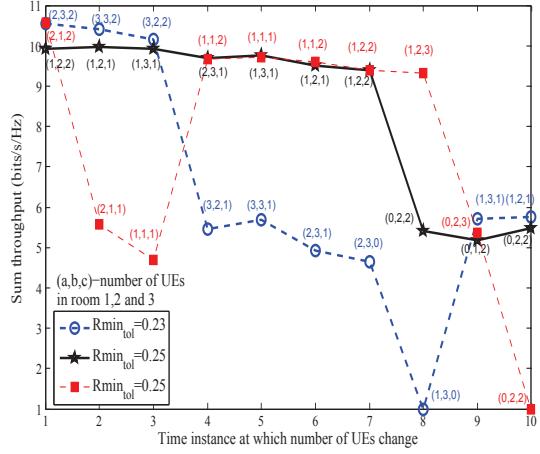


Fig. 4.6 b)

Figure 4.6: a) Comparison of max-min fairness under different grid sizes, b) Sample paths of sum throughput under dynamic entry/exit of UEs in the network.

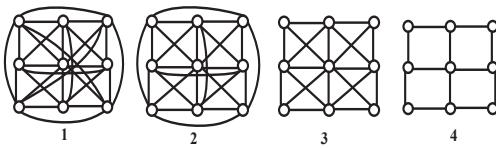


Fig. 4.7 a)

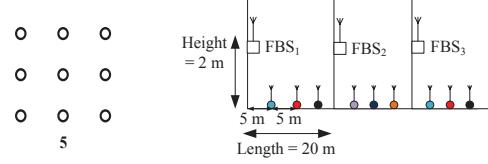


Fig. 4.7 b)

Figure 4.7: a) Different interference graphs for the  $3 \times 3$  BS grid, b) Illustration of setup with 3 rooms.

#### 4.6.2 Performance Scaling in Large Networks

Consider the uplink of a femtocell network in a building with 12 rooms adjacent to each other. Fig. 4.7 b) illustrates 3 of the 12 rooms with 5 UEs in each room. For simplicity, we consider a 2-dimensional geometry. Each room has a length of 20 meters. In each room, there are  $P$  uniformly spaced UEs, and one SBS installed on the left wall of the room at a height of 2m. The distance from the left wall to the first UE, as well as the distance between two adjacent UEs in a room, is  $\frac{20}{(1+P)}$  meters. Based on the path loss model in [SR92], the channel gain from each SBS  $i$  to a UE  $j$  is  $\frac{1}{(D_{ij})^2 \Delta^{n_{ij}}}$ , where  $\Delta = 10^{0.25}$  is the coefficient representing the loss from the wall, and  $n_{ij}$  is the number of walls between UE  $i$  and SBS  $j$ . Each UE has a maximum transmit power level of 50 mW, a minimum throughput requirement of  $R_i^{min} = 0.025$  bits/s/Hz, and a noise power level of  $10^{-11}$ mW at its receiver. Here, we

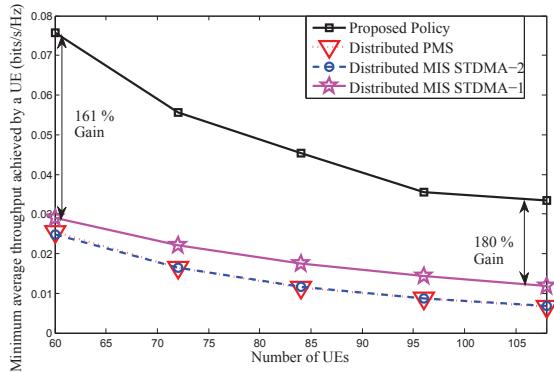


Fig. 4.8 a)

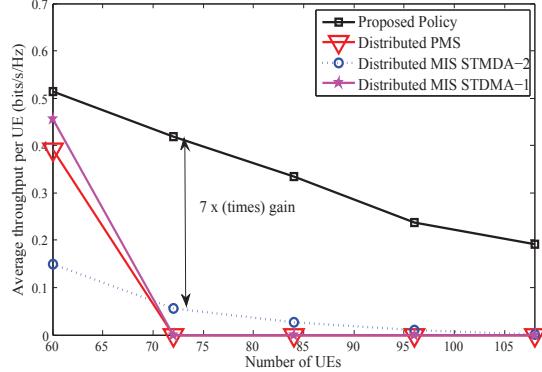


Fig. 4.8 b)

Figure 4.8: Comparison of max-min fairness and average throughput per UE against state of the art for large networks.

consider that the UEs use a distance based threshold rule as in Section 4.5.2 with  $D^{th} = 30$  m. This results in interference graphs which connects all the UE-SBS pairs within the room and in the adjacent rooms. We vary the number  $P$  of UEs in each room from 5 to 9 and compare the performance in Fig. 4.8. Note that the optimal centralized constant power policy cannot satisfy the feasibility conditions for any number of UEs in each room. Hence, only the performance of distributed MIS STDMA-1,2 and distributed PMS is shown in Fig. 4.8. We can see that under both criteria, the performance gain of our proposed policy is significant (from 160% to 700%). Note that since the number of UEs is large, solving the benchmark problem (which is NP-hard) requires enormous computational power.

#### 4.6.3 Self-adjusting Mechanism for Dynamic Entry/Exit of the UEs

The self-adjusting mechanism proposed in Section 4.5.4 is aimed to provide incoming UEs with the maximum possible throughput without affecting the incumbent UEs, and to reuse the time slots left vacant by exiting UEs efficiently. Consider the same setup as in Section 4.6.2 with 3 rooms and a maximum of  $P = 3$  UEs in each room. Each UE has a maximum transmit power of 1000 mW and a minimum throughput requirement of 0.25 bits/s/Hz.

In Fig. 4.6 b) we show different sample paths of the sum throughput under different entry and exit processes. In the legends (i.e.,  $R_{min,tol}$ ), we show the minimum throughput

achieved at any point in the sample path. We repeated the same procedure 100 times. We can see that the self-adjusting mechanism works well by guaranteeing a worst-case minimum throughput requirement of 0.23 bits/s/Hz, which is just 0.02 bits/s/Hz below the original requirement more than 80% of the time.

## 4.7 Discussion on the Generality of the Framework

The framework described in the previous sections focused on the application of interference management in wireless networks. In the framework that we described, each UE's utility for one time slot  $t$  was defined as the Shannon capacity  $r_i(\mathbf{p}^t)$ . In this section, we make no restrictions on the functional form of  $r_i(\mathbf{p}^t)$ . For a general framework, we consider a problem with  $N$  users and each user's utility for taking an action  $p_i^t \in \mathcal{P}_i$  in time slot  $t$  is defined as  $r_i(\mathbf{p}^t)$  and the long-term utility is defined as  $R_i(\boldsymbol{\pi})$ , where  $\boldsymbol{\pi}$  is the joint resource sharing policy. We define a general policy design problem as follows.

### General Policy Design Problem (GPDP)

$$\begin{aligned} \max_{\boldsymbol{\pi}} & W(R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi})) \\ \text{subject to} & R_i(\boldsymbol{\pi}) \geq R_i^{\min}, \forall i \in \{1, \dots, N\} \end{aligned}$$

Next, we describe the design framework for the above GPDP.

The framework follows the very same steps as the framework proposed earlier for PDP problem.

**Step 1. Identification of the interfering neighbors:** The interfering neighbors of a user  $i$  depend on the utility function for each user  $i$ . In the PDP case, the identification was carried out by a centralized entity using certain protocols. In a general case, we can analyze the utility functions  $r_i(\mathbf{p})$  of all the pairs of users. If the worst case utility of a user when the other user is using the resource to the maximum is below a certain threshold, then we assume that there is an edge between the two users in the interference graph.

Once we have the interference graph the remaining steps Step 2-4 are exactly the same. Theorems 6, 7, 8 continue to hold for this case. Next, we discuss how Theorem 9 can be

adapted for this case. In Theorem 9, there are two conditions that need to be satisfied to prove that the proposed distributed resource sharing policy is a constant factor approximation.

Consider a user who is using the resource to the maximum and all the non-neighbors are also using the resource to the maximum limit possible. In such a case, the minimum utility attained by the user is defined as  $R_{nb\_max}^{min}$ . Define  $\Delta^{max} = \frac{R_{nb\_max}^{min}}{R_{ub}^{min}} - 1$ . Suppose the maximum utility that a user can achieve as  $R^{max}$ . The modified Theorem 9 adapted to this setting is given below.

**Theorem 11** *For any connected interference graph, if the maximum degree  $\Delta \leq \Delta^{max}$ , our proposed framework achieves a performance  $\mathbf{W}_{distributed}^G \geq \Gamma \cdot \mathbf{W}_{opt}$  with a probability no smaller than  $(1 - \frac{1}{N^{c_1-1}})(1 - \frac{1}{N^{c_2-1}})$ . Moreover, the competitive ratio  $\Gamma = \frac{R_{ub}^{min}}{R^{max}}$  is independent of the network size.*

The proof of the above follows from the proof of Theorem 9 given in the Appendix.

## 4.8 Conclusion

In this chapter, we proposed a novel and systematic method for distributed resource sharing. We mainly focused on the problem of interference management but the framework is general and can be useful in other applications as discussed in Section 4.7. Our framework allows each UE to have only local knowledge about the network and communicate only with its interfering neighbors. There are two key steps in our framework. First, we propose a novel distributed algorithm for the UEs to generate a set of MISs that span all the UEs. The distributed algorithm for generating MISs requires  $\mathcal{O}(\log N)$  steps (which is much faster than state-of-the-art) before it converges to the set of MISs with a high probability. Second, we reformulate the problem of determining the optimal fractions of time allocated to the MISs in a novel manner such that the optimal solution can be determined by a distributed algorithm based on ADMM. Importantly, we prove that under wide range of conditions, the proposed policy can achieve a constant competitive ratio with respect to the policy

design problem which is NP-hard. Moreover, we show that our framework can adjust to UEs entering or leaving the network. Our simulation results show that the proposed policy can achieve large performance gains (160%-700%).

## 4.9 Appendix

### 4.9.1 Appendix A

**Discussion on max-min fairness:** We now discuss as to how the proposed framework can be extended to incorporate inseparable function like max-min fairness. The coupled problem with max-min fairness objective is restated below:

$$\begin{aligned}
 \textbf{Coupled Problem (CP)} \quad & \max_{\boldsymbol{\alpha}} \quad \min_{i \in \{1, \dots, N\}} W_i \left( \sum_{k=1}^H \alpha_k R_i^k \right) \\
 & \text{subject to} \quad \sum_{k=1}^H \alpha_k R_i^k \geq R_i^{\min}, \quad \forall i \in \{1, \dots, N\} \\
 & \quad \sum_{k=1}^H \alpha_k = 1, \quad \alpha_k \geq 0, \quad \forall k \in \{1, \dots, H\}
 \end{aligned}$$

Transforming the above problem into an equivalent problem with auxiliary variable  $t$  is given as

$$\begin{aligned}
 & \max_{\boldsymbol{\alpha}, t} \quad t \\
 & \text{subject to} \quad W_i \left( \sum_{k=1}^H \alpha_k R_i^k \right) \geq t, \quad \forall i \in \{1, \dots, N\} \\
 & \quad \sum_{k=1}^H \alpha_k R_i^k \geq R_i^{\min}, \quad \forall i \in \{1, \dots, N\} \\
 & \quad \sum_{k=1}^H \alpha_k = 1, \quad \alpha_k \geq 0, \quad \forall k \in \{1, \dots, H\}
 \end{aligned}$$

To decouple the above problem, we introduce local variables for each UE  $i$  given as,  $\{\beta_i^1, \dots, \beta_i^{H+1}\}$ . Now we state a problem which we claim is equivalent to CP, (the proof to

Table 4.2: Generating MIs in a distributed manner, algorithm for UE  $i$ .

<p><b>Phase 1- Initialization:</b> <math>\text{Tx}_{\text{tent}}^i = \phi</math>, <math>\text{Tx}_{\text{final}}^i = \phi</math>, tentative and final choice of UE <math>i</math>,  <math>\text{Rx}_{\text{tent}}^{\mathcal{N}(i)} = \phi</math>, <math>\text{Rx}_{\text{final}}^{\mathcal{N}(i)} = \phi</math> tentative and final choice made by the neighbors,  <math>C_i^0 = \{1, \dots, H\} \cap \{1, \dots, d_i + 1\}</math> the current list of subset of available colors,  <math>C_i = \phi</math>, list of colors used by <math>i</math>, <math>F_{\text{colored}}^i = \phi</math>, <math>C1_i^0 = \{1, \dots, H\}</math>, the current list of all available colors</p>
<pre> for <math>n = 0</math> to <math>\lceil c_1 \log_{\frac{4}{3}} N \rceil</math>     <math>\text{Tx}_{\text{tent}}^i = \phi</math>, <math>\text{Tx}_{\text{final}}^i = \phi</math>     if(<math>F_{\text{colored}}^i = \phi</math>)         <math>\text{Tx}_{\text{tent}}^i = \text{rand}\{C_i^n\}</math>, rand randomly selects a color and informs the neighbors         <math>\text{Rx}_{\text{tent}}^{\mathcal{N}(i)} = \{\text{Tx}_{\text{tent}}^k, \forall k \in \mathcal{N}(i)\}</math>         If(<math>\text{Tx}_{\text{tent}}^i \neq \text{Rx}_{\text{tent}}^{\mathcal{N}(i)}(j)</math>, <math>\forall j \in \mathcal{N}(i)</math>), UE-<math>i</math> checks if there is a conflict with any of the neighbor's choice         <math>\text{Tx}_{\text{final}}^i = \text{Tx}_{\text{tent}}^i</math>, <math>C_i = \{\text{Tx}_{\text{final}}^i\}</math>, if there is no conflict then UE-<math>i</math> transmits its final color choice to the neighbors,     else         <math>\text{Tx}_{\text{final}}^i = \phi</math>     end end <math>\text{Rx}_{\text{final}}^{\mathcal{N}(i)} = \{\text{Tx}_{\text{final}}^k, \forall k \in \mathcal{N}(i)\}</math> <math>C_i^{n+1} = C_i^n \cap \{\text{Rx}_{\text{final}}^{\mathcal{N}(i)} \cup \text{Tx}_{\text{final}}^i\}^c</math>, <math>C1_i^{n+1} = C1_i^n \cap \{\text{Rx}_{\text{final}}^{\mathcal{N}(i)} \cup \text{Tx}_{\text{final}}^i\}^c</math> if(<math>\text{Tx}_{\text{final}}^i \neq \phi</math>)     <math>F_{\text{colored}}^i = 1</math> end end </pre>

Table 4.3: Phase 2 of the distributed MIS generation.

<p><b>Phase 2-Initialization:</b> <math>\text{Tx}_{\text{tent},i}^{\text{set}} = \phi, \text{Tx}_{\text{final},i}^{\text{set}} = \phi</math>, the set of tentative and final colors chosen by <math>\text{Rx}_{\text{final},i}^{\text{set}} = \phi</math>, the set of tentative and final colors chosen that are received from the neighbors, <math>x = \frac{1}{1-(c)^H(1-c)^H}</math></p>
<p>for <math>n = \lceil c_1 \log_{\frac{4}{3}} N \rceil + 1</math> to <math>\lceil c_1 \log_{\frac{4}{3}} N \rceil + \lceil c_2 \log_x N \rceil + 1</math></p>
<p>    <math>\text{Tx}_{\text{tent},i}^{\text{set}} = \phi, \text{Tx}_{\text{final},i}^{\text{set}} = \phi</math>,</p>
<p>    for <math>m = 1</math> to <math> C1_i^n </math></p>
<p>        with probability <math>c</math>, <math>\text{Tx}_{\text{tent},i}^{\text{set}}(m) = C1_i^n(m)</math>, randomly selecting and informing the neighbors about tentative choice with probability <math>1 - c</math>, <math>\text{Tx}_{\text{tent},i}^{\text{set}}(m) = \phi</math></p>
<p>    end</p>
<p>    <math>\text{Rx}_{\text{tent},i}^{\text{set}} = \cup_{k \in \mathcal{N}(i)} \text{Tx}_{\text{tent},k}^{\text{set}}</math>, set of tentative color choices of the neighbors of <math>i</math></p>
<p>    for <math>r = 1</math> to <math> \text{Tx}_{\text{tent},i}^{\text{set}} </math></p>
<p>        If(<math>\text{Tx}_{\text{tent},i}^{\text{set}}(r) \neq \text{Rx}_{\text{tent},i}^{\text{set}}(j) \forall j \in \mathcal{N}(i)</math>)</p>
<p>            <math>\text{Tx}_{\text{final},i}^{\text{set}}(r) = \text{Tx}_{\text{tent},i}^{\text{set}}(r)</math></p>
<p>        else</p>
<p>            <math>\text{Tx}_{\text{final},i}^{\text{set}}(r) = \phi</math></p>
<p>        end</p>
<p>    <math>C_i = C_i \cup \text{Tx}_{\text{final},i}^{\text{set}}</math></p>
<p>    <math>\text{Rx}_{\text{final},i}^{\text{set}} = \cup_{k \in \mathcal{N}(i)} \text{Tx}_{\text{final},k}^{\text{set}}</math>, set of final color choices of the neighbors of <math>i</math></p>
<p>    <math>C1_i^{n+1} = C1_i^n \cap \{\text{Rx}_{\text{final},i}^{\text{set}} \cup \text{Tx}_{\text{final},i}^{\text{set}}\}^c</math></p>
<p>end</p>

Table 4.4: ADMM update algorithm for UE  $i$ .

Initialization: arbitrary $\beta_i(0) \in \mathcal{B}_i$ , $\theta_{ei}^k(0)$ such that $\theta^k \in \Theta^k$ , and $\lambda_{ei}^k(0) = 0$ , $\forall k \in \{1, \dots, H\}$ , $\forall e$ such that $i \in e$
For $t = 0$ to $P - 1$ $\beta_i(t + 1) = \arg \min_{\beta_i \in \mathcal{B}_i} - \sum_{i=1}^N W_i(\beta_i^T \mathbf{R}_i)$ $+ \sum_{k=1}^H \sum_{e \in E} \sum_{q \in e} \left[ \lambda_{eq}^k (D_{eq}^k \beta_q^k - \theta_{eq}^k) + \frac{y}{2} (D_{eq}^k \beta_q^k - \theta_{eq}^k)^2 \right]$ $\beta_i(t + 1)$ is transmitted to all of its neighbors in $\mathcal{N}(i)$ . $\lambda_{ei}^k(t)$ is transmitted to its neighbor connected with edge $e$ , $\forall k \in \{1, \dots, H\}$ and $\forall e$ such that $i \in e$ Update $\forall k \in \{1, \dots, H\}$ and $\forall e$ such that $i \in e$ $\lambda_{ei}^k(t + 1) = \frac{1}{2} (\lambda_{ei}^k(t) + \lambda_{ej}^k(t)) - \frac{y}{2} (D_{ei}^k \beta_i^k(t + 1) + D_{ej}^k \beta_j^k(t + 1)),$ where $j$ is the other endpoint of $e$ . $\theta_{ei}^k(t + 1) = \frac{1}{y} (\lambda_{ei}^k(t + 1) - \lambda_{e,i}^k(t)) + D_{ei}^k \beta_i^k(t + 1)$ end

this claim is very similar to the proof of Theorem 7 and we will highlight this fact in the proof clearly).

$$\begin{aligned}
\mathbf{P1} \quad & \max_{\boldsymbol{\beta}} \quad \sum_{i=1}^N \beta_i^{H+1} \\
\text{subject to} \quad & W_i \left( \sum_{k=1}^H \beta_i^k R_i^k \right) \geq \beta_i^{H+1}, \forall i \in \{1, \dots, N\} \\
& \sum_{k=1}^H \beta_i^k R_i^k \geq R_i^{\min}, \forall i \in \{1, \dots, N\} \\
& \sum_{k=1}^H \beta_i^k = 1, \beta_i^k \geq 0, \forall k \in \{1, \dots, H\}, \forall i \in \{1, \dots, N\} \\
& \beta_i^k = \beta_j^k, \forall j \in \mathcal{N}(i), \forall k \in \{1, \dots, H+1\}
\end{aligned}$$

Here,  $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N)$ , with  $\boldsymbol{\beta}_i = (\beta_i^1, \dots, \beta_i^{H+1})$ ,  $\forall i \in \{1, \dots, N\}$ . Now, given the two problems CP and the problem P1 are equivalent, we focus on solving P1. P1 can be changed to a problem similar to DP. To do that we introduce some additional variables similar to the ones introduced for DP. If UE  $i$  and  $l$  are connected by an edge  $(i, l)$  then for each set  $I'_k$  define  $\theta_{(i,l)i}^k = \beta_i^k$  and  $\theta_{(i,l)l}^k = -\beta_l^k$ , note that these auxiliary variables are introduced to formulate the problem into the ADMM framework [WO13]. Define a polyhedron for each  $i$ ,  $\mathcal{T}'_i = \{(\boldsymbol{\beta}\mathbf{1})_i | \text{s.t. } \mathbf{1}^t(\boldsymbol{\beta}'') = \mathbf{1}, (\boldsymbol{\beta}\mathbf{1})_i \geq \mathbf{0}, \mathbf{R}'_i(\boldsymbol{\beta}'') \geq R_i^{\min}, W_i(\mathbf{R}'_i(\boldsymbol{\beta}'')) - \beta_i^{H+1} \geq 0\}$ , here  $\boldsymbol{\beta}'' = (\beta_i^1, \dots, \beta_i^H)$  and  $\mathbf{R}_i = (R_i^1, \dots, R_i^H)$  and  $()'$  corresponds to the transpose. Let  $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N) \in \mathcal{T}'$ , where  $\mathcal{T}' = \prod_{i=1}^N \mathcal{T}'_i$  and  $\prod$  corresponds to the Cartesian product of the sets. Also, let  $\boldsymbol{\beta}^k = (\beta_1^k, \dots, \beta_N^k)$ ,  $\forall k \in \{1, \dots, H\}$ . Define another polyhedron  $\Theta_{(i,l)}^k = \{(\theta_{(i,l)i}^k, \theta_{(i,l)l}^k) : \theta_{(i,l)i}^k + \theta_{(i,l)l}^k = 0, -1 \leq \theta_{(i,l)s}^k \leq 1, \forall s \in \{i, l\}\}$ ,  $\boldsymbol{\Theta}^k = \prod_{(i,l) \in E} \Theta_{(i,l)}^k$  here  $E = (e_1, \dots, e_M)$  is the set of all the  $M$  edges in the interference graph. A vector  $\boldsymbol{\theta}^k \in \boldsymbol{\Theta}^k$  is written as  $\boldsymbol{\theta}^k = (\theta_{e_1, z(e_1)}^k, \theta_{e_1, t(e_1)}^k, \dots, \theta_{e_M, z(e_M)}^k, \theta_{e_M, t(e_M)}^k)$ , here  $z(e_i), t(e_i)$  correspond to the vertices in the edge,  $e_i$ . Similarly define,  $\boldsymbol{\theta} = (\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^{H+1}) \in \boldsymbol{\Theta}'$ , where  $\boldsymbol{\Theta}' = \prod_{k=1}^{H+1} \boldsymbol{\Theta}^k$ . The reformulated problem is stated as follows:

$$\text{DP1} \quad \min_{\boldsymbol{\beta} \in \mathcal{T}', \boldsymbol{\theta} \in \Theta'} - \sum_{i=1}^N \mathbf{W}_i (\mathbf{R}_i' \boldsymbol{\beta}_i)$$

subject to  $\mathbf{D}^k \boldsymbol{\beta}^k - \boldsymbol{\theta}^k = \mathbf{0}, \forall k \in \{1, \dots, H+1\}$

Then, DP1 can be solved using the ADMM procedure similar to the one described for DP.

#### 4.9.2 Appendix B

**Discussion on Benchmark Problem's complexity:** Benchmark Problem is restated here for convenience:

$$\begin{aligned} \text{Benchmark Problem (BP)} \quad & \max_{\boldsymbol{\pi} \in \Pi_{BC}} W(R_1(\boldsymbol{\pi}), \dots, R_N(\boldsymbol{\pi})) \\ & \text{subject to. } R_i(\boldsymbol{\pi}) \geq R_i^{min}, \forall i \in \{1, \dots, N\} \end{aligned}$$

Let the power set of  $\mathcal{U}$  be  $S_{\mathcal{U}}$ , where  $S_{\mathcal{U}}$  consists of  $2^N$  subsets of UEs. Let  $S_{\mathcal{U}}(j)$  denote the  $j^{th}$  element of  $S_{\mathcal{U}}$ . Define a set of power profiles,  $\mathcal{P}^{S_{\mathcal{U}}}$ , where the  $\mathcal{P}^{S_{\mathcal{U}}}(j)$  corresponds to the  $j^{th}$  element in the set and it corresponds to the power profile when the UEs in set  $S_{\mathcal{U}}(j)$  transmit at their maximum power levels and the rest of the UEs do not transmit. Note that for  $\boldsymbol{\pi} \in \Pi_{BC}$ ,  $\boldsymbol{\pi}(t)$  corresponds to a power profile in  $\mathcal{P}^{S_{\mathcal{U}}}$ . Therefore, the average throughput achieved by UE  $i$ ,  $R_i(\boldsymbol{\pi})$ , where  $\boldsymbol{\pi} \in \Pi_{BC}$ , can also be expressed as  $R_i(\boldsymbol{\pi}) = \sum_{j=1}^{2^N} \alpha_j r_i(\mathcal{P}^{S_{\mathcal{U}}}(j))$ , with  $\alpha_j \geq 0, \forall j \in \{1, \dots, 2^N\}$  and  $\sum_{j=1}^{2^N} \alpha_j = 1$ . Here the fraction  $\alpha_j$  associated with each profile  $\mathcal{P}^{S_{\mathcal{U}}}(j)$  corresponds to the fraction of transmission time associated with that power profile.

Consider the following problem:

$$\begin{aligned}
\textbf{BP1} \quad & \max_{\mathbf{y}, \boldsymbol{\alpha}} W(y_1, \dots, y_N) \\
& \text{subject to. } y_i \geq R_i^{\min}, \forall i \in \{1, \dots, N\} \\
& y_i = \sum_{i=1}^{2^N} \alpha_i r_i(\mathcal{P}^{Su}(j)), \forall i \in \{1, \dots, N\} \\
& \alpha_j \geq 0, \forall j \in \{1, \dots, 2^N\}, \sum_{j=1}^{2^N} \alpha_j = 1
\end{aligned}$$

Next, in order to show that the above problem is NP-hard we will show intuitively why is it so, but the detailed proof follows from proof of Theorem 1 in [LZ08]. Consider  $W(y_1, \dots, y_N) = \sum_{i=1}^N y_i$ , to be a linear function,  $R_i^{\min} = 0, \forall i \in \{1, \dots, N\}$  and the cross channel gains among some users who do not share an edge in the interference graph to be 0 and the cross channel gains among the interfering neighbors to be  $\infty$ . This implies that in any optimal solution will correspond to the transmission by a MIS of the interference graph. This can be justified as follows. Consider an optimal solution in which two neighboring UEs are transmitting, making one of the UEs not transmit will definitely increase the sum throughput contradicting the optimality. Specifically, this problem reduces to finding the maximum weighted maximum independent set which is NP-hard. Here the weight of each MIS corresponds to  $\sum_{i=1}^N r_i(\mathbf{p}^{I_j})$ .

#### 4.9.3 Appendix C

**Proof of Theorem 6.** The success probability of Phase 1 is high,  $(1 - \frac{1}{N^{c_1-1}})$  (lower bound), (see [Joh99] for detail), here we analyze Phase 2.

We first show that, if the list of remaining colors given as,  $C1_i^n$  is empty at  $n \geq \lceil c_1 \log_{\frac{4}{3}} N \rceil + \lceil c_2 \log_x N \rceil + 2$  and if this holds  $\forall i \in \{1, \dots, N\}$  then the Phase 2 has converged to a set of  $H$  MISs which span all the UEs. Let us assume otherwise, i.e.  $C1_i^n$  is empty  $\forall i \in \{1, \dots, N\}$  however, the set corresponding to some color  $h \in \{1, \dots, H\}$ ,  $I'_h$  is not a MIS.  $I'_h$  has to be an IS. Assume otherwise, i.e.  $I'_h$  is not an IS, which implies that there must exist a pair of UEs,  $i$  and  $j$ , which are neighbors and are a part of  $I'_h$ . If this is true

then both acquired the color  $h$  either in the same time slot or in different time slots, in Phase 1 or 2. In case the color is acquired in different time slots, then after the first time slot when either of the UEs in the pair acquires the color it will transmit the final color choice,  $h$  to the neighbors (see Table 4.2 and 4.3) who in turn delete that color. However, if the color is deleted by the neighbor then it cannot acquire it in the future thus, ruling out the case that the colors were acquired in two different time slots. If the color was acquired by the UEs in the same time slot, then it implies that despite the conflict in tentative choice the UEs acquire the color which is not possible (see Table 4.2 and 4.3). This shows that  $I'_h$  is an IS.

Since  $I'_h$  is not maximal then  $\exists$  at least one UE- $j \notin I'_k$  which can be added to this set without violating independence. From the assumption, we have  $C1_j^n = \phi$  which implies that the color  $h$  was deleted at some stage from the original list of all the colors either in Phase 1 or 2. The deletion of  $h$  was a result of that color being acquired finally by at least one of the neighbors  $k \in \mathcal{N}(j)$  since  $j \notin I'_k$ . In that case,  $j$  cannot acquire  $h$  as it will violate the independence property.

Next, we show that indeed the list of all colors available  $C1_i^n$  is empty at the end of Phase 2 with a high probability. Let  $U^n$  correspond to the number of UEs which have a non-empty list at the beginning of time slot  $n$  and, let  $Tn(U^n)$  correspond to the total time needed before all the UEs have an empty list. The probability that a UE at time slot  $n$  with a non-empty list will have an empty list in next time slot is always greater than  $c^H(1 - c)^{H^2}$ . This can be explained as, if the UE chooses all the colors in the list assuming (worst case  $H$  number of colors remain) and all the neighbors (worst case  $H$  neighbors) do not choose any color, then all the colors in the UE's list will be deleted. From this, we get  $E(U^{n+1}) \leq (1 - c^H(1 - c)^{H^2})U^n = \frac{1}{x}U^n$  and  $Tn(U^n) = 1 + Tn(U^{n+1})$ . Assuming that the Phase 2 will start with  $N$  UEs whose list are non-empty (worst case) and from [Kar94] we get  $P(Tn(N) \geq \lceil c_2 \log_x N \rceil) \leq \frac{1}{N^{c_2-1}}$ . This gives the lower bound on success probability of Phase 2 and thereby the result in the Theorem. ■

#### 4.9.4 Appendix D

**Proof of Theorem 7.** The two problems which are introduced to transit from CP to DP are

$$\begin{aligned}
 & \text{Global Primal Problem (GPP)} && \max_{\{\beta_i^k\}_{i,k}} \sum_{k=1}^H W_i (\sum_{i=1}^N \beta_i^k R_i^k) \\
 & \text{subject to} && \sum_{k=1}^H \beta_i^k R_i^k \geq R_i^{\min}, \sum_{k=1}^H \beta_i^k = 1, \forall i \in \{1, \dots, N\} \\
 & \beta_i^k = \beta_l^k, \forall i \neq l, \forall k \in \{1, \dots, H\}, && \beta_i^k \geq 0, \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, H\}
 \end{aligned}$$

The second problem, Local Primal Problem (LPP) is the same as GPP except we choose a subset of the constraints from the above problem. Basically, instead of an equality constraint between the UE's estimate and every other UE in the network, we only keep the equality constraints between the UE and its neighbors, i.e.  $\beta_i^k = \beta_l^k, \forall k \in \{1, \dots, H\}, \forall l \in \mathcal{N}(i)$ . This is formally stated below:

$$\begin{aligned}
 & \text{Local Primal Problem (LPP)} && \max_{\{\beta_i^k\}_{i,k}} \sum_{k=1}^H W_i (\sum_{i=1}^N \beta_i^k R_i^k) \\
 & \text{subject to} && \sum_{k=1}^H \beta_i^k R_i^k \geq R_i^{\min}, \sum_{k=1}^H \beta_i^k = 1, \forall i \in \{1, \dots, N\} \\
 & \beta_i^k = \beta_l^k, \forall l \notin \mathcal{N}(i), \forall k \in \{1, \dots, H\}, && \beta_i^k \geq 0, \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, H\}
 \end{aligned}$$

To show that problems CP and GPP are equivalent, we need to show that from  $\boldsymbol{\beta}^* = (\beta_1^*, \dots, \beta_N^*)$ , an optimal argument of GPP, we can obtain an optimal argument of CP, i.e.  $\boldsymbol{\alpha}^*$  and vice versa. Since  $\boldsymbol{\beta}^*$  is the optimal value (assuming feasibility) we know that  $\beta_i^* = \beta_j^*$  (component-wise) holds  $\forall i, j \in \{1, \dots, N\}$ .

a) Let  $\boldsymbol{\alpha}' = \boldsymbol{\beta}^*$ .  $\boldsymbol{\alpha}'$  satisfies the constraints in CP. The objective of CP at  $\boldsymbol{\alpha}'$  attains the optimal value of GPP. We need to establish that  $\boldsymbol{\alpha}'$  is indeed the optimal argument of CP. Assume that  $\boldsymbol{\alpha}'$  is not the optimal value, then there exists another  $\boldsymbol{\alpha}^*$  which is indeed optimal. Next, using  $\boldsymbol{\alpha}^*$ , we can obtain another  $\boldsymbol{\beta}'$  as follows,  $\beta'_1 = \alpha^*$  and  $\beta'_i = \beta'_1, \forall i \in \{1, \dots, N\}$ . The objective of GPP at  $\boldsymbol{\beta}'$  should be higher than  $\boldsymbol{\beta}^*$  which contradicts  $\boldsymbol{\beta}^*$  being the optimal argument. Note that if either of CP or GPP is infeasible then the other problem can be

shown to be infeasible as well. On the same lines we can show that from an  $\boldsymbol{\alpha}^*$  we can obtain  $\boldsymbol{\beta}^*$  as well.

b) Let  $\boldsymbol{\alpha}^*$  be the optimal solution to CP, and define  $\boldsymbol{\beta}''$  a solution to GPP as follows. Let  $\boldsymbol{\beta}_1'' = \boldsymbol{\alpha}^*$  and  $\boldsymbol{\beta}_i'' = \boldsymbol{\beta}_j'', \forall j \neq i$  and since  $\boldsymbol{\alpha}^*$  satisfies the constraints of CP, i.e. it is feasible, implies that  $\boldsymbol{\beta}''$  as well satisfies constraints of GPP. We want to show that  $\boldsymbol{\beta}''$  is the optimal value as well, assume that it is not and there exists an argument  $\boldsymbol{\beta}^*$  for which the objective takes a higher value. If this is the case then, from  $\boldsymbol{\beta}^*$  we can construct a  $\boldsymbol{\alpha}'$  as in part a). which, if  $\boldsymbol{\beta}^*$  takes a higher value than  $\boldsymbol{\beta}''$ , takes a higher value than  $\boldsymbol{\alpha}^*$  thus, contradicting optimality.

To show that GPP and LPP are equivalent, we use the following fact, since LPP consists of a subset of the constraints then the solution of LPP is an upper bound of the solution to GPP. We need to show that the gap between the solution of LPP and GPP is always 0. Note that for an optimal solution of LPP,  $\boldsymbol{\gamma}^* = (\gamma_1^*, \dots, \gamma_N^*)$  we know that  $\gamma_i^* = \gamma_j^* \forall j \in \mathcal{N}(i)$  (component-wise). If we can show that  $\gamma_i^* = \gamma_j^* \forall j \in \{1, \dots, N\}$  then LPP and GPP will be equivalent, since it will also satisfy all the constraints of GPP. Assume that this does not hold then  $\exists i, j$  such that  $\gamma_i^* \neq \gamma_j^*$ . Since the interference graph is connected  $\exists$  a path  $i \rightarrow j = \{i_1, \dots, i_s\}$  which implies,  $\gamma_i^* = \gamma_{i_1}^* \dots = \gamma_j^*$ . This leads to a contradiction, thereby establishing the claim.

Lastly, our goal is to show that DP is equivalent LPP. Given  $\boldsymbol{\gamma}^*$ , define  $\boldsymbol{\kappa} = \boldsymbol{\gamma}^*$  and a  $\boldsymbol{\theta} = (\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^H)$  to satisfy  $\boldsymbol{D}^k \boldsymbol{\kappa}^k - \boldsymbol{\theta}^k = \mathbf{0}, \forall k \in \{1, \dots, H\}$ , where  $\boldsymbol{\kappa}^k = (\gamma_1^{*,k}, \dots, \gamma_N^{*,k})$ . It can be shown using the same approach as we did for GPP and CP that  $(\boldsymbol{\kappa}, \boldsymbol{\theta})$  is indeed optimal argument for DP. Assume that  $(\boldsymbol{\kappa}, \boldsymbol{\theta})$  is not the optimal solution then we know that there exists  $(\boldsymbol{\kappa}^*, \boldsymbol{\theta}^*)$  for which the objective in DP takes a higher value. If this is the case, let us define  $\boldsymbol{\gamma}' = \boldsymbol{\kappa}^*$ , here  $\boldsymbol{\gamma}'$  satisfies the constraints in LPP. Also, since the objective in DP at  $(\boldsymbol{\kappa}^*, \boldsymbol{\theta}^*)$  takes a higher value than that at  $(\boldsymbol{\kappa}, \boldsymbol{\theta})$ , this yields that the objective in LPP at  $\boldsymbol{\gamma}'$  should take a higher value than that at  $\boldsymbol{\gamma}^*$ , which contradicts optimality of  $\boldsymbol{\gamma}^*$ . On the same lines, it can be easily shown that from  $(\boldsymbol{\kappa}^*, \boldsymbol{\theta}^*)$  we can construct the optimal solution  $\boldsymbol{\gamma}^*$  of the LPP. This, will establish equivalence between LPP and DP. Hence, all the four problems are equivalent. This is shown in Fig. 4.9.

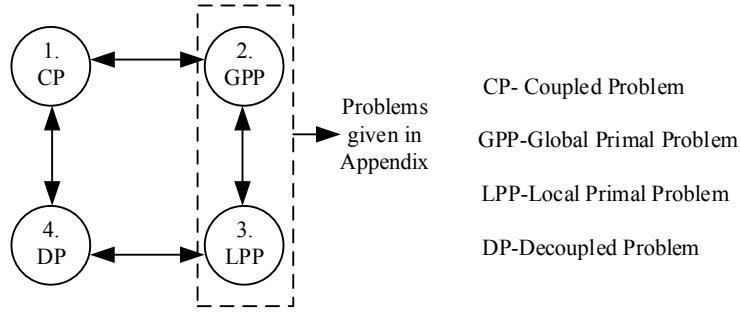


Figure 4.9: Problems used to transit from the Coupled Problem (CP) to Decoupled Problem (DP).

■

#### 4.9.5 Appendix E

**Proof of Theorem 8.** According to [WO13], the ADMM algorithm converges with rate  $O(1/P)$  if the DP is feasible and if the feasible set is compact. Since  $\mathcal{B}_i$  and  $\Theta^k$  are all closed and bounded polyhedrons, the feasible set is compact. ■

#### 4.9.6 Appendix F

**Proof of Theorem 9.** Here, we need to show three things,

- i) if  $\Delta \leq \Delta^{max}$  then the distributed policy yields a feasible solution,
- ii) the size of any MIS is  $\geq \frac{N}{\Delta+1}$ , thereby using this to show that the distributed policy if feasible will yield a network performance of at least  $\frac{N}{\Delta+1} \log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2\zeta \sigma^2})$  and
- iii) the upper bound on the network performance, sum throughput here is  $N \log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma^2})$ .

i) In the Phase 1 of the algorithm the maximum number of colors used is  $\Delta + 1$ , since each UE selects colors from a subset of  $\{1, \dots, H\} \cap \{1, \dots, d_i + 1\}$ . The first  $\Delta + 1$  output MISs,  $\{I'_1, \dots, I'_{\Delta+1}\}$  span all the UEs in the network. If the fraction of time assigned to each of these  $\Delta + 1$  MISs is,  $\alpha'_k = \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2\zeta \sigma_{ub}^2})}$ ,  $\forall k \in \{1, \dots, \Delta + 1\}$  then such an assignment satisfies the constraint that sum of fractions assigned to all the colors cannot

be more than 1, i.e. since  $\Delta \leq \Delta^{max} \implies (\Delta + 1) \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})} \leq 1$ . Using the fact that network exhibits  $\zeta$ -WNI we can write the minimum instantaneous throughput that can be obtained by UE- $i$  as,  $\log_2(1 + \frac{p_i^{max}}{(D_{iT(i)})^{np} 2^\zeta \sigma_{ub}^2})$ , and minimum instantaneous throughput of any UE as,  $\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})$ . Thus, given the fractions assigned to the MISs,  $\alpha'_k = \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})}$ ,  $\forall k \in \{1, \dots, \Delta + 1\}$ , which span all the UEs. each UE  $i$ 's throughput requirement is satisfied, i.e.,  $\frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})} \log_2(1 + \frac{p_i^{max}}{(D_{iT(i)})^{np} 2^\zeta \sigma_{ub}^2}) \geq R_{ub}^{min}$ .

ii) Assume that  $\exists$  an MIS whose size is  $S < \frac{N}{\Delta+1}$ . Each UE in the MIS can exclude a maximum of  $\Delta$  UEs from being included in the MIS. This implies that  $S(\Delta + 1)$ , represents the total number of UEs excluded and the UEs in the MIS which put together should exceed  $N$ . Since this is not the case here, the contradiction implies that  $S \geq \frac{N}{\Delta+1}$ . This combined with the minimum instantaneous throughput of any UE, we get the lower bound  $\frac{N}{\Delta+1} \log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma_{ub}^2})$ , for our policy.

iii) The upper bound on the optimal network performance is obtained by summing maximum instantaneous throughput of any UE  $\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma_{lb}^2})$  for all UEs,  $N \log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma_{lb}^2})$ . Computing the ratio of the lower bound of proposed scheme  $\frac{N}{\Delta+1} \log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma^2})$  and  $N \log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma^2})$ , we get  $\frac{\log_2(1 + \frac{p_{lb}^{max}}{(D^{ub})^{np} 2^\zeta \sigma^2})}{(\Delta+1) \log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma^2})}$  which is no less than,  $\Gamma = \frac{R_{ub}^{min}}{\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma^2})}$  since  $\Delta \leq \Delta^{max}$ .  $\blacksquare$

#### 4.9.7 Appendix G

**Proof of Theorem 10.** Let  $\Delta^* = 6\eta$  with  $\eta = \lceil \frac{\log_2(1 + \frac{1}{(D^{lb})^{np} \sigma_{lb}^2} p_{ub}^{max})}{R_{lb}^{min}} \rceil$ . We assume that the interference graph is constructed using a distance threshold rule (Section 4.5.2). Note that each UE's minimum throughput requirement is at least  $R_{lb}^{min}$ ; this combined with maximum instantaneous throughput of any UE  $\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma_{lb}^2})$  yields that each UE needs at least  $\frac{R_{lb}^{min}}{\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np} \sigma_{lb}^2})}$  fraction of time slots. First, we need to show that if there exists a clique (a subset of vertices in the graph which are mutually connected) in the interference graph of size,  $X$  greater than  $\eta$  then the minimum throughput constraints cannot be satisfied. Assume that there does exist such a clique, then any MIS based scheduling policy will allocate separate

time slots to each UE in the clique. This is true because no two UEs in the clique will belong to the same MIS. This implies that  $X \frac{R_{lb}^{min}}{\log_2(1 + \frac{p_{ub}^{max}}{(D^{lb})^{np}\sigma_{lb}^2})}$  is the total fraction of separate time slots needed which has to be less than 1. But as  $X \geq \eta$ , this leads to infeasibility. Next, if  $\Delta \geq \Delta^*$ , we claim that we will have at least one clique in the graph satisfying this condition.  $\exists$  UE- $i$  with a degree  $d_i \geq 6\eta$ , this implies that within a radius of  $D^{th}$  around SBS- $T(i)$   $\exists 6\eta$  SBSs. Also, this circle around SBS- $T(i)$  can be partitioned into 6 sectors subtending  $\frac{\pi}{3}$  at the center. The distance between any two points located in the sector is  $\leq D^{th}$ , which we justify next. Hence, all the points in a sector are mutually connected, thus forming a clique.

Let the 2-D polar coordinates of two points  $i, j$  in a sector be  $(r_i, 0)$  and  $(r_j, \theta)$ , where  $0 \leq r_i \leq D^{th}, 0 \leq r_j \leq D^{th}$  and  $0 \leq \theta \leq \frac{\pi}{3}$ . Hence, the square of the distance between the two points is expressed as  $f(r_i, r_j, \theta) = r_i^2 + r_j^2 - 2r_i r_j \cos \theta$  and our claim is that the maximum value  $f(r_i, r_j, \theta)$ , in the set of constraints above is no greater than  $(D^{th})^2$ . We formally state this as an optimization problem below:

$$\begin{aligned} \max_{r_i, r_j, \theta} \quad & f(r_i, r_j, \theta) \\ \text{subject to} \quad & 0 \leq r_i \leq D^{th}, 0 \leq r_j \leq D^{th} \\ & 0 \leq \theta \leq \frac{\pi}{3} \end{aligned}$$

Since, both  $r_i, r_j$  are non-negative, this implies that in the above optimization problem,  $\theta = \frac{\pi}{3}$  has to be satisfied in the optimal argument. Substituting  $\theta = \frac{\pi}{3}$  in  $f(r_i, r_j, \theta)$  we get,  $f(r_i, r_j, \frac{\pi}{3}) = r_i^2 + r_j^2 - r_i r_j$ . Next, we show that  $r_i^2 + r_j^2 - r_i r_j \leq (D^{th})^2$  for  $0 \leq r_i \leq D^{th}, 0 \leq r_j \leq D^{th}$ . Fix a  $0 \leq r_j \leq D^{th}$ , then  $r_i^2 + r_j^2 - r_i r_j$  takes its maximum value at  $r_i = D^{th}$ , which gives  $(D^{th})^2 + r_j^2 - D^{th} r_j$ . Since  $0 \leq r_j \leq D^{th}$ , this yields  $(D^{th})^2 + r_j^2 - D^{th} r_j \leq (D^{th})^2$  which establishes the claim.

If we have a total of  $6\eta$  SBSs in the circle then at least one sector has to have more than  $\eta$  SBSs (Pigeonhole principle), which implies that a clique of size  $X \geq \eta$  will exist.  $\blacksquare$

# CHAPTER 5

## Dynamic Matching with Strategic Agents

### 5.1 Introduction

In the previous chapters, we studied the problem of resource allocation when the agents are not selfish and work towards optimizing a common system-wide objective. In this chapter, we focus on a mechanism design problem when different agents competing for resources are selfish and want to maximize their own utilities and the mechanism designer ought to ensure that a system wide objective is maximized at the same time. We study the problem of matching strategic agents such as matching clients and workers. On crowdsourcing platforms such as Upwork, different clients compete for the workers, which are the resources for the clients, and vice-versa. Our goal is to design matching mechanism that achieve a desirable equilibrium (further details are provided later). This chapter is based on my work in [AS16].

**Motivation.** The seminal work of [Hol99] analyzes how the career concerns of an individual, i.e. the incentives to influence the current behavior of the individual and the ability of the future employers to learn about her and hence, the individual's future rewards, represent a significant force to explain the behaviors observed in many market environments. These career concerns also arise in many two-sided matching settings. For instance, in job recruitment markets, the workers desire to be matched with the clients. In industries, the managers desire to be matched with tasks/divisions. In medical school internships, the medical students desire to get internships. In these setups, the workers have career concerns, as their performance plays a significant role in determining their matches/position in the future. Both sides are self-interested and do not have sufficient information about their own and the other side's characteristics. The interactions between the two sides are repeated

in nature, and the learning influences the future opportunities. The learning during each interaction also depends on the actions taken by the two sides (e.g. the effort exerted by the workers during the interview, or the effort exerted by the managers on the tasks), which are not directly observed. Thus there is moral hazard. There can be many possible ways to organize the interactions, i.e. the matching over time. For instance, in job recruitment, the management needs to decide how to organize the interviews and how to set the payment contracts for different tasks, in crowdsourcing, the platform (such as Upwork) decides the matching rule and can prescribe payment rules for different tasks. The matching mechanism should ensure that it *facilitates learning* on both sides before final matches are achieved and that one side does not feel incentivized to *obscure learning* on the other side through their actions. Despite the ubiquitous nature of settings with matching and learning, there is no systematic theory that models these environments and characterizes the optimal mechanisms that lead to desirable matching.

**Problem overview and contributions.** In this chapter, we consider a repeated matching setting with two sides- workers and clients. All the workers and clients start with no knowledge about their characteristics, i.e. productivities of the workers, the cost of exerting effort for the workers, and the revenue generated by the tasks. Every time a worker is matched to a task for the client, she decides the amount of effort to exert; the effort is not observed by the client. Thus there is moral hazard. The client observes the output of the worker, which depends on both the productivity and the effort from the worker. Since the effort is not observed by the client, she cannot learn the worker's true productivity. The worker may feel incentivized to select actions to obscure the learning and achieve better matches in the future. The worker observes the payments made by the clients, the effort it exerted, and the cost for the effort exerted. The observations by the worker help her learn about her own characteristics.

Our main objectives in this chapter are as follows.

- a. Define coalitional stability for dynamic matching with learning under moral hazard.
- b. Construct a dynamic matching mechanism that ensures workers are not incentivized

to hinder learning through their actions, and maximizes the revenue in a coalitionally stable equilibrium.

We propose a definition of coalitional stability for environments with dynamic matching with learning in the presence of moral hazard. We construct a simple mechanism that achieves optimal revenue and coalitional stability under equilibrium in many settings. The mechanism has an initial assessment phase where each worker and client are matched exactly once followed by a reporting phase where both sides report their preferences. In the final phase, the clients and workers are matched based on their preferences using the Gale-Shapley algorithm. There can be many alternate choices for the design of the mechanism. For instance, the mechanism might solely match workers and clients based on the revenue generated/output generated and without use of reports [XDV18] or the mechanism might ask the workers to report their characteristics instead of their preferences. These alternate choices suffer from different limitations (as discussed later in the Appendix) while our mechanism satisfies the desired properties.

**Prior work.** There are several ways to categorize works in the area of matching: matching with or without transfers, matching with complete or incomplete information (with or without learning), matching with self-interested or obedient participants, matching in the presence/absence of moral hazard and adverse selection. We do not describe the works in these categories separately. Instead, in Table 5.1 at the end of the chapter, we compare with a set of representative works in each category. Next, we broadly position our work with respect to the existing works and then describe the works that are closest to us.

In many real matching setups, the presence of incomplete information is natural. For instance, in labor markets and marriage markets the two sides to be matched do not know each other's characteristics. However, in these markets when the entities on the two sides are matched to interact (worker producing output for the clients in labor markets, interaction during dating in marriage markets), they use the observations made in the interaction to learn about each other. The observations made often depend both on the characteristics and on the actions (effort in the worker-client setting) taken strategically during the interaction, which makes learning the characteristics separately non-trivial. The interaction of such a

learning process (obscured by actions) and its impact on the matching has not been studied in the existing works.

Our previous works, [XDV18], [SXZ16], have studied matching settings where both the costly unobservable effort (moral hazard) and unknown types (adverse selection) play a major role. In [XDV18], the workers are assumed to be bounded-rational as they optimize a proxy version of their utility as defined by the conjecture function, while in the present work the workers are rational, foresighted and maximize their long-run utilities. In [XDV18], [SXZ16], there is no learning of the workers' and tasks' characteristics (along the equilibrium path). The model proposed in [XDV18], [SXZ16] only applies to environments where the productivity of the worker does not vary across the tasks. In comparison, the model in this current work is more general and practical as it applies to general matching environments where the tasks are heterogeneous. In [XDV18], [SXZ16], the equilibrium matching need not necessarily be efficient: no provable guarantees with regard to optimization of revenue are given. Moreover, [XDV18], [SXZ16], do not provide any stability guarantees, unlike our work.

## 5.2 Dynamic Matching Mechanism Design

In this section, we first describe the model and problem formulation. We use  $\mathbf{A}$  for a matrix,  $\mathbf{A}(i, j)$  for an element of the matrix,  $\mathcal{A}$  for a set, and  $a/A$  for a scalar.

### 5.2.1 Model and Problem Formulation

There is one planner,  $N$  clients and  $N$  workers who desire to be matched.<sup>1</sup> We define the set of  $N$  workers as  $\mathcal{N} = \{1, \dots, N\}$  and the set of tasks as  $\mathcal{S} = \{1, \dots, N\}$ . We consider a discrete time infinite horizon model. We write each discrete time slot as  $t \in \{0, 1, \dots, \infty\}$ . Each client has one task that it wants to be repeatedly executed in each time slot. The clients and workers are assumed to be rational. In each time slot, the clients and workers

---

<sup>1</sup>The entire analysis can be extended to the setting when the number of clients and workers is not equal.

are assessed and matched according to the matching rule explained later. We assume that in each time slot one worker can be matched to at most one client and vice-versa (one-to-one matching).

**Quality distribution of the tasks.** Each task is characterized by its quality level, which is equal to the revenue generated per unit of the task.  $g : \mathcal{S} \rightarrow [g^{\min}, g^{\max}]$  maps each task to its quality level of the task, where  $g^{\min} > 0$ . We assume that  $g$  is a strictly increasing function without loss of generality. We assume that the quality of the tasks is not known to anyone.

**Productivity distribution of the workers.** Each worker  $i$ 's productivity is a measure of her skill level; it is the number of units of task a worker can complete per unit time. The productivity depends on both the worker and the type of the task that she performs.  $\mathbf{F} : \mathcal{N} \times \mathcal{S} \rightarrow [f^{\min}, f^{\max}]$  is a mapping from every combination of worker and task to a productivity level. We assume that no two workers have the same productivity for a particular task  $x$ , i.e.  $\mathbf{F}(i, x) = \mathbf{F}(k, x) \implies i = k$ . We assume that the productivity of the worker in performing a task is not known to anyone. (In Upwork, 96% of the workers have no significant experience [TSR14] to know their productivities).

**Efforts and outputs of the workers.** Each worker  $i$  decides (strategically) how much effort  $e_i$  to exert (time invested in working) on a task  $x$ , which is assigned in a particular time slot. We assume that  $e_i \in \mathcal{E}_{ix} = \{0, \delta, 2\delta, \dots e_{ix}^{\max}\}$ , where  $e_{ix}^{\max} \in [e_i^{\max}, e_u^{\max}]$ ,  $\forall i \in \mathcal{N}$ ,  $\forall x \in \mathcal{S}$ . The output produced, i.e. the total number of units of task  $x$  completed, is given as  $\mathbf{F}(i, x)e_i$  (speed of executing the task times the time spent working on it). The effort exerted by a worker is known privately to the worker only. The revenue generated is given as  $[\mathbf{F}(i, x)e_i]g(x)$ . We assume that the output produced and the revenue generated is observed by the client and the planner; this is a natural assumption, see [Hol99].

We define a cost function  $\mathbf{C} : \mathcal{S} \times \mathcal{N} \rightarrow [0, \infty)$ . It costs worker  $i$   $\mathbf{C}(i, x)e_i^2$  to exert effort  $e_i$  on task  $x$ , where  $\mathbf{C}(i, x) \in [c^{\min}, c^{\max}]$ ,  $\forall i \in \mathcal{N}, \forall x \in \mathcal{S}$ . We assume a quadratic function here for simplifying the presentation; all the results extend to any convex cost function that increases in effort. The worker  $i$  does not know their own costs  $\mathbf{C}(i, x)$ ,  $\forall x \in \mathcal{S}$ .

and no one else knows it as well. If worker  $i$  is matched to a task  $x$ , then the worker observes the cost  $\mathbf{C}(i, x)e_i^2$  and thus learns  $\mathbf{C}(i, x)$ . Also, we define a constant  $W^{max} = f^{max} [\max_{i \in \mathcal{N}, x \in \mathcal{S}} \{e_{ix}^{max}\}]$ , which denotes the maximum output across all the workers.

**Payment rule.** We assume that the payment rules are given and the clients are required to follow the payment rules; only the concerned clients know the payment rules. In the Extensions Section and the Appendix Section at the end of the chapter, we discuss the client selected payment rules. If worker  $i$  works on task  $x$  and produces  $\mathbf{W}(i, x)$  units of output (units of task completed by the worker), then the worker is paid  $p^F(\mathbf{W}(i, x), x) = \alpha \mathbf{W}(i, x)^2 g(x)$  by client  $x$ , where  $\alpha$  is a given positive constant.<sup>2</sup> We assume  $\alpha$  to be less than  $\frac{1}{2W^{max}}$  to guarantee a non-negative profit to all the clients (See Appendix Section at the end of the chapter for details.) The payment rule is quadratic in the output of the worker to ensure proportional compensation of the quadratic costs for exerting effort. We can generalize the analysis to any form of payment rule (for instance, linear etc.); we provide details in the Appendix Section at the end of the chapter.

**Set of dynamic matching mechanisms.** The planner selects the matching rule and makes it public knowledge. We first define a general vector of observations made by the planner up to time  $t - 1$  (end of time slot  $t - 1$ ) as  $\mathbf{h}_0^t$ . The elements of this general observation vector consist of the output histories of the workers, the actions that are taken by the workers (for instance, sending report about preferred clients to the planner, etc.).

We define the set of all the possible histories of all possible lengths as  $\mathcal{H}_0$ . A general matching rule is given as  $\mathbf{m} : \mathcal{H}_0 \rightarrow \Pi(\mathcal{S})$ , where  $\Pi(\mathcal{S})$  is the set of all possible permutations of  $\mathcal{S}$ . The matching rule maps each history of observations  $\mathbf{h}_0^t$  to a vector of tasks.  $\mathbf{m}(\mathbf{h}_0^t)[i]$  denotes the  $i^{th}$  element of the vector  $\mathbf{m}(\mathbf{h}_0^t)$  and corresponds to the task assigned to worker  $i$  following history  $\mathbf{h}_0^t$ .

In this chapter, we are interested in settings where each individual wants to find a long-term match. Such situations arise in long-term contracts on platforms such as Upwork, job rotation [Ort01]. Therefore, we restrict ourselves to matching rules that satisfy the following

---

<sup>2</sup>We choose a quadratic function for payments because the cost for exerting effort is quadratic.

condition:  $\lim_{t \rightarrow \infty} \mathbf{m}(\mathbf{h}_0^t)$  exists for all  $\mathbf{h}_0^t \in \mathcal{H}_0$ . Since these rules lead to a long-term match, we refer to these matching rules as *long-term matching rules*. We denote the set of all long-term matching rules as  $\mathcal{M}$ . What about the matching rules for which the limits do not exist? This is true in the settings where the workers and clients do not engage in long-term contracts and instead work on short-term basis. For instance, on platforms such as Upwork the clients in some cases offer short-term contracts and not the long-term contracts that we already discussed. We call the matching rules for which the limit do not exist as *short-term matching rules*. Note that our analysis does not apply to these short-term matching rules.

**Strategies of the workers and clients.** We define a strategy as a mapping from the history of observations of the worker to the actions. We denote the strategies of the workers as  $\{\boldsymbol{\pi}_i\}_{i=1}^N$  and the strategy for the clients as  $\{\boldsymbol{\pi}_i\}_{i=N+1}^{2N}$ . Each worker and client first need to decide whether or not to participate in the mechanism  $\mathbf{m}$ . Each client and worker starts with no observation history, i.e.  $\phi$ .  $\boldsymbol{\pi}_i(\phi) \in \{P, NP\}$  where  $P$  is for participation and  $NP$  is for not participation. Participation is the only active choice of a client (In the Extensions Section, we discuss the client selected payment rules).

In each period, each worker decides to exert some effort on the task assigned, where the effort level is only known to the worker. In some mechanisms, the planner can solicit reports from the workers about their preference over different tasks. Each worker also observes the payments made and the costs incurred for exerting effort on the tasks. We define the history of observations for each worker separately. The vector of observations of a worker  $i$  up to time  $t$  as  $\mathbf{h}_i^t$ , which consists of the efforts exerted, reports sent, the payments received and the tasks assigned upto time slot  $t - 1$  (end of time slot  $t - 1$ ). In addition,  $\mathbf{h}_i^t$  includes the task assigned in time slot  $t$ . The set of all the possible observations histories of all possible lengths is given as  $\mathcal{H}_i$ . We define the strategy of worker  $i$  as a mapping from the history of observations of the worker to the actions,  $\boldsymbol{\pi}_i : \mathcal{H}_i \rightarrow \mathcal{A}_i$ , where  $\mathcal{A}_i$  is the set of actions that a worker takes.  $a_i \in \mathcal{A}_i$  has two components  $a_i[1]$  is the effort exerted and  $a_i[2]$  is the report vector. Different choices for  $\mathbf{m}$  that impact the action set differently. We define the set of all the possible strategies as  $\Pi(\mathbf{m})$ .

**The stage game.** In time slot  $t$ , worker  $i$  is matched to play a stage game with client

$x = \mathbf{m}(\mathbf{h}_0^t)[i]$  (assuming both agreed to participate in the mechanism). The worker  $i$  exerts  $e_i^t$  effort following a private history  $\mathbf{h}_i^t$  ( $\pi_i(\mathbf{h}_i^t)[1] = e_i^t$ ). We define the output and the revenue generated by worker  $i$  in time slot  $t$  for client  $x$  as  $W_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}) = \mathbf{F}(i, x)e_i^t$  and  $r_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}) = \mathbf{F}(i, x)g(x)e_i^t$  respectively. The payment made by client  $x$  to worker  $i$  for the corresponding output is given as  $p(W_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}), x)$ . Therefore, the utility derived by the worker  $i$  in the stage game played in time slot  $t$  is computed as follows.

$$u_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}, p) = p(W_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}), x) - \mathbf{C}(i, x)(e_i^t)^2$$

Note that the above utility is quasi-linear (linear in the payments). Similarly, the utility of client  $x$  (linear in the revenue and the payments made) who is matched to worker  $i$  in time slot  $t$  is given as follows.  $v_x(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}, p) = r_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}) - p(W_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}), x)$ .

**The repeated endogenous matching game.** In every time slot, a stage game is played between a worker and a client who are matched *endogenously* based on the observation history of the planner based on  $\mathbf{m}$ . We refer to this repeated game as the “repeated endogenous matching game” and define the long-run utility for each client and each worker next.

We define the long-run utility for worker  $i$  and client  $x$  as

$$U_i(\{\pi_k\}_{k=1}^{2N} | \mathbf{m}, p) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T u_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}, p),$$

$$V_x(\{\pi_k\}_{k=1}^{2N} | \mathbf{m}, p) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T v_x(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m}, p)$$
 respectively.

The total long-run revenue is  $R(\{\pi_k\}_{k=1}^{2N} | \mathbf{m}) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T \sum_{i=1}^N r_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \pi_i | \mathbf{m})$ . It is fairly common to assume long-run average utilities in environments with career-concerns [Hol99]. We can extend the entire analysis to discounted utilities (assuming discount factor is sufficiently high).

**Knowledge and observation structure** The workers and the clients are rational, independent decision makers who do not cooperate in decision making and who wish to maximize their long-run utilities. The total number of time slots in the mechanism and the matching rules are public knowledge. The payment rules are known to the concerned client and the planner; the quality of the task is not known to anyone. The productivities and the costs of exerting effort on a task for the workers are not known to anyone. The effort exerted

by the worker and the corresponding set of effort levels are known to the worker privately. The structure of the utility (but not the parameters in the utility) of the workers and clients is known to the planner. The output and the revenue produced by the worker is observed by the concerned client and the planner. The payment made by the client to the worker are observed by the worker, the client and the planner. The reports sent by the workers to the planner are kept private between the workers and the planner. This knowledge structure is common knowledge. We summarize the knowledge structure in Table 5.2 at the end of the chapter.

### 5.2.1.1 Long-run stability of matching

We propose a definition of stability that extends the standard definitions to environments where dynamic matching is carried out with learning in the presence of moral hazard. Stability ensures that a client and a worker do not prefer to interact by themselves on the platform instead of following the matching mechanism proposed by the planner.

Consider a matching rule  $\mathbf{m} \in \mathcal{M}$  and payment rule  $p \in \mathcal{P}$ . Suppose the joint strategy of all the workers and clients is given as  $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{2N}\}$ . The history for the planner induced by the joint strategy  $\boldsymbol{\pi}$  is denoted as  $\mathbf{h}_0^{t,\boldsymbol{\pi}}$  and the history for the worker  $i$  induced by the joint strategy  $\boldsymbol{\pi}$  is given as  $\mathbf{h}_i^{t,\boldsymbol{\pi}}$ . The matching rule takes a limiting value depending upon the history, which we define as  $\mathbf{m}_{\boldsymbol{\pi}}^* = \lim_{t \rightarrow \infty} \mathbf{m}(\mathbf{h}_0^{t,\boldsymbol{\pi}})$ . The expression for the long-run utilities for worker  $i$  and client  $x = \mathbf{m}_{\boldsymbol{\pi}}^*[i]$  are simplified below (See details in the Appendix Section at the end of the chapter).

$$U_i(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T \left[ p\left(\mathbf{F}(i, \mathbf{m}_{\boldsymbol{\pi}}^*[i]) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}}), \mathbf{m}_{\boldsymbol{\pi}}^*[i]\right) - C(i, \mathbf{m}_{\boldsymbol{\pi}}^*[i]) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}})^2 \right] \quad (5.1)$$

$$V_x(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T \left[ \mathbf{F}(i, \mathbf{m}_{\boldsymbol{\pi}}^*[i]) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}}) g(j) - p\left(\mathbf{F}(i, \mathbf{m}_{\boldsymbol{\pi}}^*[i]) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}}), \mathbf{m}_{\boldsymbol{\pi}}^*[i]\right) \right] \quad (5.2)$$

The above expression for long-run utility shows that the worker's utility depends on the task assigned in the limit and not on the utility derived in the phases before being matched to this task finally. We now formalize the condition that no worker-task pair that is not matched in  $\mathbf{m}_\pi^*$  cannot strictly gain by being matched to one another by jointly choosing to deviate from participating in the matching mechanism. We assume that there are no side-payments, i.e., the payments are done solely based on worker's output following the given payment rule  $p$ . Consider worker  $i$  and a task  $y$ , where  $y \neq \mathbf{m}_\pi^*[i]$ , and suppose that this worker-task pair is matched instead of  $i$  and  $\mathbf{m}_\pi^*[i]$ . In such a case, the long-run utilities achieved by worker  $i$  and client  $y$  (in the limit), when the strategy for worker  $i$  is  $\boldsymbol{\pi}'_i$ , is defined below in (5.3) and (5.4) respectively. Observe that we are considering the case where worker  $i$ 's final match, i.e. task  $y$ , as fixed. Therefore, the strategy of others cannot impact worker  $i$  and client  $y$ 's long-run utilities. Hence, it is sufficient to consider the strategy  $\boldsymbol{\pi}'_i$  to be a function of time only.

$$\hat{U}_i^y(\boldsymbol{\pi}'_i) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \left[ \sum_{t=0}^T p \left( \mathbf{F}(i, y) \boldsymbol{\pi}'_i(t), y \right) - \mathbf{C}(i, y) \boldsymbol{\pi}'_i(t)^2 \right] \quad (5.3)$$

$$\hat{V}_y^i(\boldsymbol{\pi}'_i) = \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^T \mathbf{F}(i, y) \boldsymbol{\pi}'_i(t) g(y) - p \left( \mathbf{F}(i, y) \boldsymbol{\pi}'_i(t), y \right) \quad (5.4)$$

If a mechanism  $\mathbf{m}$  is implemented, then we define long-run stability in terms of the above expressions for long-run pairwise utilities, (5.1), (5.2), (5.3), (5.4) as follows.

**Definition 12** *Long-run pairwise stability:* A joint strategy  $\boldsymbol{\pi}$  is long-run stable under  $\mathbf{m}$  if there exists no worker-client pair  $(i, y)$ , not matched in the limit of  $\mathbf{m}$  ( $y \neq \mathbf{m}_\pi^*(i)$ ), and a strategy for worker  $i$   $\boldsymbol{\pi}'_i$  that leads to a strict increase in the long-run utility for worker  $i$  and client  $y$ , i.e.  $\hat{U}_i^y(\boldsymbol{\pi}'_i) > U_i(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p)$ ,  $\hat{V}_y^i(\boldsymbol{\pi}'_i) > V_y(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p)$ .

We extend the above definition from a pair of worker and client to any coalition of workers and clients. Suppose  $\mathcal{X}$  is the set of the workers and clients who want to deviate and we define the strategy that a worker  $i$  in the deviating set follows as  $\boldsymbol{\pi}'_i$ . The long-run utility of the worker  $i \in \mathcal{X}$  (client  $y \in \mathcal{X}$ ) is given as  $\hat{U}_i(\{\boldsymbol{\pi}'_j\}_{j \in \mathcal{X}})$  ( $\hat{V}_y(\{\boldsymbol{\pi}'_j\}_{j \in \mathcal{X}})$ ). If a subset consisting

of both workers and clients deviate, then the workers and clients can arrive at a different matching on their own. If a subset consisting of only workers deviate, then these workers can jointly try to obscure the learning on the client side.

**Definition 13** *Long-run coalitional stability:* A joint strategy  $\boldsymbol{\pi}$  is long-run coalition-stable under  $\mathbf{m}$  if there exists no subset  $\mathcal{X}$ , and a set of strategies for worker and clients in it given as  $\{\boldsymbol{\pi}'_i\}_{i \in \mathcal{X}}$  that leads to a strict increase in the long-run utility for each worker  $i \in \mathcal{S}$  and client  $y \in \mathcal{X}$ , i.e.  $\hat{U}_i(\{\boldsymbol{\pi}'_i\}_{i \in \mathcal{X}}) > U_i(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p)$ ,  $\hat{V}_y(\{\boldsymbol{\pi}'_i\}_{i \in \mathcal{X}}) > V_y(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p)$ .

From the above it is clear that long-run coalition-stability implies long-run pairwise-stability.

We now compare and contrast the difference of the proposed definition of long-run stability with the existing definitions. Shapley's works [GS62] and [SS71] proposed pairwise stability and core respectively. More recently, there have been works on stability in dynamic matching markets. In [KK18], [KMT14], the authors analyze pairwise stability in dynamic matching markets. In [DL05], [Kur09], [Dov14], authors analyze coalitional stability in dynamic matching markets. In our setup, unlike the existing setups, the preferences are learned as there is incomplete information and the preferences are affected by the actions of one side.

**Planner's problem.** The planner decides the mechanism  $\mathbf{m}$  to maximize the total long-run revenue subject to three types of constraints. The first type of constraints are the individual rationality (IR) constraints, which if satisfied guarantee that the workers and the clients participate in the mechanism. The second type of constraints are the incentive-compatibility (IC) constraints, which guarantee that every worker follows an optimal strategy (given the strategies of others). If the strategy of each worker can satisfy the IC constraint, then the joint strategy of all the workers is an equilibrium (i.e. no worker will want to deviate). We also require that the joint strategy  $\boldsymbol{\pi}$  is long-run coalitionally stable under  $\mathbf{m}$ . The planner's problem is

$$\begin{aligned}
& \max_{\boldsymbol{m} \in \mathcal{M}, p \in \mathcal{P}} R(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \boldsymbol{m}, p) \\
& \text{s.t. } V_x(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \boldsymbol{m}, p) \geq 0, \forall x \in \mathcal{S} \text{ (IR-clients)} \\
& U_i(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \boldsymbol{m}, p) \geq 0 \forall i \in \mathcal{N} \text{ (IR-workers)} \\
& U_i(\boldsymbol{\pi}_i, \{\boldsymbol{\pi}_k\}_{k=1, k \neq i}^{2N} | \boldsymbol{m}, p) \geq U_i(\boldsymbol{\pi}'_i, \{\boldsymbol{\pi}_k\}_{k=1, k \neq i}^{2N} | \boldsymbol{m}, p) \forall i \in \mathcal{N} \forall \boldsymbol{\pi}'_i; \text{ (IC-workers)} \\
& \boldsymbol{\pi} \text{ is long-run coalition-stable under } \boldsymbol{m}
\end{aligned}$$

The planner's problem outlined above is challenging because

- **Incomplete information-** The planner needs to select  $\boldsymbol{m}$  to maximize the total long-run revenue achieved by an equilibrium strategy, which depends on both the productivity of the workers  $\mathbf{F}$  and the costs  $\mathbf{C}$  that are not known to the planner. In our model, the planner and the workers do not even know the distribution of the workers' characteristics as is typically assumed in games of incomplete information.
- **Computational intractability-** The sets of possible matching rules  $\mathcal{M}$ , the payment rules  $\mathcal{P}$ , and the strategies of the workers  $\Pi(\boldsymbol{m})$  is extremely large, thus making the problem computationally intractable.

### 5.2.2 Proposed Mechanism and its Properties

First, we give a brief description of the proposed mechanism. The proposed matching rule is designed to evaluate each worker on every type of task exactly once. Since the worker is evaluated only once we refer to the proposed matching rule as “first impression is the last impression” (FILI). Based on the output of the workers a ranking of the workers over the different tasks is computed and the workers also submit a preference for the tasks to the planner. The planner computes a final matching based on these rankings and preferences, which remains fixed for all the future time slots. Next, we give a detailed description of the mechanism, which we denote as  $\boldsymbol{m}^F$ .

**Matching rule.** The FILI matching rule  $\boldsymbol{m}^F$  operates in three phases described below.

1. **Assessment phase** ( $0 \leq t \leq N - 1$ ) In this phase, the matching is carried out with the aim to evaluate workers' performances over different tasks. In time slot  $t$ , where  $t \leq N - 1$ , worker  $i$  is assigned to task  $[(t + i) \bmod N]$ , where mod is the modulus operator. Observe that in each time slot all the workers are matched to different tasks. Also, each worker is matched to every task exactly once in the first  $N$  time slots, i.e.  $0 \leq t \leq N - 1$ . At the end of each time slot, the worker, the client and the planner observe the output of the worker on the assigned task. At the end of the  $t = N - 1$  time slot, the planner must have observed the output of each worker-task combination. We write the observation of the planner in the form of a matrix  $\mathbf{W}^e$ , where  $\mathbf{W}^e(i, x)$  is the output of worker  $i$  on task  $x$  in the assessment phase.
2. **Reporting phase** ( $t = N$ ) At the start of this phase (start of the time slot  $t = N$ ), worker  $i$  is matched to task  $[(t + i) \bmod N]$  and the planner requests all the workers to submit their preferences in the form of ranks (strictly ordered) for tasks. The workers form these preferences based on the task qualities and the outputs. These rank submissions are a part of the strategy for the workers, which we describe later.<sup>3</sup> The planner computes the preferences for the clients over the workers based on the outputs  $\mathbf{W}^e$  as follows. For every client  $x$ , the planner ranks the workers based on the outputs produced on task  $x$   $\{\mathbf{W}^e(i, x)\}_{i=1}^N$ . If two workers have the same output, then the tie is broken in favor of worker with higher index.
3. **Operational phase** ( $t \geq N + 1$ ) In this phase, the final matching is computed based on the assessments in the previous phase and the preferences submitted by the workers. The planner computes the matching based on the G-S algorithm [GS62] as follows. The planner executes the G-S algorithm with the workers as the proposers and the clients as the acceptors. In each iteration of the algorithm, each worker proposes to her favorite task that has not already rejected it. Each client based on the proposals it gets keeps its favorite worker on hold and rejects the rest. At the end of at most  $N^2 - 2N + 2$  iterations, the matching that is achieved is final. The matching computed above is

---

<sup>3</sup>In practical settings, not all the tasks on the platform are very different and many of them can be categorized into one type, for instance, translation (each worker has the same productivity for tasks of the same type). In such cases, it is sufficient to evaluate the workers on tasks of different types.

fixed for the remaining time slots starting from  $N + 1$ . (Note that the  $N^2 - 2N + 2$  iterations are carried out at the start of time slot  $N + 1$ . Moreover, the G-S algorithm is executed by the planner and there is no direct interaction between the workers and the clients to execute the G-S algorithm.)

Next, we state a proposition which shows that the workers and clients are always willing to participate in the above mechanism.

**Proposition 1** *It is individually rational for all the clients and the workers to participate in the proposed mechanism.*

The proofs of all the theorems and propositions are given in the Appendix at the end of the chapter. See Appendix at the end of the chapter for the proof of Proposition 1. The proposed mechanism induces a repeated endogenous matching game as described in Section 5.2.1. In the next section, we derive an equilibrium strategy for this repeated endogenous matching game and also show that it has some very useful properties.

### 5.2.2.1 Equilibrium analysis for the repeated endogenous matching game.

For our mechanism  $\mathbf{m}^F$ , the action of the workers consists of the effort to exert in the assessment phase and the operational phase, while in the reporting phase the actions for the workers consists of both the effort to exert and the preference lists to report. Next, we propose a strategy for each worker  $i$ , which we refer to as MTBB (M-maximum, T-truthful, BB-bang-bang) strategy  $\pi_i^{MTBB}$  for the following reason. A worker following MTBB exerts maximum effort in the assessment phase, then reports the preferences truthfully in the reporting phase, and then uses a bang-bang type structure for exerting effort (maximum or no effort) in the operational phase. We will show that the MTBB strategy maximizes the long-run utility of the worker.

1. **Assessment phase** ( $0 \leq t \leq N - 1$ ) In each time slot  $t$  in this phase, where  $t \leq N$ , worker  $i$  should exert the maximum effort possible, i.e.  $e_{im^F(\mathbf{h}_0^t)[i]}^{max}$ , where  $\mathbf{m}^F(\mathbf{h}_0^t)[i] = (t + i) \bmod N$ . In each time slot, the worker receives a payment from the matched

client and also observes the cost for exerting effort. We denote the payment received by worker  $i$  in time slot  $t$  as  $\mathbf{P}(i, \mathbf{m}^F(\mathbf{h}_0^t)[i])$  and the cost incurred by worker  $i$  in time slot  $t$  as  $\bar{\mathbf{C}}(i, \mathbf{m}^F(\mathbf{h}_0^t)[i])$ . At the end of this phase, worker  $i$  knows the  $\mathbf{P}(i, x)$  and  $\bar{\mathbf{C}}(i, x)$  for all the tasks  $x \in \mathcal{S}$ .

2. **Reporting phase** ( $t = N$ ) The worker  $i$  constructs the vector of long-run utilities that the worker expects to derive by being matched as follows,  $\mathbf{U}(i, x) = \mathbf{P}(i, x) - \bar{\mathbf{C}}(i, x)$ ,  $\forall x \in \mathcal{S}$ . The worker submits a truthful ranking, i.e., ranking in the decreasing order of  $\mathbf{U}(i, x)$ . Worker  $i$  exerts maximum effort on task  $(t + i) \bmod N$  assigned to it in this time slot.
3. **Operational phase** ( $t \geq N+1$ ) The planner executes the G-S algorithm (as described above) and assigns to worker  $i$  a task  $y$ . If  $\mathbf{U}(i, y) > 0$ , then the worker exerts maximum effort  $e_{iy}^{max}$  in every time slot, and otherwise the worker exerts zero effort in every time slot. Note that the ranking list of other workers and clients is not known to worker  $i$  and thus the worker cannot predict the task she will be matched to in the operational phase.

In the next theorem, we show that the proposed MTBB is a weakly dominant strategy for each worker. Therefore, if all the workers follow the MTBB strategy, then the joint strategy will comprise an equilibrium of the repeated endogenous matching game (induced by the proposed mechanism  $\Omega^F$ ), which we refer to as the bang-bang equilibrium (BBE).

### Theorem 12 *MTBB strategy and its properties*

1. *The MTBB strategy is a weakly dominant strategy for each worker.*
2. *If all the workers follow the MTBB strategy, then the joint strategy is bang-bang equilibrium.*

See Appendix for the proof of Theorem 12.

The MTBB strategy is only weakly dominant and thus it does not imply that the bang-bang equilibrium is the unique NE. In order to play MTBB, the worker does not need information about the strategy of other workers. We study the uniqueness of BBE in next section.

**Mechanism incentivizes truthful revelation and no hindrance in learning.** The structure of the mechanism ensures that if a worker exerts maximum effort on one task, then there is no decrease in the chance of getting accepted by a task that the worker prefers more. Our design involves reporting of preferences from the worker side. Using the G-S algorithm with workers as proposers incentivizes truthful revelation [Rot82] and gives workers no incentive to hinder learning through their actions. In mechanisms that only operate based on the output and try to achieve efficient long-run performance, it can be shown that workers can strategically try to under-perform on some tasks (See Appendix at the end of the chapter for details).

**Bayesian Nash Equilibrium.** If we consider the case where the workers also have some knowledge of the form of the distribution of the productivities of other workers, then as well the above theorem continues to hold because the MTBB strategy is a dominant strategy. Therefore, the bang-bang equilibrium will be a Bayesian Nash equilibrium.

Next, we analyze properties of the BBE. We state certain assumptions next.

**Assumption 1**  $\mathbf{F}(i, x) > \mathbf{F}(k, x) \iff \mathbf{C}(k, x) > \mathbf{C}(i, x) \iff e_{ix}^{\max} > e_{kx}^{\max}, \forall i, k \in \mathcal{N}, \forall x \in \mathcal{S}$

Assumption 1 states that if a worker  $i$  has a higher productivity than another worker  $k$  on a task  $x$ , i.e.  $\mathbf{F}(i, x) > \mathbf{F}(k, x)$ , then it has a lower cost  $\mathbf{C}(i, x) < \mathbf{C}(k, x)$  for exerting effort on the same task and this is true for all the tasks  $x \in \mathcal{S}$  and vice-versa. The same condition holds for the maximum effort. Assumption 1 is natural in many settings. It states that if a worker has more experience (and skill) in performing a task, i.e. ( $\mathbf{F}(i, x) > \mathbf{F}(k, x)$ ), then the worker also has more interest in that task and is willing to spend more time on it, i.e. ( $\mathbf{C}(i, x) < \mathbf{C}(k, x)$ ).

**Theorem 13 Long-run Stability.** *If Assumption 1 holds, then bang-bang equilibrium is long-run coalition-stable under FILI matching mechanism.*

See Appendix for the proof of Theorem 13. In the Appendix, we give example of a mechanism that leads to unstable outcomes.

The above Theorem also implies that the BBE is long-run pairwise stable. Next, we compare the pairwise stability aspect with existing results in literature. Our Theorem 13 bears similarity to Theorem 5 in [Rot82]. In Theorem 5 in [Rot82] it is shown that if the matching rule is worker-optimal and outputs stable outcomes (stability in the sense of [GS62]), then the truthful revelation of preferences is the dominant strategy for all the workers. Recall that in our setting, the preference list submitted in the MTBB strategy corresponds to the true preference list. In Theorem 13, we prove that if the proposed mechanism is used (it uses worker-optimal matching in the operational phase), then we know that for every worker MTBB strategy, which leads to the truthful revelation of preferences, is a dominant strategy and is long-run stable. In both [Rot82] and our setting, it is shown that it is possible to achieve truthful revelation on the worker side and also achieve stability.

**Assumption 2** *The productivity of a worker, the cost of exerting effort, and the maximum effort of a worker, is the same across all the tasks, i.e.  $\mathbf{F}(i, x) = \mathbf{F}(i, y)$ ,  $\forall x, y$  and is denoted as  $F(i)$ ,  $\mathbf{C}(i, x) = \mathbf{C}(i, y)$ ,  $\forall x, y$  and is denoted as  $C(i)$ ,  $e_{ix}^{max} = e_{iy}^{max}$ ,  $\forall x, y$  and is denoted as  $e_i^{max}$ .*

Assumption 2 states that the type of a worker across the different tasks are the same. This is natural in settings where the tasks are homogeneous, i.e. of the same type. For instance, all the tasks can relate to a particular language of software development. This assumption requires homogeneity in task types but still allows the tasks to have different qualities. For instance, different software development tasks can generate different revenues (qualities). Moreover, the workers can still have different qualities over the tasks even though the tasks are of the same type.

**Uniqueness of the equilibrium.** In the next theorem, we show that in many cases the repeated endogenous matching game has a unique equilibrium payoff (vector of long-run utilities of the workers), which is achieved by the bang-bang equilibrium strategy. Note that the uniqueness in terms of payoffs means that there can be multiple equilibrium strategies possible but all of them lead to the same unique equilibrium payoff.

**Theorem 14 Uniqueness of the equilibrium payoff.** *If the Assumptions 1, and 2 hold, then the repeated endogenous matching game induced by the FILI matching mechanism has a unique equilibrium payoff, which is achieved by the bang-bang equilibrium strategy.*

See Appendix for the proof of Theorem 14.

It should be pointed out that we can even relax Assumption 2 to prove Theorem 14 (See details in the Appendix). Next we establish the conditions under which the proposed mechanism can be shown to be effective in mitigating both moral hazard and adverse selection and thus achieving optimal long-run revenue.

**Assumption 3** *The quality of a task  $g(x)$  is either more than  $g_u$  (high quality task) or less than  $g_l$  (low quality task).*

The above assumption ensures that if a task's quality is greater than  $g_u$ , then every worker wants to exert non-zero effort on it, else if the task's quality is lower than  $g_l$ , then no worker wants to exert effort on it.

**Theorem 15 Long-run Revenue.** *If Assumptions 2 and 3 hold, then the FILI matching mechanism  $\mathbf{m}^F$  achieves the optimal total long-run revenue among all the mechanisms  $\mathcal{M}$ .*

See Appendix for the proof of Theorem 15.

### 5.3 Numerical Experiments

In this section, we present numerical simulations to show that the performance achieved by the proposed mechanism is very high. We assume that each worker's productivity and the cost for exerting effort on every task is drawn from distributions that are known to the planner. We show that our mechanism is not restricted to quadratic payments and the results presented extend to different payment rules. The set of payment rules we use here is the union of the following two families- i) Linear payments: A worker is paid a fixed amount per unit output that it generates. Specifically, a worker is paid a fraction of the revenue

generated, where the fraction is a parameter of the payment rule that needs to be selected by the designer. ii) Quadratic payments: The client  $x$  pays a worker  $\alpha w^2 g(x)$  amount for producing  $w$  units of output. We assume that the workers also have the knowledge of the structure of the payment rule being used (linear type or quadratic type). Next, we describe the mechanisms that we will compare against.

**1. Initial belief based matching combined with optimal payment.** The planner matches the workers based on its initial beliefs about the workers as follows. The planner ranks the workers based on the mean of the beliefs across the tasks and matches the workers with the tasks assortatively, where the tasks are ranked based on their qualities. If two workers share the same rank, then the matching is done randomly for those workers. Using this as the matching rule, the planner can select the optimal payment rule from the above family of payment rules to optimize the chosen performance criterion, which can be the total long-run revenue or the total long-run profit.

Next, we describe existing mechanisms that are similar to this initial belief based matching combined with optimal payment. In many existing setups, the matching rules are similar to initial belief based matching. For instance, on Upwork the clients and the workers are matched based on the initial information provided. Also, the firms that do not practice job rotation follow a similar mechanism [Ort01] that relies only on the initial beliefs. The payment rules on platforms such as Upwork generally follow a linear payment structure.

**2. Proposed mechanism combined with optimal payment.** For our mechanism, we will use the proposed FILI matching rule  $\mathbf{m}^F$ . We will allow the planner to select the optimal payment rules from the same family of payment rules described at the beginning of this section (given the fixed choice of matching rule  $\mathbf{m}^F$ ).

**3. Upper bound on the total long-run revenue and the profit.** We use the upper bound for the total long-run revenue and the total long-run profit that we derive in the Appendix.

In Fig. 5.1, we compare the performance of the proposed mechanism combined with optimal payment against the mechanisms described above and the upper bound derived

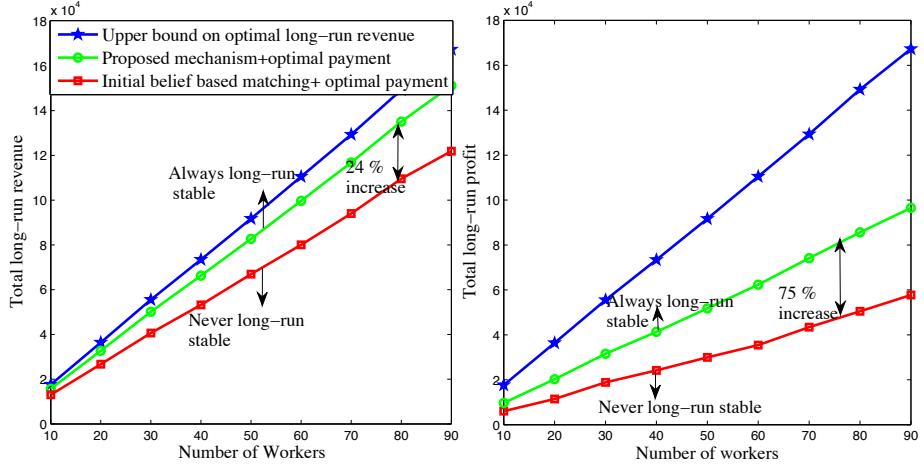


Figure 5.1: Comparison of the proposed mechanism with other approaches.

in the Appendix. The details of the setup for the numerical simulations can be found in Appendix. It can be seen that the proposed mechanism leads to large gains of over 75 percent and is always long-run stable.

## 5.4 Extensions

**Payment rules decided by the clients.** In Section 5.2, we considered the settings where the clients comply and use the payment rules set by the planner. We can extend some of the important results presented in this chapter to settings where the choice of payment rules is a part of the client's strategy and are not set by the planner. For ease of exposition, we will assume that each client has to choose from a set of linear payment rules- client pays the worker a fraction of the revenue generated, where the fraction is decided by the client. The costs for exerting effort for the workers is a linear function in the exerted effort as well. Suppose that Assumptions 1, 2, and 3 hold. We also assume that the clients know the distribution from which the workers are drawn and vice-versa. Under these conditions, we can arrive at an equilibrium strategy, which is very similar to the bang-bang equilibrium strategy. We can also show that the matching achieved is long-run stable with respect to this equilibrium strategy. The only new component in the equilibrium strategy that needs

explanation are clients' payment rules. The client with the highest quality task will want to attract the worker with highest quality (since Assumptions 1 and 2 hold). The client with highest task quality will need to use a payment rule that guarantees that the worker with highest quality is paid at least as much as being offered by the client with second highest task quality. The same argument applies to the client with second highest task quality and so on. In such a case, all the clients will set a payment rules such that the amount paid to all the workers ( $\text{fraction} \times \text{revenue}$ ) is the same. Further details are in the Appendix. We also discuss some other extensions in the Appendix.

## 5.5 Conclusion

In this chapter, we consider an environment with career concerns, where the workers are assessed by different clients over time before finally getting matched and then working for a particular client. The mechanism considered requires the planner to take actions based on the outputs produced by the self-interested workers, where the outputs depend on both productivity of workers (unknown thus adverse selection) and efforts exerted (unobserved thus moral hazard). Therefore, the model features both adverse selection (on both sides) and moral hazard (on one side). We construct a mechanism – matching and payment rules – that ensures that both moral hazard and adverse selection thereby achieving high total long-run revenue (total long-run profits) in a wide-range of settings. We also show that in the proposed mechanism, the workers find it optimal to follow simple maximum truth bang-bang (MTBB) strategies. We propose a notion of stability - “long-run stability”, which is meaningful for matching environments with incomplete information and learning. In a wide-range of settings, we prove that our proposed mechanism achieves long-run stability.

## 5.6 Appendix

In all the proofs we will use  $I(A)$  as the indicator function. If the condition  $A$  holds, then the indicator is one and zero otherwise.

### 5.6.1 Appendix A

**Proof of Proposition 1.** It is easy to see that the workers can always ensure a zero long-run utility (outside option of the worker gives zero utility) by exerting zero effort. Therefore, the participation constraint for the workers is trivially satisfied. If  $\alpha \leq \frac{1}{2W^{max}}$ , then the profit per unit output is always greater than or equal to zero which ensures that the clients cannot have a negative profit in any period. Thus the clients cannot have a negative long-run profit.

■

### 5.6.2 Appendix B

**Proof of Theorem 12.** From the Proposition 1 we know that the clients will participate in the mechanism. Hence, in this proof we only focus on the worker's strategies. There are two parts to the Theorem. In the first part, we need to show that the MTBB strategy is a weakly dominant strategy. First, we will simplify the expression for the long-run utility of the worker  $i$  when the proposed mechanism  $\Omega^F = (\mathbf{m}^F, p^F)$  is implemented. We write the joint strategy for all the workers as  $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N)$ . In Section 5.2, where we defined the strategy of the workers for a given mechanism  $\Omega$ , we used a general definition for the action set. The strategy consisted of two parts,  $\boldsymbol{\pi}_i(h_i^t)[1]$  is the effort exerted by worker  $i$  and  $\boldsymbol{\pi}_i(h_i^t)[2]$  is the reports submitted by the worker. For our proposed mechanism  $\Omega^F$ , the second component of reports only plays a role in time slot  $N$ , i.e. the reporting phase, and for the rest of the time slots the clients can choose to send no reports as it does not impact the interactions in any way.

We write the private history of worker  $i$ , which is induced by the joint strategy  $\boldsymbol{\pi}$  as  $h_i^{t,\boldsymbol{\pi}}$ . We write the preference list provided by worker  $i$  in the reporting phase as

$$\mathbf{b}_i = \boldsymbol{\pi}_i(h_i^{N,\boldsymbol{\pi}})[2] \quad (5.5)$$

The output produced in time slot  $t$  by worker  $i$  assigned to task  $j = (t+i) \bmod N$  is written as

$$\mathbf{W}^e(i, j) = \mathbf{F}(i, j)\boldsymbol{\pi}_i(h_i^{t,\boldsymbol{\pi}}) \quad (5.6)$$

The G-S algorithm executed by the planner at the beginning of the operational phase takes as input the preference lists  $\{\mathbf{b}_i\}_{i=1}^N$  and the outputs produced by the workers  $\mathbf{W}^e$ . We represent the output of the G-S algorithm as

$$\mathbf{m}^{GS}(\{\mathbf{b}_i\}_{i=1}^N, \mathbf{W}^e) \quad (5.7)$$

where  $\mathbf{m}^{GS}$  is a function that takes the preference lists and performance of workers as input and outputs the matching. Note that we do not explicitly write the observation history of the planner  $\mathbf{h}_0^t$ . The joint strategy  $\boldsymbol{\pi}$  induces an observation history for the planner, which we write as  $\mathbf{h}_0^{t,\boldsymbol{\pi}}$ . Note that  $\mathbf{h}_0^{t,\boldsymbol{\pi}}$  and  $\{\mathbf{b}_i\}_{i=1}^N, \mathbf{W}^e$  contain the same relevant information needed for the final matching to be determined by G-S algorithm. For consistency, we state that when  $t \geq N + 1$ ,

$$\mathbf{m}^F(\mathbf{h}_0^{t,\boldsymbol{\pi}}) = \mathbf{m}^{GS}(\{\mathbf{b}_i\}_{i=1}^N, \mathbf{W}^e) \quad (5.8)$$

is the notation for the proposed matching rule given in Section 5.2.

The expression for the long-run utility for worker  $i$  defined in Section 5.2 is simplified by substituting (5.8) as follows.

$$\begin{aligned} U_i(\{\boldsymbol{\pi}_k\}_{k=1}^N | \mathbf{m}^F, p^F) = \\ \lim_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=N+1}^T \left[ \alpha \mathbf{F}(i, \mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i])^2 g(\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i]) - \right. \\ \left. \mathbf{C}(i, \mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e[i])) \right] (e_i^t)^2 \end{aligned} \quad (5.9)$$

In the above expression (5.9),  $e_i^t = \boldsymbol{\pi}_i(h_i^{t,\boldsymbol{\pi}})[1]$ . In the above expression (5.9), we did not write the utility from the assessment and reporting phase because the number of time slots in assessment phase are finite  $N + 1$  and thus utilities in the assessment phase do not contribute to the long-run utility.

We define

$$\bar{e}_i^2 = \lim_{T \rightarrow \infty} \sum_{t=0}^T \frac{1}{T+1} (e_i^t)^2 \quad (5.10)$$

We define

$$\begin{aligned} \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) = \\ \alpha \mathbf{F}(i, \mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i])^2 g(\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i]) - \mathbf{C}(i, \mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e[i])) \end{aligned} \quad (5.11)$$

Thus we can simplify the above utility (5.9) by substituting (5.10), (5.11) as follows.

$$U_i(\{\boldsymbol{\pi}_k\}_{k=1}^N | \mathbf{m}, p) = \bar{e}_i^2 \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \quad (5.12)$$

Next, we want to solve for the optimal strategy  $\boldsymbol{\pi}_i$  given the fixed strategy of the rest of the workers  $\boldsymbol{\pi}_{-i}$ . Formally stated, the optimization problem is given as follows.

$$\max_{\boldsymbol{\pi}_i} U_i(\{\boldsymbol{\pi}_k\}_{k=1}^N | \mathbf{m}^F, p^F) \quad (5.13)$$

We will first compute an upper bound for (5.12). Observe that

$$\begin{aligned} U_i(\{\boldsymbol{\pi}_k\}_{k=1}^N | \mathbf{m}, p) = \bar{e}_i^2 \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \leq \\ (e_{i\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i]}^{\max})^2 \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) I\left(\mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \geq 0\right) \end{aligned} \quad (5.14)$$

In the above expression (5.14), the LHS will achieve the same value as the RHS provided worker  $i$  follows the following strategy. If  $t \geq N + 1$  and  $\mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \geq 0$ , then  $e_i^t = e_{i\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i]}^{\max}$  and  $e_i^t = 0$  zero otherwise. We now compute the optimal value for the maximum for the RHS. The expression in RHS depends only on the actions taken in the assessment and the reporting phase. Based on the above inequality (5.14), we can say that the optimizer of the RHS in terms of the actions in the assessment and reporting phase will be an upper bound of the maximization problem in (5.12). We first maximize the expression in RHS with respect to the choice of preference lists submitted in the reporting phase.

We claim that if worker  $i$  ranks the clients in the order of  $[\alpha \mathbf{F}(i, j)^2 g(j) - \mathbf{C}(i, j)](e_{ij}^{\max})^2$  for all  $j$ , then it corresponds to the best choice of the preference list. We denote this preference list as  $\mathbf{b}_i^*$ . This claim follows from Theorem 5 [Rot82], where it is shown that the truthful reporting is a dominant strategy when the matching rule is worker optimal and leads to stable outcomes.

Thus we can write

$$\begin{aligned}
& U_i(\{\boldsymbol{\pi}_k\}_{k=1}^N | \mathbf{m}, p) = \\
& \bar{e}_i^2 \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \leq (e_{i\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e)[i]}^{\max})^2 \mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) I\left(\mathbf{H}_i(\{\mathbf{b}_k\}_{k=1}^N, \mathbf{W}^e) \geq 0\right) \leq \\
& (e_{i\mathbf{m}^{GS}(\{\mathbf{b}_k\}_{k=1, k \neq i}^N, \mathbf{b}_i^*, \mathbf{W}^e)[i]}^{\max})^2 \mathbf{H}_i(\mathbf{b}_i^*, \{\mathbf{b}_k\}_{k=1, k \neq i}^N, \mathbf{W}^e) I\left(\mathbf{H}_i(\mathbf{b}_i^*, \{\mathbf{b}_k\}_{k=1, k \neq i}^N, \mathbf{W}^e) \geq 0\right)
\end{aligned} \tag{5.15}$$

Next, we will show that if the preference list is fixed for worker  $i$  to  $\mathbf{b}_i^*$ , then the choice of effort level for task  $j$  in the assessment phase, which is denoted as  $e_{ij}^{\text{eval}}$ , that maximizes the RHS of the above expression (5.15) is  $e_{ij}^{\max}$ . We do so by arguing that the long-run utility of the worker increases in  $e_{ij}^{\text{eval}}$ .

If the worker increases  $e_{ij}^{\text{eval}}$  to  $e_{ij}^{\text{eval}} + \delta$ , then the ranking of the worker by task  $j$  can either stay the same or increase. Since other parameters remain the same, the ranking of worker  $i$  on other tasks does not change. In this case, there are three possibilities. Suppose that the worker exerts effort levels  $\{e_{ik}^{\text{eval}}\}_{k=1}^N$  in the assessment phase on different tasks and submits the preference list  $\mathbf{b}_i^*$  in the reporting phase and is matched to task  $j_1$  in the operation phase. We will analyze the behavior of the (5.15) when we vary the effort level of worker  $i$  on task  $j$   $e_{ij}^{\text{eval}}$ . It is possible that rank of task  $j_1$  in the preference list  $\mathbf{b}_i^*$  is greater than task  $j$  or equal or lesser. If the rank of  $j_1$  is greater than  $j$ , then the worker even after increasing effort on task  $j$  will still be accepted by  $j_1$  as the ranking of the worker for  $j_1$  and ranking of  $j_1$  for all workers is not affected by  $e_{ij}^{\text{eval}}$ . Therefore, in this case, increasing the effort  $e_{ij}^{\text{eval}}$  will not change the rank of the task that is assigned.

If the rank of  $j_1$  is equal to  $j$ , then by increasing the effort  $e_{ij}^{\text{eval}}$  can only improve worker's ranking for task  $j$ . The ranking of worker  $i$  on tasks ranked higher than task  $j$  is still the same, thus worker  $i$  will be rejected by all those tasks. But since the ranking of worker  $i$  on task  $j$  is the same or higher it means that the worker will be assigned to  $j$ .

If the ranking of task  $j_1$  is lesser than the rank of task  $j$ , then note that the ranking of the worker on task  $j_1$  will not change and thus the worker will still be accepted by task  $j_1$ .

at least. However, since the worker increases effort on task  $j$ , the ranking of the worker can improve on task  $j$ . This means that it is possible that the worker is accepted by a strictly higher ranked task. Thus we know that increasing effort  $e_{ij}^{eval}$  can lead to the worker being matched to a task with higher or the same rank as before. A task with higher or the same rank will imply a higher or the same value for the long-run utility of the worker. Hence, the  $e_{ij}^{eval} = e_{ij}^{max}$  is the optimal choice at which the upper bound in the RHS is maximized. This holds for all the tasks that worker  $i$  is matched to for the first time in the assessment phase. Observe that the proposed MTBB strategy achieves the value for the upper bound in the RHS, thus it has to be the best response for a worker to every strategy of other workers.

The next part of the theorem follows easily from the fact that since all the workers use their best response strategies the joint strategy has to be an equilibrium. ■

### 5.6.3 Appendix C

**Proof of Theorem 13.** We first prove long-run pairwise stability. Before we give the proof of Theorem 13, we first need to simplify and arrive at the expressions for the long-run utilities for the workers and clients as given in Section 5.2.1. We only consider the matching rules for which the limit of the matching exists across all the histories. Suppose the joint strategy being used by the workers and the clients is  $\pi$ . Under this joint strategy the limit of the matching rule is given as  $\mathbf{m}_\pi^*$ . The history that is induced by the joint strategy  $\pi$  is defined as  $\mathbf{h}_0^{t,\pi}$  and the  $\mathbf{h}_i^{t,\pi}$  for worker  $i$ . Note that the  $\lim_{t \rightarrow \infty} \mathbf{m}(\mathbf{h}_0^{t,\pi}) = \mathbf{m}_\pi^*$ , where the limit is defined using the standard Euclidean norm in the space  $\mathbb{R}^N$  as the distance metric. Next, we will show that the above limit is attained after a finite number of time slots denoted as  $T_{lim}$ . Note that the minimum distance between any two distinct matching is finite and is given as  $d_{min}$ . From the definition of limit, it is clear that there exists a constant  $T_{lim}$  such that if  $t \geq T_{lim}$ , then the distance between  $\mathbf{m}(\mathbf{h}_0^{t,\pi})$  and  $\mathbf{m}_\pi^*$  is less than  $d_{min}$ . Therefore, for all  $t \geq T_{lim}$

$$\mathbf{m}(\mathbf{h}_0^{t,\pi}) = \mathbf{m}_\pi^* \quad (5.16)$$

Based on the above simplification we can write the long-run utility of a worker  $i$  and

client  $x = \mathbf{m}_\pi^*[i]$  as follows.

$$\begin{aligned} U_i(\{\boldsymbol{\pi}_k\}_{k=1}^{2N} | \mathbf{m}, p) &= \lim_{T \rightarrow \infty} \sum_{t=T_{lim}}^T \frac{1}{T+1} p(\mathbf{F}(i, x) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}}), x) - \mathbf{C}(i, x) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}})^2 \\ &= \lim_{T \rightarrow \infty} \sum_{t=0}^T \frac{1}{T+1} p(\mathbf{F}(i, x) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}}), x) - \mathbf{C}(i, x) \boldsymbol{\pi}_i(\mathbf{h}_i^{t,\boldsymbol{\pi}})^2 \end{aligned} \quad (5.17)$$

Similar justification applies for the clients' long-run utilities as well.

We write the matching achieved in bang-bang equilibrium at the start of the operational phase as  $\mathbf{m}^{BBE}$ . The long-run utility for worker  $i$  in bang-bang equilibrium

$$U_i(\{\boldsymbol{\pi}_k^{MTBB}\}_{k=1}^N | \mathbf{m}^F, p^F)$$

is simplified below

$$\begin{aligned} &[\alpha \mathbf{F}(i, \mathbf{m}^{BBE}[i])^2 g(\mathbf{m}^{BBE}[i]) - \mathbf{C}(i, \mathbf{m}^{BBE}[i])] (e_{i\mathbf{m}^{BBE}[i]}^{\max})^2 \\ &I(\alpha \mathbf{F}(i, \mathbf{m}^{BBE}[i])^2 g(\mathbf{m}^{BBE}[i]) - \mathbf{C}(i, \mathbf{m}^{BBE}[i]) \geq 0) \end{aligned} \quad (5.18)$$

Define  $\mathbf{J} : \mathcal{N} \times \mathcal{S} \rightarrow \mathbb{R}$  and  $\mathbf{L} : \mathcal{N} \times \mathcal{S} \rightarrow \mathbb{R}$  as follows.

$$\begin{aligned} \mathbf{J}(k, x) &= [\alpha \mathbf{F}(k, x)^2 g(x) - \mathbf{C}(k, x)] \\ \mathbf{L}(k, x) &= I(\alpha \mathbf{F}(k, x)^2 g(x) - \mathbf{C}(k, x) \geq 0) \end{aligned} \quad (5.19)$$

We can write (5.18) using (5.19) more succinctly as follows.

$$U_i(\{\boldsymbol{\pi}_k^{MTBB}\}_{k=1}^N | \mathbf{m}^F, p^F) = \mathbf{J}(i, \mathbf{m}^{BBE}[i]) \mathbf{L}(i, \mathbf{m}^{BBE}[i]) (e_{i\mathbf{m}^{BBE}[i]}^{\max})^2 \quad (5.20)$$

The long-run utility for client  $\mathbf{m}^{BBE}[m]$ , where  $m \neq i$ , in bang-bang equilibrium is given as follows.

$$\begin{aligned} &(1 - \alpha \mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{\max}) \\ &I(\alpha \mathbf{F}(m, \mathbf{m}^{BBE}[m])^2 g(\mathbf{m}^{BBE}[m]) - \mathbf{C}(m, \mathbf{m}^{BBE}[m]) \geq 0) \mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{\max} \end{aligned} \quad (5.21)$$

We can simplify (5.21) using (5.19) as follows.

$$(1 - \alpha \mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{\max}) \mathbf{L}(m, \mathbf{m}^{BBE}[m]) \mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{\max} \quad (5.22)$$

Suppose worker  $i$  is matched to client  $\mathbf{m}^{BBE}[m]$  instead in the operational phase. Our objective here is to show that it is not possible for both worker  $i$  and client  $\mathbf{m}^{BBE}[m]$  to increase their long-run utilities by being matched to one another and this holds true for every  $i \neq m$ .

If the utility for worker  $i$  strictly increases by being matched to  $\mathbf{m}^{BBE}[m]$ , then it has to hold true that  $[\alpha F(i, \mathbf{m}^{BBE}[m])^2 g(\mathbf{m}^{BBE}[m]) - C(i, \mathbf{m}^{BBE}[m])] (e_{im^{BBE}[m]}^{\max})^2$  has to be strictly higher than  $[\alpha F(i, \mathbf{m}^{BBE}[i])^2 g(\mathbf{m}^{BBE}[i]) - C(i, \mathbf{m}^{BBE}[i])] (e_{im^{BBE}[i]}^{\max})^2$ . This has to hold true because otherwise the maximum utility that worker  $i$  can achieve by getting matched to  $\mathbf{m}^{BBE}[m]$  will always be lesser than or equal to the long-run utility that the worker can achieve by getting matched to task  $\mathbf{m}^{BBE}[i]$  in the operational phase of the bang-bang equilibrium.

We can write the utility for worker  $i$  when it is matched to  $\mathbf{m}^{BBE}[m]$  (denoted as  $\hat{U}_i^{\mathbf{m}^{BBE}[m]}(\boldsymbol{\pi}'_i | p^F)$ ) in the operational phase and when it follows strategy  $\boldsymbol{\pi}'_i$  as follows. As explained in Section 5.2, that it is sufficient to consider the strategies  $\boldsymbol{\pi}'_i$  that only depend on time.

$$\hat{U}_i^{\mathbf{m}^{BBE}[m]}(\boldsymbol{\pi}'_i | p^F) = \mathbf{J}(i, \mathbf{m}^{BBE}[m]) \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \lim_{T \rightarrow \infty} \sum_{t=N+1}^T \frac{\boldsymbol{\pi}'_i(t)^2}{T+1} \quad (5.23)$$

We write  $\lim_{T \rightarrow \infty} \sum_{t=N+1}^T \frac{\boldsymbol{\pi}'_i(t)^2}{T+1} = \bar{e}_i^2$ ,  $\lim_{T \rightarrow \infty} \sum_{t=N+1}^T \frac{\boldsymbol{\pi}'_i(t)}{T+1} = \bar{e}_i$  and substitute in (5.23) to obtain the following.

$$\hat{U}_i^{\mathbf{m}^{BBE}[m]}(\boldsymbol{\pi}'_i | p^F) = \mathbf{J}(i, \mathbf{m}^{BBE}[m]) \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \bar{e}_i^2 \quad (5.24)$$

Also, the utility for client  $\mathbf{m}^{BBE}[m]$  in this case is derived as follows.

$$\begin{aligned}
& \hat{V}_{\mathbf{m}^{BBE}[m]}^i(\boldsymbol{\pi}'_i | p^F) \\
&= \lim_{T \rightarrow \infty} \sum_{t=N+1}^T \frac{1}{T+1} \left[ 1 - \alpha \mathbf{F}(i, \mathbf{m}^{BBE}[m]) \boldsymbol{\pi}'_i(t) \right] \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \mathbf{F}(i, \mathbf{m}^{BBE}[m]) \boldsymbol{\pi}'_i(t) \\
&= [\mathbf{F}(i, \mathbf{m}^{BBE}[m]) \bar{e}_i - \alpha \mathbf{F}(i, \mathbf{m}^{BBE}[m])^2 \bar{e}_i^2] \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \\
&\leq [\mathbf{F}(i, \mathbf{m}^{BBE}[m]) \bar{e}_i - \alpha \mathbf{F}(i, \mathbf{m}^{BBE}[m])^2 (\bar{e}_i)^2] \mathbf{L}(i, \mathbf{m}^{BBE}[m])
\end{aligned} \tag{5.25}$$

Based on the G-S algorithm and the fact that every worker uses MTBB strategy, we know that the rank of worker  $m$  is higher than the rank of worker  $i$  for task  $\mathbf{m}^{BBE}[m]$ .

$$\mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{max} > \mathbf{F}(i, \mathbf{m}^{BBE}[m]) e_{i\mathbf{m}^{BBE}[m]}^{max} \tag{5.26}$$

From the above (5.26), either the productivity or the maximum effort has to be strictly higher. From Assumption 1, we know that if one of them is true, then the other is also true. In addition, we can say the following:

$$\mathbf{F}(m, \mathbf{m}^{BBE}[m]) \geq \mathbf{F}(i, \mathbf{m}^{BBE}[m]) \implies \mathbf{C}(i, \mathbf{m}^{BBE}[m]) \geq \mathbf{C}(m, \mathbf{m}^{BBE}[m]) \tag{5.27}$$

Based on the above (5.27), we can show the following.

$$\mathbf{J}(m, \mathbf{m}^{BBE}[m]) \geq \mathbf{J}(i, \mathbf{m}^{BBE}[m]) \tag{5.28}$$

$$\mathbf{L}(m, \mathbf{m}^{BBE}[m]) \geq \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \tag{5.29}$$

Observe that the function  $(1 - \alpha x)x$  is increasing in  $[0, \frac{1}{2\alpha}]$ . We assumed that  $\alpha \leq \frac{1}{2W^{max}}$ . Therefore,  $(1 - \alpha x)x$  is increasing in  $x \in [0, W^{max}]$ . Note that

$$\begin{aligned}
W^{max} &\geq \mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{max} \geq \mathbf{F}(i, \mathbf{m}^{BBE}[m]) e_{i\mathbf{m}^{BBE}[m]}^{max} \geq \mathbf{F}(i, \mathbf{m}^{BBE}[m]) \boldsymbol{\pi}'_i(t)
\end{aligned} \tag{5.30}$$

We can use the above relations (5.27), (5.28), (5.29), (5.30) to derive the following condition on the expression in (5.25).

$$\begin{aligned}
& [\mathbf{F}(i, \mathbf{m}^{BBE}[m]) \bar{e}_i - \alpha \mathbf{F}(i, \mathbf{m}^{BBE}[m])^2 \bar{e}_i^2] \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \leq \\
& [\mathbf{F}(i, \mathbf{m}^{BBE}[m]) e_{i\mathbf{m}^{BBE}[m]}^{max} - \alpha \mathbf{F}(i, \mathbf{m}^{BBE}[m])^2 (e_{i\mathbf{m}^{BBE}[m]}^{max})^2] \mathbf{L}(i, \mathbf{m}^{BBE}[m]) \leq \\
& [\mathbf{F}(m, \mathbf{m}^{BBE}[m]) e_{m\mathbf{m}^{BBE}[m]}^{max} - \alpha \mathbf{F}(m, \mathbf{m}^{BBE}[m])^2 (e_{m\mathbf{m}^{BBE}[m]}^{max})^2] \mathbf{L}(m, \mathbf{m}^{BBE}[m])
\end{aligned} \tag{5.31}$$

Therefore, from the above (5.31), we can see that the client  $\mathbf{m}^{BBE}[m]$  cannot have a strict gain at the same time as worker  $i$ . Thus we can conclude that the proposed matching rule has to be long-run pairwise-stable w.r.t bang-bang equilibrium strategy (joint MTBB strategy). We now move on to long-run coalition-stability.

Let us assume that the BBE strategy is not long-run coalitionally stable. Therefore, we know that there exists a subset  $\mathcal{S}$ , which strictly benefits from deviating. We analyze three possibilities for the deviating subset. Suppose that the deviating subset consists of only workers. The joint strategy of the deviating subset and the equilibrium strategy of the non-deviating set outperforms the BBE strategy. Therefore, for each worker  $i$  the following condition is true

$$\hat{U}_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) > U_i(\boldsymbol{\pi}^{BBE})$$

We know that  $\boldsymbol{\pi}_i^{BBE}$  is a weakly dominant strategy. Therefore,

$$\hat{U}_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) = U_i(\boldsymbol{\pi}_i^{BBE}, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c})$$

Firstly, we know that  $U_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) > 0$  otherwise it cannot be the case that

$$\hat{U}_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) > U_i(\boldsymbol{\pi}^{BBE})$$

The minimum possible utility in the equilibrium is zero, i.e.  $U_i(\boldsymbol{\pi}^{BBE}) > 0$ .

We know that the utility that the worker  $i$  gets is non-zero and based on our assumptions, we know that there are no ties in the preferences in BBE. Therefore, by switching from  $\boldsymbol{\pi}'_i$  to  $\boldsymbol{\pi}_i^{BBE}$  the worker  $i$  is matched to the same task  $z$  as it was matched under  $U_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c})$ .

For a worker  $j \neq i$  that is in  $\mathcal{S}$ , we calculate the impact when worker  $i$  switches from  $\boldsymbol{\pi}'_i$  to  $\boldsymbol{\pi}_i^{BBE}$ .

Suppose worker  $j$  was matched to task  $w$  under  $\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}$ . When worker  $i$  switched to the MTBB strategy from  $\boldsymbol{\pi}'_i$  there are two possibilities

- Task  $w$  was not above the task  $z$  under  $\boldsymbol{\pi}_i^{BBE}$  in which case the change in strategy does not impact worker  $j$

- Task  $w$  was above task  $z$  in  $\pi_i^{BBE}$ . Worker  $i$  is rejected in favor of another worker. Since the strategies of all the other workers remain fixed the worker  $j$  continues to be ranked above all the workers that propose to task  $z$ . Hence, worker  $j$  is matched to task  $w$ .

$$\hat{U}_j(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) = U_j(\boldsymbol{\pi}_i^{BBE}, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c})$$

This holds true for all  $j \neq i$ . Hence, all the workers  $j \neq i$  continue to be matched to the same task and derive the same utility when worker  $i$  switches. We can continue this argument and get that

$$\hat{U}_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}'_{-i}, \boldsymbol{\pi}_{\mathcal{S}^c}) = U_i(\boldsymbol{\pi}^{BBE})$$

The above is a contradiction. Hence, the workers cannot strictly gain by deviating.

Suppose that the subset consists of only clients. Clients alone cannot gain from deviating as no workers to match to.

Suppose that the subset consists of at least one worker and one client. In this case, there will be at least one pair of a client and worker that are matched that gain. If this is the case, then that violates pairwise-stability. Hence, in all the three cases we arrive at a contradiction. This shows that it is not possible to have a profitable deviation by a coalition.

■

#### 5.6.4 Appendix D

**Proof of Theorem 14.** Before we provide the proof, it is important to be reminded of how we define the uniqueness of the equilibrium. Each equilibrium strategy has a corresponding equilibrium payoff. If for the repeated game that we analyze all the possible equilibrium strategies lead to the same payoff, then we call the equilibrium payoff to be unique. In this Theorem, we will assume that the Assumption 1 and 2 hold. We can combine the Assumption 1 and 2 and interpret them together as follows.

From Assumption 1 and 2, we can see that the preference list for all the workers in the MTBB strategy is the same and corresponds to the ranking of the tasks in order of their qualities. Also, if the workers follow the MTBB strategy, then the ranking of the workers as computed by the planner for every client is the same as well. Specifically, the ranking of the workers is based on the outputs in the assessment phase, where the set of outputs in assessment phase is given as  $\{F(k)e_k^{max}\}_{k=1}^N$ . Hence, from now on in this proof when we refer to the ranking of the tasks, it is the same as the ranking done by every worker in the MTBB strategy unless stated specifically otherwise. Similarly, when we refer to ranking of workers it is the same as the ranking of the workers based on their maximum outputs computed by the planner for every client.

First, we show that there does not exist another equilibrium in which at least one worker  $i$  can achieve a higher utility than the utility achieved in bang-bang equilibrium (i.e. the joint MTBB strategy). If all the workers play the MTBB strategy, then the matching that is computed at the start of the operational phase after the execution of G-S algorithm is denoted as  $\mathbf{m}^{BBE}$ , where  $\mathbf{m}^{BBE}[i]$  is the index of the task assigned to worker  $i$ . Suppose that there exists another equilibrium in which worker  $i$  can strictly gain. If worker  $i$  strictly gains in this equilibrium in comparison to the utility achieved in the bang-bang equilibrium, then it has to be matched to a task that is ranked higher than  $\mathbf{m}^{BBE}[i]$ . Let the task that worker  $i$  is assigned to in the new equilibrium be denoted as  $\mathbf{m}^{BBE}[j]$ . In this new equilibrium, we claim that at least one of the workers that were matched to a task ranked greater than or equal to  $\mathbf{m}^{BBE}[j]$  in the bang-bang equilibrium will now be matched to a task that is ranked strictly less than its match in the bang-bang equilibrium. Next, we justify this claim.

Consider the set of the workers who were matched to tasks ranked greater than or equal to  $\mathbf{m}^{BBE}[j]$  in the matching achieved in the operational phase in bang-bang equilibrium. Let us denote this set by  $\mathcal{Z}$ . Suppose that the number of workers in this set are  $N_1$ . In the new equilibrium in which  $i$  strictly gains, suppose that every worker in this set is matched to a task that is ranked strictly higher than or equal to its match in bang-bang equilibrium. First, note that if this supposition is not true, then the claim that there is at least one worker matched to a task ranked less than its match in the bang-bang equilibrium is already true.

Next, we assume that the supposition is true and proceed. Since the worker  $i$  is matched to  $\mathbf{m}^{BBE}[j]$ , the workers in  $\mathcal{Z}$  have to be matched to tasks that are ranked strictly higher than  $\mathbf{m}^{BBE}[j]$ . The total number of tasks that are ranked strictly higher than  $\mathbf{m}^{BBE}[j]$  are  $N_1 - 1$ . Therefore, if the supposition were true, then  $N_1$  workers have to be matched to at most  $N_1 - 1$  tasks. Hence, it is not possible to assign each of these workers to a strictly higher task (From Pigeonhole principle).

Consider the worker that has the highest ranking among all the workers that are assigned to a task that is ranked lower than their corresponding match in the bang-bang equilibrium. Let this worker be worker  $k$  and let the task assigned to  $k$  in the new equilibrium be  $\mathbf{m}^{BBE}[l]$ . In this new equilibrium, let the worker who is matched to  $\mathbf{m}^{BBE}[k]$  be worker  $r$ . Note that the rank of worker  $r$  has to be lesser than the rank of worker  $k$ . Next, we argue that in this new equilibrium, worker  $k$  must have used a strategy different than MTBB. More specifically, worker  $k$  either does not exert maximum effort on at least one task ranked ahead of  $\mathbf{m}^{BBE}[l]$  in the assessment phase or ranks  $\mathbf{m}^{BBE}[l]$  ahead of at least one task that was ranked higher in the preference list used in the MTBB strategy. Suppose that this is not the case, which means that worker  $k$  exerts maximum effort on all the tasks ahead of  $\mathbf{m}^{BBE}[l]$  and worker  $k$  also ranks all the tasks that were ahead of  $\mathbf{m}^{BBE}[l]$  to be higher than  $\mathbf{m}^{BBE}[l]$ .

Now since worker  $k$  exerts maximum effort on all the tasks ahead of  $\mathbf{m}^{BBE}[l]$ , it will be ranked ahead of  $r$  by  $\mathbf{m}^{BBE}[k]$  because it has a higher maximum output (rank of  $k$  is higher than  $k$  in the bang-bang equilibrium). We also know that worker  $k$  ranks  $\mathbf{m}^{BBE}[k]$  ahead of  $\mathbf{m}^{BBE}[l]$ . Therefore, the matching achieved is not stable w.r.t the preferences of the workers and the clients. This is a contradiction as the matching achieved must be stable as we use the G-S algorithm. Hence, in the new equilibrium, it must be that the worker must have either not exerted maximum effort on at least one task ranked ahead of  $\mathbf{m}^{BBE}[l]$  in the assessment phase or the preference list that it submits must rank  $\mathbf{m}^{BBE}[l]$  ahead of at least one task that was ranked higher in the preference list in the bang-bang equilibrium. Next, we analyze what happens if worker  $k$  instead uses the MTBB strategy in this case.

In this case, worker  $k$  will approach all the tasks ranked higher than  $\mathbf{m}^{BBE}[l]$  before  $\mathbf{m}^{BBE}[l]$ . We claim that the worker will be accepted by at least one task ranked higher

than or equal to  $\mathbf{m}^{BBE}[k]$ . Suppose that this is not the case, i.e. no task higher or equal to  $\mathbf{m}^{BBE}[k]$  accepts  $k$ . Observe that in the matching achieved in the bang-bang equilibrium, the number of tasks that are ranked higher or equal to  $\mathbf{m}^{BBE}[k]$  is the same as the number of workers with output greater than or equal to  $F(k)e_k^{max}$ . Based on this observation and the supposition above, it has to be true that at least one of the tasks ranked higher or equal to  $\mathbf{m}^{BBE}[k]$  accepts a worker with productivity lower than  $F(k)e_k^{max}$ . Let this task be denoted as  $\mathbf{m}^{BBE}[q]$ . We also know that  $\mathbf{m}^{BBE}[q]$  is also preferred more by worker  $k$  than its current match. Therefore, the matching that is achieved is not stable. This is a contradiction because the matching achieved by the G-S algorithm has to be stable. Hence, it must be true that if worker  $k$  uses MTBB strategy, then it is accepted by a task that is ranked at least as high as  $\mathbf{m}^{BBE}[k]$ .

We assume that no two tasks have the same qualities (follows from the assumption that  $g$  is strictly increasing). Therefore,  $\alpha F(k)^2 g(\mathbf{m}^{BBE}[k]) - C(k) > \alpha F(k)^2 g(\mathbf{m}^{BBE}[l]) - C(k)$ . If  $\alpha F(k)^2 g(\mathbf{m}^{BBE}[k]) - C(k) > 0$ , then worker  $k$  will exert maximum effort in the operational phase and thus deviating to the MTBB strategy will lead to a profitable deviation. This is a contradiction as the new equilibrium does not satisfy the incentive compatibility for all the workers. Therefore,  $\alpha F(k)^2 g(\mathbf{m}^{BBE}[k]) - C(k) \leq 0$ . In this case, worker  $k$  will have no incentive to exert maximum effort. Thus the deviation cannot be strictly profitable. However, since  $\alpha F(k)^2 g(\mathbf{m}^{BBE}[k]) - C(k) \leq 0$  we can claim in the new equilibrium, worker  $i$  will also exert no effort. If this claim is true, then it will imply that worker  $i$  cannot strictly gain in the new equilibrium, which is a contradiction to the original claim that we can find another equilibrium in which worker  $i$  strictly gains. We justify this claim next.

In the new equilibrium, worker  $i$  is matched to task  $\mathbf{m}^{BBE}[j]$ . We know that the rank of  $\mathbf{m}^{BBE}[k]$  is greater than or equal to  $\mathbf{m}^{BBE}[j]$ , which implies the following

$$g(\mathbf{m}^{BBE}[k]) > g(\mathbf{m}^{BBE}[j]) \quad (5.32)$$

We also know that the rank of worker  $k$  is more than the rank of worker  $i$  because in bang-bang equilibrium worker  $k$  is matched to a task that is ranked higher than the task assigned to worker  $i$ . Therefore,  $F(k)e_k^{max} \geq F(i)e_i^{max}$ . From Assumption 1, we can conclude

that

$$F(k)e_k^{max} \geq F(i)e_i^{max} \implies F(k) \geq F(i) \implies C(k) \leq C(i) \quad (5.33)$$

Based on (5.32) and (5.33), we have  $\alpha F(i)^2 g(\mathbf{m}^{BBE}[j]) - C(i) \leq 0$ , which implies that the utility achieved by worker  $i$  is 0. This establishes the claim. Hence, there cannot be another equilibrium in which a worker gains strictly in comparison to bang-bang equilibrium.

Next, we argue that there cannot be another equilibrium in which at least one worker gets a strictly lower payoff than in bang-bang equilibrium. We develop the proof for this on the same lines as the above. Note that for a worker to have strictly lower utility it has to be that the worker is matched to a task that is ranked strictly lesser than the task the worker is matched to in bang-bang equilibrium. Also, for the worker to have a strictly lower utility, it has to be true that the worker gets a strictly positive utility from its match in the bang-bang equilibrium. In this new equilibrium, we define the set of workers who are matched to tasks, which are strictly less in ranking in comparison to their match in the bang-bang equilibrium. From this set, we choose the worker with the highest rank. Let us denote this worker by  $s$ . Since the worker  $s$  has a positive utility from its match in the bang-bang equilibrium it has to be true that for  $s$ ,  $\alpha F(s)^2 g(\mathbf{m}^{BBE}[s]) - C(s) > 0$ . We can show that this worker  $s$  must have used a strategy different than MTBB. The proof of this is exactly on the same lines as the one for worker  $k$  given above. Based on the above proof for worker  $k$  it can also be shown that if worker  $s$  instead uses the MTBB strategy, then it will be matched to a task that has at least the same rank as  $\mathbf{m}^{BBE}[s]$ . Since  $\alpha F(s)^2 g(\mathbf{m}^{BBE}[s]) - C(s) >$  this deviation has to be profitable for worker  $s$ . Thus in this new equilibrium, incentive compatibility constraints are not satisfied that leads to a contradiction. Hence, there will be no equilibrium in which at least one worker gets strictly lower payoff than the bang-bang equilibrium. We can conclude that in every equilibrium each worker will have the same payoff as in the bang-bang equilibrium.

**Remark about Assumption 2.** In the above proof, we require that every worker has the same ranking over the tasks and vice-versa. Assumption 2 is a sufficient condition to

establish that the rankings are the same on both sides. Consider the case where Assumption 2 does not hold. However, if the rankings are still the same on both sides, then Theorem 14 continues to hold.  $\blacksquare$

### 5.6.5 Appendix E

**Proof of Theorem 15.** Consider a fixed payment rule with parameter  $\alpha$ . We consider matching rules in which each worker is finally matched to some client (in the limit) and thereafter there is no change in the matching. Given the fixed payment rule, it is easy to check that the only incentive compatible choice for worker  $i$ 's effort for task  $x$  is

$$e_i^{\max} I(\alpha F(i)^2 g(x) - C(i)) \quad (5.34)$$

Therefore, if worker  $i$  is matched to task  $x$ , then the long-run revenue generated by worker  $i$  is  $F(i)e_i^{\max} I(\alpha F(i)^2 g(x) - C(i))g(x)$ . Based on this we can write the expression for the maximum total long-run revenue that can be generated as follows.

$$\max_{\tilde{\mathbf{m}}} \sum_{i=1}^N F(i)e_i^{\max} I(\alpha F(i)^2 g(\tilde{\mathbf{m}}[i]) - C(i))g(\tilde{\mathbf{m}}[i]) \quad (5.35)$$

Observe that the indicator function in the objective above (5.35) increases with  $\alpha$ . Therefore, the above maximum value (5.35) is an increasing function of  $\alpha$ . Based on the constraint that  $\alpha \leq \frac{1}{2W^{\max}}$ , we can conclude the optimal value over all the payment rules in  $\mathcal{P}^F$  is given as

$$\max_{\tilde{\mathbf{m}}} \sum_{i=1}^N F(i)e_i^{\max} I\left(\frac{1}{2W^{\max}} F(i)^2 g(\tilde{\mathbf{m}}[i]) - C(i)\right) g(\tilde{\mathbf{m}}[i]) \quad (5.36)$$

Next, we simplify the above expression (5.36). We claim that (5.36) is simplified as follows.

$$\begin{aligned} \max_{\tilde{\mathbf{m}}} \sum_{i=1}^N F(i) e_i^{max} I\left(\frac{1}{2W^{max}} F(i)^2 g(\tilde{\mathbf{m}}[i]) - C(i)\right) g(\tilde{\mathbf{m}}[i]) = \\ \sum_{i=1}^N F(m_i) e_{m_i}^{max} I\left(\frac{1}{2W^{max}} F(m_i)^2 g(i) - C(m_i)\right) g(i) \end{aligned} \quad (5.37)$$

In the above expression (5.37),  $\{F(m_i) e_{m_i}^{max}\}_{i=1}^N$  corresponds to the set  $\{F(i) e_i^{max}\}_{i=1}^N$  ordered in the increasing order. In the matching in RHS above (5.37), worker  $m_i$  is matched to client  $i$ . We denote this matching as  $\hat{\mathbf{m}}$ . For consistency, we state that  $\hat{\mathbf{m}}(m_i) = i, ; \forall i \in \{1, .., N\}$ . Also, note that the optimal value for  $\alpha = \alpha^* = \frac{1}{2W^{max}}$ .

Next, we establish the above claim by deriving the RHS in (5.37).

First, we will establish a property that is a consequence of Assumption 2 and Assumption 3.

If worker  $i$  is matched to a task  $g(y)$  of quality greater than or equal to  $g(y) > g_u$ , then it will exert maximum effort. To prove this we need to show that the value of the indicator function in (5.34) is always one when  $g(y) > g_u$ .

$$\begin{aligned} I\left(\frac{1}{2W^{max}} F(i)^2 g(y) - C(i)\right) &\geq I\left(\frac{1}{2W^{max}} (f^{min})^2 g(y) - c^{max} \geq 0\right) \\ &\geq I\left(\frac{1}{2W^{max}} (f^{min})^2 g_u - c^{max} \geq 0\right) = 1 \end{aligned} \quad (5.38)$$

If worker  $i$  is matched to a task  $g(x)$  of quality less than or equal to  $g(x) < g_l$ , then it will exert no effort. To prove this we need to show that the value of the indicator function in (5.34) is always zero when  $g(x) < g_l$ .

$$\begin{aligned} I\left(\frac{1}{2W^{max}} F(i)^2 g(x) - C(i)\right) &\leq I\left(\frac{1}{2W^{max}} (f^{max})^2 g(x) - c^{min} \geq 0\right) \leq \\ I\left(\frac{1}{2W^{max}} (f^{max})^2 g_l - c^{min} \geq 0\right) &= 0 \end{aligned} \quad (5.39)$$

Let us assume that there is a  $\tilde{\mathbf{m}}^*$  different than  $\hat{\mathbf{m}}$ , which is optimal and leads to a strictly higher value for the objective, i.e. the total long-run revenue. We need to consider the following three cases:

Suppose that there exists at least one task that has a quality more than  $g_u$ . Therefore, we can find a task denoted as  $\hat{j}$  that satisfies the following condition:  $g(j) > g_u, \forall j \geq \hat{j}$

and  $g(j) < g_l$ ,  $\forall j \leq \hat{j}$ . For a given matching  $\tilde{\mathbf{m}}$ , we partition the workers into two sets: workers that are matched to tasks with quality greater than or equal to  $g(\hat{j})$  and the tasks with quality lesser than  $g(\hat{j})$ . Let the two sets for the matching  $\hat{\mathbf{m}}$  be denoted as  $S_1$  and  $S_2$ , where  $S_1$  is the set of workers matched with tasks of quality greater than or equal to  $g(\hat{j})$  and  $S_2$  is the set of workers matched with tasks with quality lesser than  $g(\hat{j})$ . Similarly, the two sets corresponding to the matching  $\tilde{\mathbf{m}}^*$  be  $R_1$  and  $R_2$ .

Suppose that  $R_1$  is not equal to  $S_1$ . Thus we can conclude that  $R_1 \cap S_2$  and  $R_2 \cap S_1$  is non-empty. Consider a worker  $i_1$  from the set  $R_1 \cap S_2$  and another worker  $i_2$  from the set  $R_2 \cap S_1$ . From the definition of the matching  $\hat{\mathbf{m}}$ , we can conclude that  $F(i_1)e_{i_1}^{max} < F(i_2)e_{i_2}^{max}$ . In the matching  $\tilde{\mathbf{m}}^*$ , worker  $i_1$  is matched to task greater than or equal to  $g(\hat{j})$  and worker  $i_2$  is matched to task less than  $g(\hat{j})$ . Suppose that we swap the worker  $i_1$  and worker  $i_2$  in the matching  $\tilde{\mathbf{m}}^*$ . The worker  $i_2$  will now exert maximum effort and worker  $i_1$  will now exert zero effort (This is due to the property that we established above). Since  $F(i_1)e_{i_1}^{max} < F(i_2)e_{i_2}^{max}$  the total long-run revenue will increase, thus contradicting the fact that  $\tilde{\mathbf{m}}^*$  is optimal. Therefore, the supposition that  $R_1$  is not equal to  $S_1$  cannot be true. So, we know that  $R_1 = S_1$  and  $R_2 = S_2$ . Next, we provide the expressions for the total long-run revenues under  $\tilde{\mathbf{m}}^*$  and  $\hat{\mathbf{m}}$ .

$$\sum_{i \in R_1} F(i)e_i^{max} g(\tilde{\mathbf{m}}^*[i]) \quad (5.40)$$

$$\sum_{i \in R_1} F(i)e_i^{max} g(\hat{\mathbf{m}}[i]) \quad (5.41)$$

Since  $\tilde{\mathbf{m}}^*$  is strictly better than  $\hat{\mathbf{m}}$ , it has to be true that the matching  $\tilde{\mathbf{m}}^*$  of the workers within the set  $R_1$  is different from  $\hat{\mathbf{m}}$ . Due to the claim that  $\tilde{\mathbf{m}}^*$  is strictly better than  $\hat{\mathbf{m}}$ , the following has to be true

$$\sum_{i \in R_1} F(i)e_i^{max} g(\tilde{\mathbf{m}}^*[i]) > \sum_{i \in R_1} F(i)e_i^{max} g(\hat{\mathbf{m}}[i]) = \sum_{j=\hat{j}}^N F(m_j)e_{m_j}^{max} g(j) \quad (5.42)$$

From rearrangement inequality, we know that

$$\sum_{j=\hat{j}}^N F(m_j)e_{m_j}^{max} g(j) \geq \sum_{i \in R_1} F(i)e_i^{max} g(\tilde{\mathbf{m}}^*[i]) \quad (5.43)$$

The condition above (5.43) contradicts (5.42).

From the above we get that the set of workers in  $R_1$  have to be matched to the same clients by both the matchings  $\tilde{\mathbf{m}}^*$  and  $\hat{\mathbf{m}}$ , which means  $\tilde{\mathbf{m}}^*$  cannot be strictly better than  $\hat{\mathbf{m}}$ .

Observe that the output of our matching rule is the same as  $\hat{\mathbf{m}}$  because all the workers rank the clients in the order of their qualities and the all the clients rank the workers based on their maximum outputs. This shows that the total long-run revenue achieved by the proposed mechanism is the same as in (5.37). ■

### 5.6.6 Appendix F

**Upper Bound on the Performance.** We write the maximum outputs of workers sorted in the increasing order as follows  $\{F(m_1)e_{m_1}^{max}, \dots, F(m_N)e_{m_N}^{max}\}$ , where  $m_x$  is the index of the worker with the  $x^{th}$  highest output.

**Proposition 2** *If Assumption 2 holds, then*

- *The maximum total long-run revenue generated when the workers are obedient and their productivities are known is  $\sum_{x=1}^N F(m_x)g(x)e_{m_x}^{max}$ .*
- *The total long-run revenue generated in the bang-bang equilibrium is  $\sum_{x \in \mathcal{S}_{max}}^N F(m_x)g(x)e_{m_x}^{max}$*

We write the set of outputs as follows  $\{F(1)e_1^{max}, \dots, F(N)e_N^{max}\}$  and we write the outputs sorted in the increasing order as follows  $\{F(m_1)e_{m_1}^{max}, \dots, F(m_N)e_{m_N}^{max}\}$ . Let us first establish the upper bound on the output. First, we will compute an upper bound on the total revenue that can be generated in one period. Clearly, the revenue generated is monotonic in the effort exerted by any worker. Since we are computing the upper bound here we will assume that each worker exerts maximum effort. Each worker  $i$  should exert maximum effort  $e_i^{max}$  otherwise the effort can always be increased to improve the output. Consider a general matching  $\mathbf{m}' : \mathcal{N} \rightarrow \mathcal{S}$ , where  $\mathbf{m}'[i]$  is the task allocated to worker  $i$ .

We can write the total revenue for this matching  $\mathbf{m}'$  as follows  $\sum_{i=1}^N F(i)e_i^{max}g(\mathbf{m}'[i])$ .

The inequality given below is a consequence of the rearrangement inequality.

$$\sum_{i=1}^N F(i) e_i^{\max} g(\mathbf{m}'[i]) \leq \sum_{i=1}^N F(m_i) e_i^{\max} g(i), \forall \mathbf{m}' \quad (5.44)$$

Therefore, we can also write the following for every matching rule  $\mathbf{m}$  and joint strategy  $\pi$  as defined in Section 5.2.

$$\sum_{i=1}^N r_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \boldsymbol{\pi}_i | \mathbf{m}) \leq \sum_{i=1}^N F(m_i) e_{m_i}^{\max} g(i)$$

The above holds true because

$$r_i(\mathbf{h}_0^t, \mathbf{h}_i^t, \boldsymbol{\pi}_i | \mathbf{m}) = F(i) g(\mathbf{m}(\mathbf{h}_0^t)) \boldsymbol{\pi}_i(\mathbf{h}_i^t) \leq F(i) g(\mathbf{m}(\mathbf{h}_0^t)[i]) e_i^{\max}$$

and  $\mathbf{m}' = \mathbf{m}(\mathbf{h}_0^t)$ . Note that the upper bound is same for each time slot, the same upper bound continues to hold for the long-run average too. Since the revenue is always more than the profit we use the same upper bound for profit (note that it won't be a tight bound). ■

### 5.6.7 Appendix G

**Example of a mechanism that only uses outputs and not the reports.** In this section, we show through an example that in some mechanisms (similar to existing mechanisms) that only operate on the performance of the workers, the workers may not be willing to exert maximum effort in the assessment phase and may instead try to manipulate the beliefs of the planner.

The planner observes the workers performance in the assessment phase and then matches the workers to the tasks as follows. We use the same notation  $\mathbf{W}^e$  as given in Section 5.2 for the outputs observed by the planner, where  $\mathbf{W}^e(i, x)$  is the output generated by worker  $i$  when matched to task  $x$ . The planner expects that the workers will continue to perform at the same level in the operational phase as well. Based on this expectation, the planner solves the following optimization to compute the best possible matching from the set of all the possible matching.

$$\max_{\tilde{\mathbf{m}}} \sum_{k=1}^N \mathbf{W}^e(k, \tilde{\mathbf{m}}[k]) g(\tilde{\mathbf{m}}[k]) \quad (5.45)$$

In the above,  $\tilde{\mathbf{m}}$  is a bijective mapping given as  $\tilde{\mathbf{m}} : \mathcal{N} \rightarrow \mathcal{S}$ .

Suppose there are two workers and two tasks. The productivities and the costs of the workers are given as  $\mathbf{F} = \begin{bmatrix} 6 & 6 \\ 5 & 4 \end{bmatrix}$ ,  $\mathbf{C} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$ . The maximum effort for every worker on each task is the same and is equal to 1. Suppose that task 1 pays more for than task 2, i.e.  $p(w, 1) > p(w, 2)$  for the same amount of output  $w$ .

In the assessment phase, if worker 1 and worker 2 both exert maximum effort on both task 1 and 2, then the mechanism will assign worker 1 to task 2 and worker 2 to task 1. Observe that both workers prefer task 1 to task 2. Therefore, worker 1 should instead exert no effort on task 2 and continue to exert maximum effort on task 1. In this case, worker 1 is assigned task 1 and worker 2 is assigned to task 2. Also, it is important to note that for worker 1 to know whether to reduce its effort on task 2 or not, it needs much more information about the other worker and its costs. The above example shows that how a worker can manipulate the beliefs through its performance in order to be assigned the task of its choice.

### 5.6.8 Appendix H

#### Details of the simulation setup in Section 5.3.

In the numerical simulation setup, we will consider the settings where the Assumption 2 holds. Half of the workers' productivities are independently drawn from a uniform distribution  $U \sim [0, w_1]$  and the rest of the workers are drawn independently from a uniform distribution  $U \sim [0, w_2]$ . If worker  $i$ 's productivity is given as  $F(i)$ , then the cost for exerting effort is defined as  $C(i) = C_1 - C_2 F(i)$ . The task qualities are drawn independently from a distribution  $t_1 + t_2 \times U[0, 1]$ . The maximum effort for all the workers is the same and given as  $e^{max}$ . The linear payment rule is defined as: each client pays the worker a fraction of the revenue generated  $\beta \in [0, 1]$  to the worker. The quadratic payment rule is defined as: each client  $x$  pays the worker  $\alpha w^2 g(x)$  for generating  $w$  output, where  $\alpha \leq \frac{1}{2 \max\{w_1, w_2\} e^{max}}$ . The number of workers are allowed to vary from 10 to 100. The other parameters are set as

follows  $w_1 = 20$ ,  $w_2 = 14$ ,  $C_1 = 2$ ,  $C_2 = 0.05$ ,  $t_1 = 2$ ,  $t_2 = 10$  and the number of draws are set to 10000.

### 5.6.9 Appendix I

#### Examples when long-run stability is not achieved by other mechanisms.

In this section, our aim is to show that mechanisms that share some similarities with our proposed mechanism need not be long-run stable. We describe one such mechanism next.

i) In the first  $N$  time slots, each workers works on one task exactly once (similar to the assessment phase in our mechanism). Each worker's average output is computed, where the average is taken uniformly across the tasks that the worker performs. The workers are finally matched to the tasks assortatively as follows. The worker with the highest average output gets the task with the highest quality and so on. Note that if a worker's productivity does not vary across the tasks, then this mechanism can be shown to be the same as our mechanism provided that it is combined with our proposed payment rule.

We will show that the above mechanism cannot always achieve long-run stability. Consider the following case with two workers and two tasks. The productivity of the workers, the cost of the workers and the maximum effort that can be exerted by the workers is given as  $\mathbf{F} = \begin{bmatrix} 6 & 2 \\ 5 & 4 \end{bmatrix}$ ,  $\mathbf{C} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$  and,  $e^{max} = 1$  respectively. The productivity of the tasks is given as  $g = [2 \ 1]$ . The payment rules for tasks 1 and 2 are  $p(x, 1)$  and  $p(x, 2)$ , where it is given that  $p(x, 1) > p(x, 2)$ . In this setting, for the mechanism described above, in the equilibrium the workers will exert maximum effort while being assessed. Finally, worker 1 is matched to task 2 and worker 2 is matched to task 1, which is not a long-run stable matching with respect to the equilibrium strategy. On the other hand, the proposed assessment-matching rule will lead to the long-run stable matching of worker 1 with task 1 and worker 2 with task 2.

There can be other mechanisms that match workers based on the planner's initial beliefs about the worker's types. Such mechanisms also lead to outcomes that are not long-run

stable. For instance, if the planner's beliefs about two workers are very similar, then the assignment for such workers is done randomly, thus leading to unstable outcomes.

### 5.6.10 Appendix J

#### Extensions

**General payment and cost functions.** Suppose the cost of exerting effort level  $e$  for a worker  $i$  on a task  $x$  is  $C(i, x)c(e)$ , where  $c$  is a convex increasing function of  $e$ . Suppose that the payment for producing output  $w$  on task  $x$  is  $b(w)g(x)$ , where  $b$  is an increasing function of  $w$ . For this cost and payment functions, we continue to use the proposed matching mechanism and we can show that most of the results that we presented extend to this case. We describe the weakly dominant equilibrium strategy of the worker. In the assessment phase, the worker exerts maximum effort on every task that it is assigned to. In the operational phase, the worker decides the optimal effort level to exert in order to maximize the utility. The main difference between the equilibrium strategy derived in the main part of the chapter and here is that the effort exerted by the workers does not exhibit a bang-bang structure. We can also show that the equilibrium strategy is coalitionally stable (the proof follows the same steps as in Theorem 13).

**Client selected payment rules.** In this section we expand on the discussion in Section 5.4. We make the following assumptions. Clients (task qualities) and workers (productivity, efforts and costs) are drawn i.i.d. from some distribution (known to everyone). For ease of exposition, we assume that the clients use a linear payment rule and the costs for effort are linear in the effort as well. We require the Assumption 1 and 2 to hold as well. The cost for exerting effort  $e_i$  for worker  $i$  is  $C(i)e_i$ . A client with task of quality  $g(x)$  uses a payment rule  $\alpha(x)Wg(x)$ , where  $\alpha(x)$  is the fraction that is set by the client,  $W$  is the output. As we described in the Section 3, the clients make the same payment per unit output to the workers, which implies  $\alpha(x)g(x)$  is the same value for all the clients. This means client with higher task quality pays a lower fraction. Therefore, the client with lowest quality will pay the highest fraction. Suppose the client with lowest quality say client  $x$  sets  $\alpha(x) = \zeta$ , where  $\zeta < 1$ .

Based on this the payment rules of the other clients are determined. For instance, a client  $y$  will pay  $\alpha(y) = \frac{g(x)\zeta}{g(y)}$ . Note that  $\alpha(y) < 1$  since  $g(x)$  is lowest quality task. The expected profit for client  $x$  is written as  $E[(1 - \zeta)F(m_x))g(x)e_{m_x}^{\max}I(F(m_x)g(x)\zeta - C(m_x) \geq 0)]$  where  $m_x$  is the index of worker that the client is matched with and the expectation is computed using the joint distribution of  $C(m_x)$ ,  $F(m_x)$ ,  $e_{m_x}^{\max}$ . Client  $x$  optimizes the above and obtains a  $\zeta^*$  as the optimal fraction. This determines the payment rules for all the other clients as well, as described above. The client  $y$  will pay  $\alpha(y) = \frac{g(y)\zeta^*}{g(x)}$ .

	Matching with transfer	Matching with strategic workers	Matching with incomplete info.	MH, AS
[GS62]	No	No	No	No, No
[SS71]	Yes	No	No	No, No
[SS00], [Bec73]	Yes	Yes	No	No, No
[Rot82], [IM15] [FIT16]	No	Yes	No	No, No
[Rot89]	No	Yes	Yes (no learning)	No, Yes
[RCI13], [LS09]	No	No	Yes (with learning)	No, Yes
[Hop12]	Yes	Yes	Yes (with learning)	No, Yes
[TSR12]	No	No	Yes (with learning)	No, Yes
[HZV12]	Yes	Yes	Yes (no learning)	Yes, No
[LD17]	No	Yes	Yes (with learning)	No, Yes
[AKL17]	No	Yes	Yes (with learning)	No, Yes
[AJK14]	Yes	Yes	Yes (with learning)	No, Yes
[Koc14]	No	Yes	Yes (with learning)	No, Yes
[KOS14]	Yes	No	Yes (with learning)	No, Yes
[DNG15]	Yes	Yes	Yes (no learning )	No, No
[Fis15]	Yes	Yes	No	Yes, No
[CTW03]	Yes	Yes	No	No, No
[XDV18]	Yes	Bounded rational (one-step foresight)	Yes (learning out of equilibrium)	Yes, Yes
[SXZ16]	Yes	Yes	Yes (no learning)	Yes, Yes
This work	Yes	Yes	Yes (with learning)	Yes, Yes

Table 5.1: Comparison of the different works. AS: Adverse Selection, MH: Moral Hazard

	Client	Worker	Planner
$T$	✓	✓	✓
Matching rule	✓	✓	✓
Payment rule	✓		✓
Cost $\mathbf{C}(i, x)$			
Productivity $\mathbf{F}(i, x)$			
Task quality $g(x)$			
Effort		✓	
Output	✓		✓

Table 5.2: Summary of the knowledge structure.

# CHAPTER 6

## Dynamic Resource Allocation Planning

### 6.1 Introduction

In the previous chapters, we studied the problem of resource sharing among multiple agents. In this chapter, we focus on a different resource allocation problem. We start with an agent who is given limited resources to monitor a dynamic environment/subject. Our objective is to develop plans for the agent to monitor the environment, i.e., how should the agent screen the stochastic process to gather information in a timely manner about the path of the process. The development of optimal screening plans is a computationally intractable task because of the exponentially large number of possible ways in which a stochastic process can evolve. In this chapter, we focus on developing a framework for this task, which is computationally tractable and has provable performance guarantees. We focus on the application of breast cancer screening framework. However, the framework is general and can be applied to other screening settings such as sensor scheduling. This chapter is based on my work in [AZS17].

Screening plays an important role in the diagnosis and treatment of a wide variety of diseases, including cancer, cardiovascular disease, HIV, diabetes and many others by leading to early detection of disease [Siu16] [CHH13] [WJO68]. For some diseases (e.g., breast cancer, pancreatic cancer), the benefit of early detection is enormous [JSX10] [RKV03]. Because screening – especially screening that requires invasive procedures such as mammograms, CT scans, biopsies, angiograms, etc. – imposes financial and health costs on the patient and resource costs on society, good screening policies should trade off benefit and cost [PK14]. The *best* screening policies should take into account that the trade-off between benefit and cost should be different for different diseases – but also for different patients – patients

whose features suggest that they are at high risk should be screened more often; patients whose features suggest that they are at low risk should be screened less often – and even different for the same individual at different points in time, as the perceived risk for that patient changes. Thus the best screening policies should account for the disease type and be *personalized* to the features of the patient and to the history of the patient (including the history of screening) [Lie07]. This chapter develops the first such personalized screening policies in a very general setting.

A screening policy prescribes what tests should/should not be done and when. Developing personalized screening policies that optimally balance the frequency of testing against the delay in the detection of the disease is extremely difficult for a number of reasons. (1) The onset and progression of different diseases varies significantly across the diseases. For instance, in [AAS12] the development of breast cancer is modeled as a stationary Markov process, in [Riz11] the development of HIV is modeled using a non-stationary survival process. The test outcomes observed over time may follow a non-stationary stochastic process that depends on the disease process upto that time and the features of the patient [SA16] [Riz11]. Existing works on screening [AAS12] [EAS14] are restricted to Markov disease processes and stationary Markov test outcome models, while this is not the case for many diseases and their test outcomes [RTV15] [Riz11] [SA16] [MDC06]. (2) The cost of not screening is the delay in detection of disease, which is not known. Hence the decision maker must act on the basis of beliefs about *future* disease states in addition to beliefs about the current disease state. (3) Patients can arrive at the scheduled time but may also arrive earlier on the basis of external information so the decision maker’s beliefs must take this external information into account. For instance, external information can be the development of lumps on breasts [BP01] [TGR02], or the development of a comorbidity [MWM92] [DCK11]. (4) Given models of the progression of the disease and of the external information, *solving* for that policy is computationally intractable in general.

This chapter addresses all of these problems. We provide a computationally effective procedure that solves for an *approximately* optimal policy and we provide bounds for the approximation error (loss in performance) that arises from using the approximately optimal

policy rather than the exactly optimal policy. Our procedure is applicable to many disease models such as dynamic survival models [Mil11] [Cro01] [LW06] [LW03] [Riz11] [MDC06], first hitting time models [AAS12] [EAS14] [LW03] [Cox92] [SWH11] [LW06].

Evaluating a proposed personalized screening policy using observational data is challenging. Observational data does not contain the counterfactuals: we cannot know what would have happened if a patient had been screened more often or an additional test had been performed. Instead, we follow an alternative route that has become standard in the literature [AAS12] [MIR08] [EAS14] [RTV15]: we learn the disease progression model from the observational data and then evaluate the screening policy on the basis of the learned model. We also account for the fact that the disease model may be incorrectly estimated. We show that if the estimation error and the approximation error are small, then the policy we construct is very close to the policy for the *correctly* estimated model.

In this chapter, a large breast cancer data set is used to illustrate the proposed personalized screening policy. We show that high risk patients are screened more often than low risk patients (personalization to the features of the patient) and that patients with bad test results are screened more often than patients with good test results (personalization to the dynamic history of the patient). The effect of these personalizations is that, in comparison with existing clinical policies, the policy we construct leads to large reductions (28-68%) in screening while achieving the same expected delays in disease detection. To illustrate the impact of the disease on the policy, we carry out a synthetic exercise across diseases, one for which the delay cost is linear and one for which the delay cost is quadratic. We show that the regime of operation (frequency of tests vs expected delay in detection) for the policies for the two costs are significantly different, thus highlighting the importance of choice of costs.

## 6.2 Model and Problem Formulation

**Time.** Time is discrete and the time horizon is finite; we write  $\mathcal{T} = \{1, \dots, T\}$  for the set of time slots.

**Patient features.** Patients are distinguished by a (fixed) *feature*  $x$ . We assume that

the features of a patient (age, sex, family history, etc.) are observable and that the set  $X$  of all patient features is finite.

**Disease model.** We model the disease in terms of the (*true physiological state*, where the state space is  $\mathcal{S}$ ). The disease follows a finite state stochastic process;  $\mathcal{S}^T$  is the space of *state trajectories*. The probability distribution over trajectories depends on the patient's features; for  $\vec{s} \in \mathcal{S}^T$ ,  $x \in X$  we write  $Pr(\vec{s}|x)$  for the probability that the state trajectory is  $\vec{s}$  given that the patient's features are  $x$ . We distinguish one state  $D \in \mathcal{S}$  as the *disease state*; the disease state  $D$  is absorbing.<sup>1</sup> Hence  $Pr(s(t) = D, s(t') \neq D) = 0$  for every time  $t$  and every time  $t' > t$ . The true state is hidden/not observed.<sup>2</sup>

Our stochastic process model of disease encompasses many of the disease models in the literature, including discrete time survival models. The (discrete time) Cox Proportional Odds model [Cox92], for instance, is the particular case of our model in which there are two states (Healthy  $H$  and Disease  $D$ ) and the probability distribution over state trajectories is determined from the hazard rates. To be precise: if  $\vec{s}$  is the state trajectory for which the disease state first occurs at time  $t_0$ , so that  $s(t) = H$  for  $t < t_0$  and  $s(t) = D$  for  $t \geq t_0$ ,  $\lambda(t|x)$  is the hazard at time  $t$  conditional on  $x$ , then  $Pr(\vec{s}|x) = [1 - \lambda(1|x)] \cdots [1 - \lambda(t_0-1|x)][\lambda(t_0|x)]$  and  $Pr(\vec{s}|x) = 0$  for all trajectories not having this form. Similar constructions show that other dynamic survival models [LW03] [Cox92] [SWH11] [RTV15] [MDC06] also fit in the rubric of the general model presented here.<sup>3</sup>

**External information.** The clinician performs tests that are informative about the patient's true state; in addition, external information may also arrive (for instance, patient self-examines breasts for lumps, patient discovers comorbidities, etc.). The patient observes an external information process modeled by a finite state stochastic process with state space

<sup>1</sup>The restriction to a single absorbing disease state is only for expositional convenience.

<sup>2</sup>For many diseases, it seems natural to identify states intermediate between Healthy and Disease. For instance, because breast lumps [TGR02] or colon polyps [EAS14] that are found to be benign may become malignant, it seems natural to distinguish at least one Risky state, intermediate between the Healthy and Disease states.

<sup>3</sup>We can encompass the possibility of competing risks (e.g., different kinds of heart failure) [Cro01] simply by allowing for multiple absorbing states.

$\mathcal{Y}$ ; the information at time  $t$  is  $Y(t) \in \mathcal{Y}$  (for instance,  $\mathcal{Y} = \{\text{Lump}, \text{No Lump}\}$ ). If the patient visits clinician at time  $t$ , then this external information  $Y(t)$  arrives to the clinician.  $Y(t)$  may be correlated with the patient's state trajectory through time  $t$  and the patient's features; we write  $Pr(Y(t) = y | \vec{s}(t), x)$  for the probability that the external information at time  $t$  is  $y \in \mathcal{Y}$ , conditional on the state trajectory through time  $t$  and features  $x$ . We assume that at each time  $t$  the external information  $Y(t)$  is independent of the past observations conditional on the state trajectory through time  $t$ ,  $\vec{s}(t)$ , and features  $x$ .

**Arrival.** The patient visits the clinician at time  $t$  if either (a) the information process  $Y(t)$  exceeds some threshold  $\tilde{y}$  or (b)  $t$  is the time for the next recommended screening (determined in the Screening Policies described below). The first visit of the patient to the hospital depends on the screening policy and the patient's features (See the description below). If the patient visits the clinician at time  $t$ , the clinician performs a sequence of tests and observes the results. For simplicity of exposition, we assume that the clinician performs only a single test, with a finite set  $\mathcal{Z}$  of outcomes. We write  $Pr(Z(t) = z | \vec{s}(t), x)$  as the probability that test performed at time  $t$  yields the result  $z$ , conditional on the (unobserved) state trajectory and the patient's features. We assume that the current test result is independent of past test results, conditional on the state trajectory and patient features. We also assume that current test result is independent of the external information conditional on the state trajectory through time  $t$  and the patient features. These assumptions are standard [AAS12] [Riz11]. We adopt the convention that  $z(t) = \emptyset$  if the patient does not visit the clinician at time  $t$  so that no test is performed. If the test outcome  $z \in \mathcal{Z}^+ \subset \mathcal{Z}$ , then the patient is diagnosed to have the disease. We assume that there are no false positives. If a patient is diagnosed to be in the disease state, then screening ends and treatment begins.

**Screening policies.** The *history* of a patient through time  $t$  consists of the trajectories of external information, test results and screening recommendations through time  $t$ . Write  $\mathcal{H}(t)$  for the set of histories through time  $t$  and  $\mathcal{H} = \bigcup_{t=0}^T \mathcal{H}(t)$  for the set of all histories. By convention  $\mathcal{H}(0)$  consists only of the empty history. A *screening policy* is a map  $\pi : X \times \mathcal{H} \rightarrow \{1, \dots, T\} \cup \{D\}$  that specifies, for each feature  $x$  and history  $h$  either the next screening time  $t^+$  or the decision that the patient is in the disease state  $D$  and so treatment should

begin. A screening policy  $\pi$  begins at time 0, when the history is empty, so  $\pi(x, \emptyset)$  specifies the first screening time for a patient with features  $x$ . (For riskier patients, screening should begin earlier.) Write  $\Pi$  for the space of all screening policies.

**Screening cost** We normalize so that the *cost of each screening* is 1. (We can easily generalize to the more general setting in which the clinician decides from multiple tests [50], and different tests have different costs.) The cost of screening is a proxy for some combination of the monetary cost, the resource cost and the health cost to the patient. We discount screening costs over time so if  $\mathcal{T}_s$  is the set of times at which the patient is screened then the *screening cost* is  $\sum_{t \in \mathcal{T}_s} \delta^t$ , where  $\delta \in (0, 1)$ .

**Delay cost.** If disease first occurs at time  $t_D$  (the *incidence time*) but is detected only at time  $t_d > t_D$  (the *detection time*) then the patient incurs a delay cost  $C(t_d - t_D; t_D)$ . If the disease is never detected the delay cost is  $C(T - t_D; t_D)$ . We assume that the delay cost function  $C : \{1, \dots, T\} \times \{1, \dots, T-1\} \rightarrow (0, \infty)$  is increasing in the first argument (the *lag* in detection) and decreasing in the second argument (the incidence time). The cost of delay is 0 if disease never occurs or occurs only at time  $t = T$ . Note that as soon as the disease is detected screening ends and treatment begins; in particular, there is a single unique time of incidence and a single unique time of detection. We allow for general delay costs because the impact of early/late detection on the probability of survival/successful treatment is different for different diseases.

**Expected costs.** If the patient features are  $x \in X$  then every screening policy  $\pi \in \Pi$  induces a probability distribution  $Pr(\cdot|x, \pi)$  on the space  $\mathcal{H}(T)$  of all histories through time  $T$  and in particular induces probability distributions  $\sigma = Pr(\cdot|x, \pi)$  on the families  $\mathcal{T}_s \subset 2^{\{1, \dots, T-1\}}$  of screening times and  $\beta = Pr((\cdot, \cdot)|x, \pi)$  on the pairs  $(t_D, t_d)$  of incidence time and detection time. The *expected screening cost* is  $E_\sigma \left[ \sum_{t \in \mathcal{T}_s} \delta^t \right]$  and the *expected delay cost* is  $E_\beta \left[ C(t_d - t_D, t_D) \right]$ . We provide a graphical model for the entire setup in the Appendix at the end of this chapter.

**Optimal screening policy.** The objective of the screening policy is to minimize a weighted sum of the screening cost and the delay cost; i.e. the optimal screening policy is

defined by

$$\operatorname{argmin}_{\pi \in \Pi} \left\{ (1-w) E_\sigma \left[ \sum_{t \in \mathcal{T}_s} \delta^t \right] + w E_\beta [C(t_d - t_D, t_D)] \right\} \quad (6.1)$$

The weight  $w$  reflects social/medical policy; for instance,  $w$  might be chosen to minimize cost subject to some accepted tolerance in delay (Further discussion on this is in the Experiments Section).

**Comment.** The standard decision theory methods [PGT03] [KLK11] [Yu06] [Kri17] used in screening [AAS12] [EAS14] cannot be used to solve the above problem. In standard partially observable Markov decision processes (POMDPs), the interval between two decision epochs (in this case, screening times) is fixed exogenously; in standard partially semi-Markov decision processes (POSMDPs), the time between two decision epochs is the sojourn time for the underlying core-state process. In our setting, the time between two decision epochs depends on the action (follow-up date), the external information process, and the state trajectory. In standard POMDPs (POSMDPs) the cost incurred in a decision epoch depend on the current state, while in the above problem the delay cost depends on the state trajectory. Moreover, in our setting the disease state trajectory is not restricted to a Markovian or semi-Markovian process.

### 6.3 Proposed Approach

**Beliefs.** By a *belief*  $\mathbf{b}$  we mean a probability distribution over the pairs consisting of state trajectories and a label  $l$  for the diagnosis:  $l = 1$  if the patient has been diagnosed with the disease,  $l = 0$  otherwise. By definition, a belief is a function  $\mathbf{b} : \mathcal{S}^T \times \{0, 1\} \rightarrow [0, 1]$  such that  $\sum_{\vec{s}, l} b(\vec{s}, l) = 1$  but it is often convenient to view a belief as a vector. Beliefs are updated using Bayesian updating every time there is a new observation (test outcomes, patient arrival, external information). Knowledge of beliefs will be sufficient to solve the optimization problem (6.1); see the Appendix at the end of this chapter. We write  $\mathcal{B}$  for the space of all beliefs.

**Bellman equations.** To solve (6.1) we will formulate and solve the Bellman equations. To this end, we begin by defining the various components of the Bellman equations. Fix a time  $t$ . The cost  $\tilde{C}$  incurred at time  $t$  depends on what happens at that time: i) if the patient (with diagnosis status  $l = 0$  before the test) is tested and found to have acquired the disease, the cost is the sum of the cost of testing and the cost of delay, ii) if the patient has the disease and is not detected, then the cost of delay is incurred in the time slot  $T$ , and iii) if the patient does not have the disease, then the cost incurred in time slot  $t$  depends on whether a test was done in time slot  $t$  or not. We write these cases below.

$$\tilde{C}(\vec{s}, t, z, l) = \begin{cases} wC(t - t_D; t_D) + (1 - w)\delta^t I(z \neq \emptyset) & t \leq T, l = 0, z \in \mathcal{Z}^+ \\ wC(T - t_D; t_D) & t = T, l = 0 \\ (1 - w)\delta^t I(z \neq \emptyset) & \text{otherwise} \end{cases} \quad (6.2)$$

A *recommendation plan*  $\boldsymbol{\tau} : \mathcal{Z} \rightarrow \mathcal{T}$  maps the observation  $z$  at the end of time slot  $t$  to the next scheduled follow-up time. Note that the recommendation plan is defined for a time  $t$  and is different than the policy. Denote the probability distribution over the observations (test outcome  $z$ , duration to the next arrival  $\tilde{\tau}$ , and the external information at the next arrival time  $y$ ) conditional on the current belief  $\mathbf{b}$  and the current recommendation plan  $\boldsymbol{\tau}$  by  $Pr(z, y, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, x)$ . The belief  $\mathbf{b}$  is updated to  $\hat{\mathbf{b}}$  in the next arrival time  $\tilde{\tau}$  based on the observations, current recommended plan and the current beliefs using Bayesian updating as  $\hat{b}(\vec{s}, l) = Pr(\vec{s}, l | \mathbf{b}, \boldsymbol{\tau}, y, z, \tilde{\tau}, x)$ .

The optimal values for the objective in (2) starting from different initial beliefs can be expressed in terms of a value function  $V : \mathcal{B} \times \{1, \dots, T + 1\} \rightarrow \mathbb{R}$ . The value function at time  $t$  when the patient is screened solves the Bellman equation:

$$V(\mathbf{b}, t) = \max_{\boldsymbol{\tau}} \left[ \sum_{\vec{s}, l, z} -b(\vec{s}, l) Pr(z | \vec{s}, x) [\tilde{C}(\vec{s}, t, z, l)] + \sum_{z, \tilde{\tau}, y} Pr(z, y, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, x) V(\hat{\mathbf{b}}, t + \tilde{\tau}) \right] \quad (6.3)$$

We define  $V(\mathbf{b}, T + 1) = 0$  for all beliefs. Note that the computation of the first term in the RHS of (6.3) has a worst case computation time of  $|\mathcal{S}|^T$ . Therefore, solving for exact  $V(\mathbf{b}, T)$  that satisfies (6.3) is computationally intractable when  $T$  is large. Next, we derive

a useful property of the value function. (The proof of this and all other results are in the Appendix at the end of the chapter).

**Lemma 3** *For every  $t$ , the value function  $V(\mathbf{b}, t)$  is the maximum of a finite family of functions that are linear in the beliefs  $\mathbf{b}$ . In particular, the value function is convex and piecewise linear.*

The above property was shown for POMDPs in [SS73], we use the same ideas to extend it to our setup.

### 6.3.1 Constructing the Exactly Optimal Policy

Every linear function of beliefs is of the form  $\boldsymbol{\alpha}^* \mathbf{b}$  for some vector  $\boldsymbol{\alpha}$ . (We view  $\boldsymbol{\alpha}, \mathbf{b}$  as column vectors and write  $\boldsymbol{\alpha}^*$  for the transpose.) Hence Lemma 3 tells us that there is a finite set of vectors  $\Gamma(t)$  such that  $V(\mathbf{b}, t) = \max_{\boldsymbol{\alpha} \in \Gamma(t)} \boldsymbol{\alpha}^* \mathbf{b}$ . We refer to  $\Gamma(t)$  as the *set of alpha vectors*. In view of Lemma 3, to determine the value functions we need only determine the sets of alpha vectors. If we substitute the expression  $V(\mathbf{b}, t) = \max_{\boldsymbol{\alpha} \in \Gamma(t)} \boldsymbol{\alpha}^* \mathbf{b}$  into (6.3), then we obtain a recursive expression for  $\Gamma(t)$  in terms of  $\Gamma(t+1)$ . By definition, the value function at time  $T+1$  is identically 0 so  $\Gamma(T+1) = \{\mathbf{0}\}$ , where  $\mathbf{0}$  is the  $|\mathcal{S}^T \times \{0, 1\}|$  dimensional zero vector, so we have an explicit starting point for this recursive procedure. There is an optimal action associated with each alpha vector. The action corresponding to the optimal alpha vector at a certain belief is the output of the optimal action given that belief, and so constructing the sets of alpha vectors yields the optimal policy; the details of the algorithm are in the Algorithm 4 in the Appendix. Unfortunately, the algorithm to compute the sets of alpha vectors is computationally intractable (as expected). We therefore propose an algorithm that is tractable to compute an approximately optimal policy.

### 6.3.2 Constructing the Approximately Optimal Policy

Point-Based Value Iteration (PBVI) approximation algorithms are known to work well for standard POMDPs [PGT03]. These algorithms rely on choosing a finite set of belief vectors

and constructing alpha vectors for these belief vectors and their success depends very much on the efficient construction of the set of belief vectors. The standard approaches [PGT03] for belief construction are not designed to cope with settings like ours when beliefs lie in a very high dimensional space; in our setup belief has  $|\mathcal{S}^T \times \{0, 1\}|$  dimensions. In Algorithm 1, 2, 3 (pseudo-code in the Appendix at the end of the chapter), we first construct a lower dimensional belief space by sampling trajectories that are more likely to occur for the disease and then sampling the set of beliefs in the lower dimensional space that are likely to occur over the course of various screening policies. The key steps for our approach, which we refer to as DPSCREEN (DPSCREEN is comprised of Algorithm 1, 2, 3) are

- 1. Sample typical physiological state trajectories.** Sample a set  $\tilde{\mathcal{S}} \subset \mathcal{S}^T$  of  $K$  physiological trajectories from the distribution  $Pr(\vec{s}|x)$ .
- 2. Construct the set of reachable belief vectors.** Say that a belief vector  $\mathbf{b}_2$  is *reachable* from the belief vector  $\mathbf{b}_1$  if it can be derived by Bayesian updating on the basis of some underlying screening policy. We construct the sets of belief vectors that can be reached under different screening policies. For the first time slot, we start with a belief vector that lies in the space  $\tilde{\mathcal{S}} \times \{0, 1\}$  given as  $Pr(\vec{s}|x)/Pr(\tilde{\mathcal{S}}|x)$ ,  $\forall \vec{s} \in \tilde{\mathcal{S}}, l = 0$ . For subsequent times, we select the beliefs that are encountered under random exploration of the actions (recommendation of future test dates). In addition to using random exploration, we can choose actions determined from a set of policies such as the clinical policies used in practice [OFE15] [NTN09] [ZLK08] to construct the set of reachable belief vectors.

Denote the set of belief vectors constructed at time  $t$  by  $\bar{B}[t]$  and the set of all such beliefs as  $\bar{B} = \{\bar{B}[t], \forall t\}$ . We carry out point-based value backups on these beliefs  $\bar{B}$  (see Algorithm 2, 3 in the Appendix at the end of this chapter), to construct the alpha vectors and thus the approximately optimal policy.

**Computational complexity.** The policy computation requires  $\mathcal{O}(T^3 BK|\mathcal{Y}||\mathcal{Z}|)$  steps, where  $B = \max_t |\bar{B}[t]|$  is the maximum over the number of points sampled by the Algorithm 1 for any time slot  $t$ . The complexity can be reduced by restricting the space of actions; e.g. by bounding the amount of time allowed between successive screenings. Moreover, the

proposed algorithms can be easily parallelized (many operations carried inside the iterations in Algorithm 2, 3 can be done parallel), thus significantly reducing computation time.

**Approximation error.** Because we only sample a finite number of trajectories, the policy we construct is not optimal but we can bound the loss of performance in comparison to the exactly optimal policy and hence justify the term “approximately optimal policy.” Define the *approximation error* to be the difference between the value achieved by the exact optimal policy (solution to (6.1)) and the value achieved by the approximately optimal policy (output from Algorithm 2, 3). As a measure of the density of sampling of the belief simplex we set  $\Omega(\bar{B}) = \zeta \max_{t \in \mathcal{T}} \max_{\mathcal{B}} \min_{\mathbf{b} \in \bar{B}[t]} \|\mathbf{b} - \mathbf{b}'\|_1$ , where  $\zeta$  is a constant that measures the maximum expected loss that can occur in one time slot. We make a few assumptions for the proposition to follow. The cost for delay is  $C(t_d - t_D; t_D) = c(t_d - t_D)\delta^{t_D}$ , where  $c(d)$  is a convex function of  $d$ . The test outcome is accurate, i.e. no false positives and no false negatives. The maximum screening interval is bounded by  $W < T$ . The time horizon  $T$  is sufficiently large. We show that the loss of performance is bounded by the sampling density.

**Proposition 3** *The approximation error is bounded above by  $\Omega(\bar{B})$ .*

### 6.3.3 Robustness

**Estimation error.** To this point, it has been assumed that the model parameters are known. In practice, the model parameters need to be estimated using the observational data. In the next section, we will give a concrete example of how we estimate these parameters using observational data for breast cancer. Here we discuss the effect of error in estimation. Suppose that the model being estimated (true model) is  $\mathbf{m}' \in \mathbf{M}$ , where  $\mathbf{M}$  is the space of all the possible models (model parametrizations) under consideration. (We assume that the probability distribution of the physiological state transition, the patient’s self-observation outcomes, and the clinician’s observation outcomes are continuous on  $\mathbf{M}$ .) Write  $\mathbf{L} = \mathbf{M} \times \mathcal{B}$  for the joint space of models and beliefs. Let the estimate of the model be  $\hat{\mathbf{m}}$ . Let us assume that for every model in  $\mathbf{M}$  the solution to (6.1) is unique. Therefore, we can define a mapping  $\boldsymbol{\tau}^* : \mathbf{L} \times \mathcal{Z} \times \mathcal{T} \rightarrow \mathcal{T}^{|\mathcal{Z}|}$ , where  $\boldsymbol{\tau}^*(\mathbf{l}, z, t)$  is the optimal recommended screening time at  $\mathbf{l}$ , at

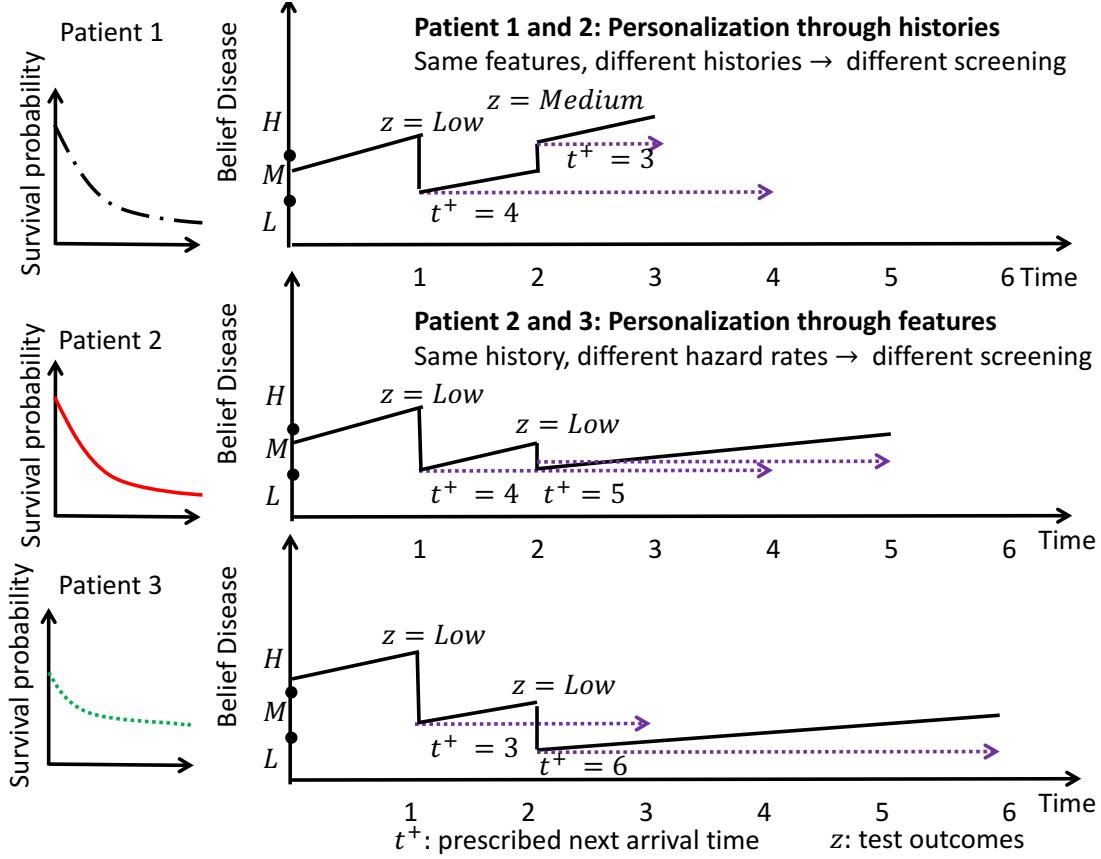


Figure 6.1: Illustration of dynamic personalization.

time  $t$  following  $z$ . For a fixed model  $\mathbf{m}$ ,  $\tau^*((\mathbf{m}, \mathbf{b}), z, t)$  is the maximizer in (6.3).

**Theorem 16** *There is a closed lower dimensional set  $\mathbf{E} \subset \mathbf{L}$  such that the function  $\tau^*$  is locally constant on the complement of  $\mathbf{E}$ .*

Theorem 16 implies that, with probability 1, if the model estimate  $\hat{\mathbf{m}}$  and the true model  $\mathbf{m}'$  are sufficiently close, then the actions recommended by the exactly optimal policies for both models are identical. Therefore, the impact of estimation error on the exactly optimal policy is minimal. However, we construct approximately optimal policies. We can combine these conditions with Proposition 3 to say that if the approximation error  $\Omega(\bar{B})$  goes to zero, then the approximately optimal policy (for  $\hat{\mathbf{m}}$ ) will also converge to the exactly optimal policy for true model  $\mathbf{m}'$ .

**Personalization.** Fig. 6.1 provides a graphical representation of the way in which

DPSCREEN is personalized to the patients. We consider three Patients. The disease model for each patient is given by the *ex ante* survival curve (the probability of not becoming diseased by a given time). As shown in the graphs, the survival curves for Patients 1, 2 are the same; the survival curve for Patient 3 begins below the survival curve for Patients 1, 2 but is flatter and so eventually crosses the survival curve for Patients 1, 2. All three patients are screened at date 1; for all three the test outcome is  $z = \text{Low}$ . Hence the belief (risk assessment) for all three patients decreases. As a result, Patients 1, 2 are scheduled for next screening at date 4 but Patient 3, who has a lower *ex ante* survival probability, is scheduled for next screening at date 3. Thus, the policy is *personalized to the ex ante risk*. However, at date 2, all three patients experience an external information shock which causes them to be screened early. The test outcome for Patient 1 is  $z = \text{Medium}$  so Patient 1 is assessed to be at higher risk and is scheduled for next screening at date 3; the test outcome for Patient 2 is  $z = \text{Low}$  so Patient 2 is assessed to be at lower risk and is scheduled for next screening at date 5. Thus the policy is *personalized to the dynamic history*. The test outcome for Patient 3 is  $z = \text{Low}$  and Patient 3's *ex ante* survival probability is higher so Patient 3's risk is assessed to be very low, and Patient 3 is scheduled for next screening at date 6. Thus the policy adjusts to time-varying model parameters.

## 6.4 Illustrative Experiments

Here we demonstrate the effectiveness of our policy in a real setting: screening for breast cancer.

**Description of the dataset.** We use a de-identified dataset of 45,000 patients aged 60–65 who underwent screening for breast cancer. For most individuals we have the following associated features: age, the number of family members with breast cancer, weight, etc. Each patient had at least one mammogram; some had several. (In total, there are 84,000 mammograms in the dataset.) The outcome of a mammogram is given in the form of a BIRADS (Breast Imaging Report and Data System) score  $\{1, 2, 3, 4, 4A, 4B, 4C, 5, 6\}$ , The outcome was considered positive if the BIRADS scores is 4 or above, in which case a biopsy

was performed. 98% of patients had BIRADS scores below 4; of the 900 who had BIRADS scores 4 or above, 426 patients were diagnosed with cancer.

**Model description.** We model the disease progression using a two-state Markov model:  $\mathcal{S} = \{H, D\}$  ( $H$  = Healthy,  $D$  = Disease/Cancer). Given patient features  $x$ , the initial probability of cancer is  $p_{in}(x)$  and the probability of transition from the  $H$  to  $D$  is  $p_{tr}(x)$ . The external information  $Y$  is the size (perhaps 0) of a breast lump, based on the patient's own self-examination. In view of the universal growth law for tumor described in [23], we model  $Y(t) = g(t) + \epsilon(t)$ , where  $g(t) = (1 - e^{-\iota(t-t_s)})I(t > t_s)$  is the size of the tumor and  $t_s$  is the time at which patient actually develops cancer (the lump exists),  $\epsilon(t)$  is a zero mean white noise process with variance  $\sigma^2$  and  $I()$  is the indicator function. If the lump size  $Y$  exceeds the threshold  $\tilde{y}$ , then the patient visits the clinician, where tests are carried out. The set of test outcomes is  $\mathcal{Z} = \{\emptyset, 1, 2, 3\}$ , where  $z = \emptyset$  when no test is done,  $z = 1$  when the mammogram is negative and no biopsy is done,  $z = 2$  when the mammogram is positive and the biopsy is negative,  $z = 3$  when both mammogram and biopsy is positive.

**Model estimation.** We use the specificity and sensitivity for the mammogram from [AAS12]. Each patient has a different (initial) risk for developing cancer; we compute the risk scores using the Gail model [GBB89], which we use as the feature  $x$ . We assumed  $p_{in}(x)$  and  $p_{tran}(x)$  are logistic functions of  $x$ . We use standard Markov Chain Monte Carlo methods to estimate these functions  $p_{in}(x)$  and  $p_{tran}(x)$ . We use independent normal priors for the parameters of the functions  $p_{in}(x)$  and  $p_{tr}(x)$ . We compute the posterior (up to a constant) of the parameters in terms of the likelihood of the observed data (described above). We estimate the posterior distribution using the Metropolis Hastings method with a Gaussian random walk as the proposal distribution.

We assume that each woman has one self-examination per month [BP01] [TGR02]. We use the value  $\iota = 0.9$  as stated in [GDD03]. We estimate the parameters for the self-examinations  $\sigma = 0.43$  and  $\tilde{y} = 1$  on the basis of the values of sensitivity and specificity for the self-examination from the literature [43]. In the comparisons to follow, we will also analyze the setting when there are no self-examinations. We divide the population into two risk groups; the Low risk group consists of patients whose prior estimated risk of developing

cancer within five years is less than 5%; the High risk group consists of patients whose prior estimated risk exceeds 5%.

**Performance metrics, Objective and Benchmarks.** Our objective is to minimize the number of screenings subject to a constraint on expected delay cost. We assume the delay cost is linear:  $C(t_d - t_D, t_D) = t_d - t_D$ . To derive the solution to this constrained problem from construction, which minimizes the weighted sum of screening cost and delay cost, we solve the weighted problem for some weight  $w$ , and then tune  $w$  to select the policy that minimizes the number of screenings subject to a constraint on expected delay cost. For comparison purposes, we take the constraint on expected delay cost to be the expected delay that arises from current clinical practice (annual screening in the US [OFE15] [NTN09], biennial screening in some other countries [KBN07]). (Because our objective is to minimize the *number* of screenings, we take the cost of each screening to be 1, whether or not a biopsy is performed.)

**Comment.** At this point, we remind that existing frameworks [AAS12] [EAS14] [RTV15] cannot be used to solve for the optimal screening policy in the above setup because: i) the costs incurred (delay) depends on the state trajectory and not just the current state, and ii) the lump growth model and the patient's self-examination of the lump is not easy to incorporate in these works.

**Comparisons with clinical screening policies.** We compare our constructed policies (for the two groups), with and without self-examination, in terms of three metrics: i)  $E[N|R]$ : the expected number of tests per year, conditional on the risk group; ii)  $E[\Delta|R]$ : the expected delay, conditional on the risk group; iii)  $E[\Delta|R, D]$ : the expected delay, conditional on the risk group and the patient actually developing cancer. Because  $E[\Delta|R]$  is the expected *unconditional delay*, it accounts for patients who do not develop cancer as well as for patients who do have cancer; because most patients do not develop cancer,  $E[\Delta|R]$  is small. We show the comparisons with the annual policies in Table 6.1 and with biennial policies in Table 6.2.

In Table 6.1 we compare the performance of DPSCREEN (with and without self examination) for Low and High risk groups against the current clinical policy of annual screening. For

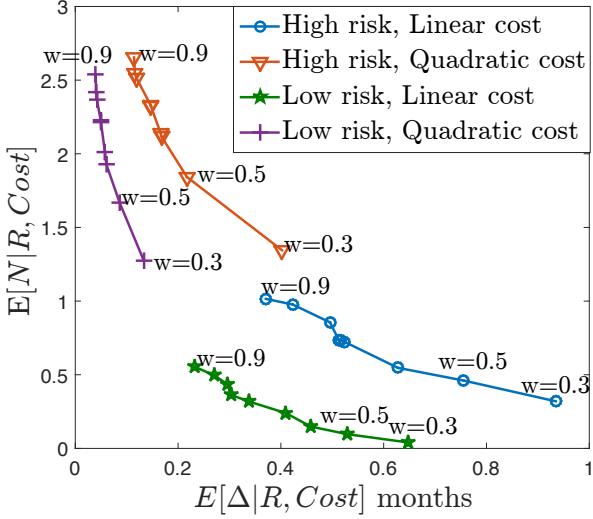


Figure 6.2: Impact of the type of disease.

both risk groups, the proposed policy achieves approximately the same expected delay as the benchmark policy while doing many fewer tests (in expectation). With self-examinations, the expected reduction in number of screens is 57-68% (depending on risk group); even without self-examinations, the expected reduction in number of screens is 28-45% percent (depending on risk group).

In Table 6.2 we compare the performance of DPSCREEN (with and without self examination) for Low and High risk groups against the current clinical policy of biennial screening. For both risk groups, the proposed policy achieves approximately the same expected delay as the benchmark policy while doing many fewer tests (in expectation). With self-examinations, the expected reduction in number of screens is 56-58% (depending on risk group); even without self-detection, the expected reduction in number of screens is 16-24% percent (depending on risk group).

In Table 6.3 we contrast the difference in DPSCREEN across the two risk groups. To keep the comparison fair, we fix the tolerance in the delay to a fixed value. The proposed policy is personalized as it recommends significantly fewer tests to the low risk patients in

Table 6.1: Comparison of the proposed policy with annual screening for both high and low risk group.

Risk Group	Metrics	DPSCREEN with self-exam	DPSCREEN w/o self-exam	Annual
Low	$E[N R]$	0.32, 0.23, 9.2	0.55, 0.23, 9.2	1, 0.24, 9.6
	$E[\Delta R], E[\Delta R, D]$			
High	$E[N R]$	0.43, 0.50, 6.7	0.72, 0.52, 7.07	1, 0.52, 7.07
	$E[\Delta R], E[\Delta R, D]$			

Table 6.2: Comparison of the proposed policy with biennial policy.

Risk Group	Metrics	DPSCREEN with self-exam	DPSCREEN w/o self-exam	Biennial
Low	$E[N R]$	0.21,	0.42, 0.29, 12.36	0.5, 0.29, 12.36
	$E[\Delta R], E[\Delta R, D]$	0.29, 12.36		
High	$E[N R],$	0.22,	0.38, 0.90, 12.13	0.5, 0.88, 11.8
	$E[\Delta R], E[\Delta R, D]$	0.90, 12.13		

contrast to the high risk patients.

**Impact of the type of disease.** We have so far considered breast cancer as an example and assumed linear delay costs. For some diseases (such as Pancreatic cancer [SFM99] [RKV03]) the survival probability decreases very quickly with the delay in detection and therefore it might be reasonable to assume a cost of delay that is strictly convex (such as quadratic costs) in delay time for some disease. In Fig. 6.2, we show that for a fixed risk group and for the same weights the policy constructed using quadratic costs is much more aggressive in testing. Moreover, the regime of operation of the policy (the points achieved by the policy in the 2-D plane  $E[N|R, Cost]$  vs  $E[\Delta|R, Cost]$ ) can vary a lot depending on the choice of cost function even though the same weights are used. Therefore, the cost should be chosen based on the disease.

Table 6.3: Comparison of the proposed policies across different risk groups.

Risk Group	DPSCREEN with self-exam $E[N R], E[\Delta R], E[\Delta R, D]$	DPSCREEN w/o self-exam $E[N R]E[\Delta R], E[\Delta R, D]$
Low	0.12, 0.33, 13.7	0.32, 0.33, 13.7
High	0.80, 0.35, 4.73	1.09, 0.35, 4.73

## 6.5 Related Works

In Section 6.2, following the equation (6.1), we compared our methods with frameworks to some general frameworks in decision theory [PGT03] [KLK11] [Yu06] [Kri17]. Next, we compare with other relevant works.

**Screening frameworks for different diseases in operations research:** Many works have focused on optimizing population-based screening schedules, which are not personalized (See [AAE10] and references therein). In [AAS12] [EAS14] the authors develop personalized POMDP based screening models. The underlying disease evolution (breast and colon cancer) is assumed to follow a Markov process. External information process such as self-exams and the test outcomes over time are assumed to follow a stationary i.i.d process given the disease process. In [RTV15] authors develop personalized screening models based on principles of Bayesian design for maximizing information gain (based on [VK92]). The underlying disease model (cardiac disease) is a dynamic (two-state) survival model and the cost of misdetection is a constant and does not depend on the delay. The test outcomes are modeled using generalized linear mixed effects models, and there is no external information process. To summarize, all of the above methods rely on very specific models for their disease, test outcomes, and external information, while our method imposes much less restrictions on the same.

**Screening frameworks for different diseases in medical literature:** The Medical research literature on screening (e.g., Cancer Intervention and Surveillance Modelling Network, US preventive services task force, etc.) relies on stochastic simulation based

methods: fix a disease model and a set of screening policies to be compared; for each policy in the set, simulate outcome paths from the model; compare across the set of policies [VFC14] [TKS16] [LBO99] [ZLK08] [FCF00]. The clinical guidelines for screening issued by the US preventive services task force [ZLK08] [WLL08] for colon cancer cancers are created based on the MISCAN-COLON [LBO99] model for colon cancer. Simulations were carried out to compare different screening policies suggested by experts for that specific disease model- MISCAN-COLON. This approach allows more realistically complex models but it only compares a fixed set of policies, all of which may be far from optimal.

**Controlled Sensing:** In controlled sensing [Kri16] [AV16] [Kri16] the problem of sensor scheduling requires deciding which sensor to use and when; this problem is similar the personalized screening problem studied here. In these works [Kri16] [AV16] [Kri16], the main focus is to exploit (or derive) structural properties of the process being sensed and the cost functions such that the exactly optimal sensing schedule is easy to characterize and compute. The structural assumptions such as the process that is sampled is stationary and Markov make these works less suited for personalized screening.

## 6.6 Discussion on Generality of the Proposed Framework

In the entire discussion so far, we focused on the application of screening for timely detection of disease. The proposed framework is developed for general cost functions and a general stochastic process. The proposed framework can be applied to other problems such as controlled sensing, sensor scheduling and other stopping time problems, where one needs to gather information from a stochastic process using limited resources.

## 6.7 Conclusion

In this chapter, we develop a novel methodology for constructing personalized screening policies that balance the cost of screening against the cost of delay in gathering important information. In this work, we focus the application of disease screening such as breast

cancer screening. The disease is modeled as an arbitrary finite state stochastic process with an absorbing disease state. Our method incorporates the possibility of external information, such as self-examination or discovery of co-morbidities, that may trigger arrival of the patient to the clinician in advance of a scheduled screening appointment. We use breast cancer data to develop the disease model. In comparison with current clinical policies, our personalized screening policies reduce the number of screenings performed while maintaining the same delay in detection of disease.

## 6.8 Appendix

### 6.8.1 Appendix A

In this section, we define the belief update expressions that are required in Algorithm 1. Compute the belief updates  $\forall \vec{s}, l \in \tilde{\mathcal{S}} \times \{0, 1\}$

$$\Theta(b, z)[\vec{s}, l] = \frac{Pr(\vec{s}, l, z|x)}{Pr(z|x)} = \frac{\sum_{\tilde{l}} Pr(\vec{s}, l, z, \tilde{l}|x)}{\sum_{\vec{s}} \sum_{\tilde{l}} Pr(\vec{s}, l, z, \tilde{l}|x)} = \frac{\sum_{\tilde{l}} b(\vec{s}, \tilde{l}) Pr(z|\vec{s}) Pr(l|\tilde{l}, z, x)}{\sum_{\vec{s} \in \tilde{\mathcal{S}}} \sum_{\tilde{l}} b(\vec{s}, \tilde{l}) Pr(z|\vec{s}, x) Pr(l|\tilde{l}, z, x)} \quad (6.4)$$

$$\begin{aligned} \Phi(b, y, \tilde{\tau}, \tau, t)[\vec{s}, l] &= \frac{Pr(\vec{s}, l, y, \tilde{\tau}|\tau, t, x)}{\sum_{\vec{s}, l} Pr(\vec{s}, l, y, \tilde{\tau}|\tau, t, x)} = \frac{Pr(\vec{s}, l, y, \tilde{\tau}|\tau, t, x)}{\sum_{\vec{s}, l} Pr(\vec{s}, l, y, \tilde{\tau}|\tau, t, x)} \\ &= \frac{b(\vec{s}, l) Pr(y, \tilde{\tau}|\vec{s}, \tau, t, x)}{\sum_{\vec{s}, l} b(\vec{s}, l) Pr(y, \tilde{\tau}|\vec{s}, \tau, t, x)} \end{aligned} \quad (6.5)$$

For all  $\tilde{\tau} \leq \tau$  we have

$$Pr(y, \tilde{\tau}|\vec{s}, \tau, t, x) = Pr(\{Y(s) \leq \tilde{y}; \forall s \leq t + \tilde{\tau}\}, Y(t + \tilde{\tau}) = y|\vec{s}, \tau, t, x) \quad (6.6)$$

For all  $\tilde{\tau} > \tau$  we have

$$Pr(y, \tilde{\tau}|\vec{s}, \tau, t, x) = 0 \quad (6.7)$$

---

**Algorithm 1** Constructing the Belief Sets

---

**Initialization:**  $r = \mathbf{0}_T$  a  $T$  dimensional zero vector,  $\bar{B}$  is the array to store the belief vectors that can be achieved at the  $T$  time instances

Sample  $K$  iid samples of the trajectory  $Pr(\vec{s}|x)$  to form the set  $\tilde{\mathcal{S}}$

**for**  $\vec{s} \in \tilde{\mathcal{S}}$  **do**

$$\bar{B}[1, 0](\vec{s}, l = 0) = Pr(\vec{s})/Pr(\tilde{\mathcal{S}})$$

**for**  $\vec{s} \in \tilde{\mathcal{S}}$  **do**

**for**  $t = 1 : T$  **do**

    Sample  $a \sim \text{Bernoulli}(p)$  (if the patient arrives in that time slot  $a = 1$  or not  $a = 0$ )

    Sample  $z \sim Pr(z|\vec{s}, x, a)$ , If  $a = 0$  (patient does not arrive), then  $z = \emptyset$

$$\bar{B}[j, t] = \Theta(\bar{B}[j, t - 1], z, t) \text{ (See equation (6.4))}$$

    Sample  $\tau \sim \text{Multi}[1, \dots, T - t - 1]$

    Sample  $y, \tilde{\tau} \sim Pr(y, \tilde{\tau}|\vec{s}, t, \tau, x)$  (See equation (6.6) and equation (6.7))

$$\hat{B} = \Phi(\bar{B}[j, t], y, \tilde{\tau}, \tau, t) \text{ (See equation (6.5))}$$

$$\bar{B}[j + K + r(t + \tilde{\tau}), t + \tilde{\tau}] = \hat{B}$$

$$r(t + \tilde{\tau}) = r(t + \tilde{\tau}) + 1$$

$$j = j + 1$$

---

Copy belief vectors at time  $t$  to the belief at  $t + 1$ .

---

---

**Algorithm 2** Approximate Policy Computation Part

---

Function: TPBVI

**Input:** Sets  $\Gamma(q), \forall q > t$ ,

**for**  $\tau \in \{1, \dots, T - t - 1\}$  **do**

**for**  $z \in \mathcal{Z}$  **do**

**for**  $(\tilde{\tau}, y) \in \{t + 1, \dots, t + \tau\} \times \mathcal{Y}$  **do**

**for**  $\alpha \in \Gamma(t + \tilde{\tau})$  **do**

**for**  $\vec{s}, l \in \tilde{\mathcal{S}} \times \{0, 1\}$  **do**

$$\theta[\vec{s}, l] = \sum_{\tilde{l}} \alpha[\vec{s}, l] Pr(\tilde{l}|l, z) Pr(y, \tilde{\tau}|\vec{s}, \tau, x)$$

$$\theta[\vec{s}, l] = \theta[\vec{s}, l] Pr(z|\vec{s}, x)$$

$$\Gamma(t, z, \tau, \tilde{\tau}, y) = \Gamma(t, z, \tau, \tilde{\tau}, y) \cup \{\theta\}$$

**for**  $b \in \bar{B}[, t]$  **do**

**for**  $z \in \mathcal{Z}$  **do**

**for**  $\tau \in \{1, \dots, T - t - 1\}$  **do**

$$\zeta = \sum_y \sum_{\tilde{\tau}} \arg \max_{\alpha \in \Gamma(t, z, \tau, \tilde{\tau}, y)} [\alpha]' b$$

$$\Gamma(t, z, \tau) = \Gamma(t, z, \tau) \cup \zeta$$

**for** belief point  $b \in \bar{B}[, t]$  **do**

**for**  $z \in \mathcal{Z}$  **do**

$$\{\alpha', (\tau)'\} = \arg \max_{\tau, \alpha \in \Gamma(t, z, \tau)} \sum_{\vec{s}, l} \left[ -\tilde{C}(\vec{s}, t, z, l) + \alpha[\vec{s}, l] \right] b(\vec{s}, l)$$

$$\Gamma(t, z) = \Gamma(t, z) \cup \alpha'$$

$$A(t, z) = A(t, z) \cup (\tau)'$$

$$\Gamma(t) = \Gamma(t) + \Gamma(t, z)$$

**Output:**  $\Gamma(t), \{\Gamma(t, z), \forall z\}, \{A(t, z), \forall z\}$

$\Gamma(t, z)$  is set of alpha vectors and each one of them is optimal for one of the beliefs in the

set  $\bar{B}(, t)$ ,  $A(t, z)$  is the set of optimal actions corresponding to the alpha vectors in  $\Gamma(t, z)$

For any belief  $b, z$  find the nearest point in  $\bar{B}[, t]$  and use the corresponding alpha vector

in  $\Gamma(t, z)$  and the corresponding action  $A(t, z)$

---

---

**Algorithm 3** Approximate Policy Computation Part Continued

---

$\Gamma(T + 1) = \{\mathbf{0}\}$   
**for**  $t = 0$  to  $T - 1$  **do**  
 $\Gamma(T - t) = \text{TPBVI}(\{\Gamma(T - r)\}_{-1 \leq r \leq t-1})$

---



---

**Algorithm 4** Exact Policy Computation Part

---

NOTATION:  $\oplus$  is the Cartesian sum

Function EVI

**Input:** Sets  $\Gamma(q), \forall q > t,, \Gamma(T + 1) = \{\mathbf{0}\}$ ,

**for**  $z \in \mathcal{Z}$  **do**

**for**  $\tau \in \{1, \dots, T - t - 1\}$  **do**

**for**  $(\tilde{\tau}, y) \in \{t + 1, \dots, t + \tau\} \times \mathcal{Y}$  **do**

**for**  $\alpha \in \Gamma(t + \tilde{\tau})$  **do**

**for** each  $\vec{s}, l$  **do**

$$\boldsymbol{\theta}[\vec{s}, l] = \frac{-1}{|\mathcal{Y}| |\tilde{\tau}|} \tilde{C}(\vec{s}, t, z, l) + \delta \sum_{\tilde{l}} \boldsymbol{\alpha}[\vec{s}, l] Pr(\tilde{l}|l, z, x) Pr(y, \tilde{\tau}|\vec{s}, \tau, x)$$

$$\boldsymbol{\theta}[\vec{s}, l] = \boldsymbol{\theta}[\vec{s}, l] Pr(z|\vec{s}, x)$$

$$\Gamma(t, z, \tau, \tilde{\tau}, y) = \Gamma(t, z, \tau, \tilde{\tau}, y) \cup \{\boldsymbol{\theta}\}$$

$$\Gamma(t, z, \tau) = \text{prune}(\Gamma(t, z, \tau) \oplus \Gamma(t, z, \tau, \tilde{\tau}, y))$$

$$\Gamma(t, z) = \text{prune}(\Gamma(t, z) \oplus \Gamma(t, z, \tau))$$

$$\Gamma(t) = \text{prune}(\Gamma(t) \oplus \Gamma(t, z))$$

**Output:**  $\Gamma(t), \{\Gamma(t, z), \forall z \in \mathcal{Z}\}$ ,

For optimal action at belief  $b$  at time  $t$  following observation  $z$  choose the optimal alpha vector from  $\Gamma(t, z)$  and choose the action corresponding to the alpha vector selected

$\Gamma(T + 1) = \{\mathbf{0}\}$

**for**  $t = 0$  to  $T - 1$  **do**

$\Gamma(T - t) = \text{EVI}(\{\Gamma(T - r)\}_{-1 \leq r \leq t-1})$

---

In Algorithm 4, the prune function is taken from [Kri17].

## 6.8.2 Appendix B

### Graphical model.

We define a random variable  $V(t)$ , where  $V(t) = 1$  indicates that the patient visits the clinician in time slot  $t$  and is zero otherwise. Write the realization of the visit trajectory as  $\vec{v} = [v(1), \dots, v(T)]$ . Let the screening policy be  $\pi$ . Next, we define the joint distribution of all the random variables that appear in the model.

The joint distribution of the state trajectory  $\vec{s}$ , the external information process trajectory  $\vec{y}$ , the visit trajectory  $\vec{v}$ , the test outcome trajectory  $\vec{z}$  is given as

$$\begin{aligned} & Pr(\vec{s}, \vec{y}, \vec{v}, \vec{z} | x) = \\ & Pr(\vec{s} | x) Pr(\vec{y} | \vec{s}, x) Pr(\vec{v}, \vec{z} | \vec{s}, \vec{y}, x) \\ & Pr(\vec{s} | x) \Pi_t Pr(y(t) | \vec{s}(t), x) \Pi_t Pr(v(t) | \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi) Pr(z(t) | v(t), \vec{s}) \end{aligned} \quad (6.8)$$

We compute  $Pr(\vec{y} | \vec{s})$  as  $Pr(\vec{y} | \vec{s}) = \bar{\Pi}_t Pr(y(t) | \vec{s}, x)$ , where  $\bar{\Pi}$  is the product operator and  $Pr(y(t) | \vec{s}, x)$  is the probability of  $Y(t) = y(t)$  conditional on the entire state trajectory. We assumed that the observations  $y(t)$  conditional on state trajectory through  $\vec{s}(t)$  is independent of other random variables in the model. Therefore,  $Pr(y(t) | \vec{s}, x) = Pr(y(t) | \vec{s}(t), x)$ .

We compute  $Pr(\vec{v}, \vec{z} | \vec{s}, \vec{y}, x)$  as

$$Pr(\vec{v}, \vec{z} | \vec{s}, \vec{y}, x) = \bar{\Pi}_t Pr(v(t) | \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi) Pr(z(t) | v(t), \vec{s})$$

where  $Pr(v(t) | \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi)$  is the probability of visit in time  $t$  conditional on visit indicator in time  $t-1$ , the test outcomes through time  $t-1$ , the entire external information process trajectory, the state trajectory  $\vec{s}$  and the policy  $\pi$  and  $Pr(z(t) | v(t), \vec{s}(t))$  is the probability of test outcome conditional on visit and the state trajectory. Note that  $z(t)$ 's value when there is a visit depends only on the state trajectory through time  $t$ . If there is no visit, then  $z(t) = \emptyset$ . It is easy to simplify  $Pr(v(t) | \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi)$ . Based on all the observations until time  $t-1$  the policy  $\pi$  would have recommended a next screening time.

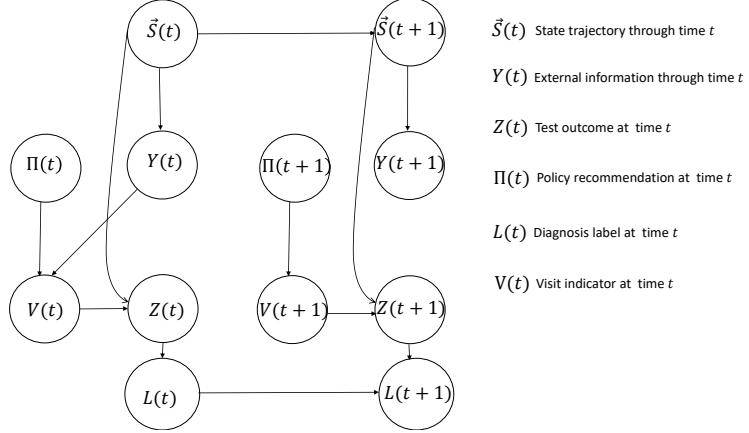


Figure 6.3: Graphical model for the screening setting.

If the next screening time is not  $t$ , then  $\Pr(v(t) = 1 \mid \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi) = \Pr(Y(t) > \tilde{y} \mid \vec{s}, x)$ . If the screening time is  $t$ , then  $\Pr(v(t) = 1 \mid \vec{v}(t-1), \vec{z}(t-1), \vec{y}, \vec{s}, \pi) = 1$

### 6.8.3 Appendix C

**Sufficient statistic for the history.** In this section, our aim is to show that instead of considering the entire history the clinician can only use the belief that we constructed in Appendix A.

The history through time  $t$  when the patient arrives can be written as

$$h(t) = \left[ z(t), y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1}, \tilde{\tau}, \tau(t - \tilde{\tau}), h(t - \tilde{\tau}) \right]$$

where  $z(t)$  is the test outcome at time  $t$ ,  $y(t)$  is the external observation at time  $t$ , and  $\tau(t - \tilde{\tau})$  was the prescribed arrival time in the last arrival which occurred at time  $t - \tilde{\tau}$ .

Write the probability that the patient's state trajectory is  $\vec{s}$ , the diagnosis state  $l$  (the diagnosis state corresponds to the state after the observation  $z(t)$  in time slot  $t$ ), conditioned on the history  $h(t)$  as  $\Pr(\vec{s}, l \mid h(t))$ . Next, we describe how to compute  $\Pr(\vec{s}, l \mid h(t))$  in terms of the probability distribution  $\Pr(\vec{s}, \tilde{l} \mid h(t - \tilde{\tau}))$ .

$$\begin{aligned}
& \Pr(\vec{s}, l, v | h(t)) \\
&= \frac{\Pr(\vec{s}, l, z(t), y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | \tau(t-\tilde{\tau}), h(t-\tilde{\tau}))}{\Pr(z(t), y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | \tau(t-\tilde{\tau}), h(t-\tilde{\tau}))} \\
&= \frac{\sum_{\tilde{l}} \Pr(\vec{s}, \tilde{l}, l, z(t), y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | \tau(t-\tilde{\tau}), h(t-\tilde{\tau}))}{\Pr(z(t), y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | h(t-\tilde{\tau}))} \\
&\frac{\sum_{\tilde{l}} \Pr(\vec{s}, \tilde{l} | h(t-\tilde{\tau})) \Pr(y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | \vec{s}, \tau(t-\tilde{\tau})) \Pr(z(t) | \vec{s}) \Pr(l | \tilde{l}, z(t))}{\Pr(z(t), y(t) | h(t-\tilde{\tau}))} \tag{6.9}
\end{aligned}$$

In the above equation,  $\Pr(l | \tilde{l}, z(t))$  is the probability of the new diagnosis state conditional on the existing diagnosis state. If the existing diagnosis state is 1, then the new diagnosis state has to be 1. If the existing diagnosis state is 0, then the new diagnosis state is 1 if  $z \in \mathcal{Z}^+$  and 0 otherwise.

By definition the belief at time  $t$  is  $\Pr(\vec{s}, l | h(t))$ , which we write as  $\hat{\mathbf{b}}$  and we write  $\Pr(\vec{s}, l | h(t-\tilde{\tau}))$  as  $\mathbf{b}$ .

$$\hat{b}(\vec{s}, l) = \frac{\sum_{\tilde{l}} b(\vec{s}, \tilde{l}) \Pr(y(t), \{y(r) \leq \tilde{y}\}_{r=t-\tilde{\tau}}^{t-1} | \vec{s}, \tau(t-\tilde{\tau})) \Pr(z(t) | \vec{s}) \Pr(l | \tilde{l}, z(t))}{\Pr(z(t), y(t) | h(t-\tilde{\tau}))} \tag{6.10}$$

From the above equation, we can conclude that keeping a track of beliefs is sufficient as the previous belief can be used to compute the new belief (combined with the distributions over the observations).

#### 6.8.4 Appendix D

**Proof of Lemma 3.** We re-write the value function defined for time slot  $t$ , which is also a decision epoch (the patient arrives in this slot and test is done) in equation (6.3) in the

main text below.

$$V(\mathbf{b}, t) = \max_{\boldsymbol{\tau}} \left[ - \sum_{\vec{s}, l, z} b(\vec{s}, l) Pr(z | \vec{s}, x) \left[ \tilde{C}(\vec{s}, t, z, l) \right] + \sum_{z, \tilde{\tau}, y} Pr(z, y, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, x) V(\hat{\mathbf{b}}, t + \tilde{\tau}) \right] \quad (6.11)$$

where  $\hat{b}(\vec{s}, l) = Pr(\vec{s}, l | \mathbf{b}, \boldsymbol{\tau}, y, z, \tilde{\tau}, x) = \frac{Pr(\vec{s}, l, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})}{Pr(y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})}$ . Note that  $l$  in the above equation is the diagnosis state before the test outcome  $z$  is observed.

$Pr(\vec{s}, l, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})$  is the probability that the patient's trajectory is  $\vec{s}$ , the test outcome in time slot  $t$  is  $z$ , the external information on patient's next arrival, which occurs  $\tilde{\tau}$  time slots later is  $y$  conditioned on the recommendation plan  $\boldsymbol{\tau}$ . We simplify  $Pr(\vec{s}, l, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, t)$  as

$$Pr(\vec{s}, l, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, t) = \sum_{\tilde{l}} Pr(\vec{s}, \tilde{l}, l, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}, t) = \sum_{\tilde{l}} b(\vec{s}, \tilde{l}) Pr(l, y, z, \tilde{\tau} | \vec{s}, \tilde{l}, \boldsymbol{\tau}, t)$$

We simplify  $Pr(l, y, z, \tilde{\tau} | \vec{s}, \tilde{l}, \boldsymbol{\tau}, t)$  as

$$\begin{aligned} Pr(l, y, z, \tilde{\tau} | \vec{s}, \tilde{l}, \boldsymbol{\tau}, t) &= Pr(z | \vec{s}, \tilde{l}, \boldsymbol{\tau}, t) Pr(l | \tilde{l}, z, \boldsymbol{\tau}, t, \vec{s}) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, l, \tilde{l}, t) \\ &= Pr(z | \vec{s}) Pr(l | \tilde{l}, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t) \end{aligned} \quad (6.12)$$

where  $Pr(l | \tilde{l}, z)$  is the transition probability from current diagnosis label  $\tilde{l}$  to the new label  $l$  following the observation  $z$ . If the patient is diagnosed to be unhealthy, then the diagnosis label continues to be one. If the patient is not diagnosed, then the label turns to one from zero as soon as the patient is diagnosed. Formally stated,  $Pr(l = 0 | \tilde{l} = 0, z) = 0, \forall z \in \mathcal{Z}^+$ ,  $Pr(l = 0 | \tilde{l} = 0, z) = 1, \forall z \in [\mathcal{Z}^+]^c$ ,  $Pr(l = 0 | \tilde{l} = 1, z) = 0, \forall z \in \mathcal{Z}$ . In the above equation (6.12), we wrote  $Pr(z | \vec{s}, \tilde{l}, \boldsymbol{\tau}, t) = Pr(z | \vec{s})$ ; this is true because the test outcome is independent of whether the patient has been diagnosed or not, the recommendation plan and the time. We also state  $Pr(l | \tilde{l}, z, \boldsymbol{\tau}, t, \vec{s}) = Pr(l | \tilde{l}, z)$  where we use the condition that  $l$  is independent of  $\boldsymbol{\tau}, t, \vec{s}$  conditional on  $\tilde{l}, z$  (this follows from the definition of  $l$ ). Also, if the state trajectory, the recommended time of next arrival, and the current time are known, then the distribution of external information at next arrival time and next arrival time is completely specified by  $Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t)$  and whether the patient has been diagnosed or not

does not enter the external information process  $Pr(y, \tilde{\tau}|z, \vec{s}, \boldsymbol{\tau}, l, \tilde{l}, t) = Pr(y, \tilde{\tau}|z, \vec{s}, \boldsymbol{\tau}, t)$  (this follows from the definition of external information process).

For all  $\tilde{\tau} \leq \boldsymbol{\tau}(z)$  we have

$$Pr(y, \tilde{\tau}|z, \vec{s}, \boldsymbol{\tau}, t) = Pr\left(\left\{Y(s) \leq \tilde{y}; t < \forall s \leq t + \tilde{\tau}\right\}, Y(t + \tilde{\tau}) = y \middle| \vec{s}\right) \quad (6.13)$$

For  $\tilde{\tau} > \boldsymbol{\tau}(z)$

$$Pr(y, \tilde{\tau}|z, \vec{s}, \boldsymbol{\tau}, t) = 0 \quad (6.14)$$

Thus we can write the updated belief as

$$\begin{aligned} \hat{b}(\vec{s}, \tilde{l}) &= \frac{Pr(\vec{s}, \tilde{l}, y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})}{Pr(y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})} = \\ &\frac{\sum_l b(\vec{s}, l) Pr(\tilde{l}, y, z, \tilde{\tau} | \vec{s}, l, \boldsymbol{\tau}, t)}{Pr(y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})} = \frac{\sum_l b(\vec{s}, l) Pr(z | \vec{s}) Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t)}{Pr(y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})} \end{aligned} \quad (6.15)$$

In this proof up until now we have computed the expression for  $\hat{\mathbf{b}}$ .

We will use the principle of induction to prove the above result. The claim in the Lemma holds for the value function in time slot  $T + 1$  as it is defined to be identically zero. Next, we assume that the condition in the Lemma holds for all  $r > t$ . Therefore, we can write

$$\begin{aligned} V(\hat{\mathbf{b}}, t + \tilde{\tau}) &= \max_{\boldsymbol{\alpha} \in \Gamma(t + \tilde{\tau})} \sum_{\vec{s}, \tilde{l}} \boldsymbol{\alpha}[\vec{s}, \tilde{l}] \hat{b}(\vec{s}, \tilde{l}) \\ &= \max_{\boldsymbol{\alpha} \in \Gamma(t + \tilde{\tau})} \sum_{\vec{s}, \tilde{l}, l} \boldsymbol{\alpha}[\vec{s}, \tilde{l}] \frac{b(\vec{s}, l) Pr(z | \vec{s}) Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau})}{Pr(y, z, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau})} \end{aligned} \quad (6.16)$$

Suppose that each  $\boldsymbol{\alpha} \in \Gamma(t + \tilde{\tau})$  is indexed. Henceforth, we also write the index of  $\boldsymbol{\alpha}$  in superscript as well, i.e.  $\boldsymbol{\alpha}^k$ . Define a function  $\tilde{k} : \Delta \times \mathcal{T} \times \mathcal{T} \times \mathcal{Y} \times \mathcal{T} \rightarrow \mathbb{Z}$  as follows.

$$\tilde{k}(\mathbf{b}, \boldsymbol{\tau}, \tilde{\tau}, y, t, z) = \arg \max_k \sum_{\vec{s}, l, \tilde{l}} \boldsymbol{\alpha}^k[\vec{s}, \tilde{l}] b(\vec{s}, l) Pr(z | \vec{s}) Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \quad (6.17)$$

Substituting (6.16) and (6.17) into (6.11) we obtain

$$\begin{aligned}
V(\mathbf{b}, t) = & \\
& \max_{\boldsymbol{\tau}} \left[ \sum_{\vec{s}, l, z} b(\vec{s}, l) Pr(z | \vec{s}) [\tilde{C}(\vec{s}, t, z, l)] + \right. \\
& \left. \sum_{z, \tilde{\tau}, y} \sum_{\vec{s}, l, \tilde{l}} \boldsymbol{\alpha}^{\tilde{k}}[\vec{s}, l] b(\vec{s}, l) Pr(z | \vec{s}) Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \right] \\
& \max_{\boldsymbol{\tau}} \left[ \sum_{\vec{s}, l, z} b(\vec{s}, l) Pr(z | \vec{s}) \left[ \tilde{C}(\vec{s}, t, z, l) + \sum_{\tilde{\tau}, y, \tilde{l}} \boldsymbol{\alpha}^{\tilde{k}}[\vec{s}, \tilde{l}] Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \right] \right] \\
& \max_{\boldsymbol{\tau}} \left[ \sum_{\vec{s}, l, z, \tilde{\tau}, y, \tilde{l}} b(\vec{s}, l) Pr(z | \vec{s}) \left[ \tilde{C}(\vec{s}, t, z, l) \frac{1}{\omega} + \boldsymbol{\alpha}^{\tilde{k}}[\vec{s}, l] Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \right] \right]
\end{aligned} \tag{6.18}$$

where  $\omega$  is the total possible combinations of  $\tilde{\tau}$ ,  $y$  and  $\tilde{l}$ . In the above (6.18), we only use  $\tilde{k}$  instead of the entire function  $\tilde{k}(\mathbf{b}, \boldsymbol{\tau}, \tilde{\tau}, y, t, z)$  for clearer notation.

Observe that the function  $\tilde{k}(\mathbf{b}, \boldsymbol{\tau}, \tilde{\tau}, y, t, z)$  can take finitely many values. Therefore, for a fixed combination of values  $z, \tilde{\tau}, y$  the space  $\mathcal{B}$  is thus partitioned into regions where  $\tilde{k}(\mathbf{b}, \boldsymbol{\tau}, \tilde{\tau}, y, t, z)$  takes a fixed value. Hence, the term

$$\left[ \tilde{C}(\vec{s}, t, z, l) \frac{1}{\omega} + \boldsymbol{\alpha}^{\tilde{k}}[\vec{s}, l] Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \right]$$

takes a fixed value in each partition as well and this is true of  $\forall \vec{s}, l$ . Finally, we can create a common refinement of the partitions such that the  $\sum_{z, \tilde{\tau}, y} Pr(z | \vec{s}) \left[ \tilde{C}(\vec{s}, t, z, l) \frac{1}{\omega} + \boldsymbol{\alpha}^{\tilde{k}}[\vec{s}, l] Pr(\tilde{l} | l, z) Pr(y, \tilde{\tau} | z, \vec{s}, \boldsymbol{\tau}, t + \tilde{\tau}) \right]$  is fixed for each partition. Therefore, we have so far shown that the term inside (6.18) is piecewise linear. The first term inside (6.18) is convex (since it is linear). The second term inside (6.18) is convex because of the definition of (6.17). Thus, the term inside the max operator (6.18) is piecewise linear and convex. The maximum of piecewise linear and convex functions is also piecewise linear and convex. This proves the result. ■

### 6.8.5 Appendix E

**Proof of Proposition 3.** We first re-write the expression for the cost incurred in a time slot below

$$\tilde{C}(\vec{s}, t, z, l) = \begin{cases} wC(t - t_D; t_D) + (1 - w)\delta^t I(z \neq \emptyset) & t \leq T, l = 0, z \in \mathcal{Z}^+ \\ wC(T - t_D; t_D) & t = T, l = 0 \\ (1 - w)\delta^t I(z \neq \emptyset) & \text{otherwise} \end{cases} \quad (6.19)$$

We substitute  $C(t - t_D; t_D) = c(t - t_D)\delta^{t_D}$  to obtain

$$\tilde{C}(\vec{s}, t, z, l) = \begin{cases} wc(t - t_D)\delta^{t_D} + (1 - w)\delta^t I(z \neq \emptyset) & t \leq T, l = 0, z \in \mathcal{Z}^+ \\ wc(t - t_D)\delta^{t_D} & t = T, l = 0 \\ (1 - w)\delta^t I(z \neq \emptyset) & \text{otherwise} \end{cases} \quad (6.20)$$

Define another function  $\hat{C}$  as follows.

$$\hat{C}(\vec{s}, t, z, l) = \begin{cases} wc(t - t_D)\delta^t + (1 - w)\delta^t I(z \neq \emptyset) & t \leq T, l = 0, z \in \mathcal{Z}^+ \\ wc(t - t_D)\delta^t & t = T, l = 0 \\ (1 - w)\delta^t I(z \neq \emptyset) & \text{otherwise} \end{cases} \quad (6.21)$$

Next, we derive an upper bound on time to detection  $t_d$  in terms of the time of incidence  $t_D$ . Disease starts at  $t_D$  and the next screening has to occur at time at most  $t_D + W$ . Since there are no false positives and false negatives the patient is detected in the next screening. Therefore, we have  $t_D \leq t_d \leq t_D + W$ .

We derive an upper bound on the difference between  $\hat{C}$  and  $\tilde{C}$  as

$$\begin{aligned} \tilde{C}(\vec{s}, t, z, l) - \hat{C}(\vec{s}, t, z, l) &\leq (\delta^{T_D} - \delta^{t_d})(c(t_d - t_D)) \\ &\leq \delta^{T_D}(1 - \delta^W)(c(W)) \\ &\leq (1 - \delta^W)(c(W)) \end{aligned} \quad (6.22)$$

We require that  $\tilde{C}(\vec{s}, t, z, l) - \hat{C}(\vec{s}, t, z, l) \leq \kappa$ . It is sufficient to bound  $(1 - \delta^W)(c(W)) \leq \kappa \implies \delta \geq (1 - \frac{\kappa}{c(W)})^{1/W}$ . Henceforth, we assume that  $\delta \geq \delta^* = (1 - \frac{\kappa}{c(W)})^{1/W}$ . Therefore, we have

$$\hat{C}(\vec{s}, t, z, l) \leq \tilde{C}(\vec{s}, t, z, l) \leq \hat{C}(\vec{s}, t, z, l) + \kappa \quad (6.23)$$

$\forall \vec{s}, t, z, l$ . It can be shown that the solutions to (6.3) with  $\hat{C}$  instead of  $\tilde{C}$  only differ by  $\kappa$  (at most). Let the optimal policy and the corresponding optimal value when cost is  $\tilde{C}$  be given as  $\pi_1$  and  $C_1(\pi_1)$  ( $C_2(\pi_2)$ ). From (6.23) we have

$$\begin{aligned} C_2(\pi_1) &\leq C_1(\pi_1) \leq C_2(\pi_1) + \kappa \\ C_2(\pi_2) &\leq C_1(\pi_2) \leq C_2(\pi_2) + \kappa \end{aligned} \quad (6.24)$$

From the definition of  $C_1$  and  $C_2$  the following can be derived

$$C_2(\pi_2) \leq C_1(\pi_1) \leq C_2(\pi_1) + \kappa \leq C_2(\pi_2) + \kappa \quad (6.25)$$

Next, we will use  $\hat{C}$  instead of  $\tilde{C}$ . Define a function  $\bar{C}(\vec{s}, t, z, s) = \frac{\hat{C}(\vec{s}, t, z, l)}{\delta^t}$ .

We write the value function for the modified objective as

$$\bar{V}(\mathbf{b}, t) = \max_{\boldsymbol{\tau}} \left[ \sum_{\vec{s}, l, z} b(\vec{s}, l) Pr(z|\vec{s}) [\bar{C}(\vec{s}, t, z, l)] + \delta \sum_{z, \tilde{\tau}, y} Pr(z, y, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}) \bar{V}(\hat{\mathbf{b}}, t + \tilde{\tau}) \right] \quad (6.26)$$

If  $T$  is sufficiently large, then the difference between the value function of the finite horizon and the infinite horizon version of the problem can be made as small as desired.

The maximum difference between the value function computed upto the infinite horizon versus one that is truncated at time  $T$  is  $\delta^T(1/(1 - \delta) + c(W))$ . Suppose we want to bound the difference by  $\eta$ . If  $\delta^T(1/(1 - \delta) + c(W)) \leq \eta \implies \delta^T \leq \eta(1 - \delta) + c(W)\eta$ . If  $T \geq \max\{\frac{\log(\frac{\eta}{2+\eta})}{\log(\delta)}, \frac{\log(\frac{\eta}{2c(W)})}{\log(\delta)}\}$ , then the difference is bounded by  $\eta$ . Let us consider the infinite horizon for  $\bar{V}$  above. We will construct the proof for the infinite horizon version of the problem and then use the above observation to extend the proof to finite horizon.

From equation (6.26), we can define an operator given as  $\Phi_t$  defined as follows.

$$\Phi_t(V) = \max_{\tau} \left[ \sum_{\vec{s}, l, z} b(\vec{s}, l) Pr(z|\vec{s}) \left[ \bar{C}(\vec{s}, t, z, l) \right] + \delta \sum_{z, \tilde{\tau}, y} Pr(z, y, \tilde{\tau} | \mathbf{b}, \boldsymbol{\tau}) V(\gamma(\mathbf{b}, y, z, \boldsymbol{\tau})) \right] \quad (6.27)$$

where  $\gamma$  is the belief update operator that can be defined based on the equation (6.15) in the proof of Lemma 3. Based on standard arguments used to show that a Bellman operator is a contraction mapping [21], we can show that the above operator is a contraction mapping as well with a contraction factor  $\delta$ .

Similarly, we define an operator  $\tilde{\Phi}_t$  associated with our algorithm. Our algorithm takes alpha vectors as input and generates a new set of alpha vectors. Since the set of alpha vectors define the value function (see Lemma 3), we can view the proposed procedure to be an operator that maps a value function to another value function. Define the error introduced by one iteration of the approximate backup  $\tilde{\Phi}_t V^B(:, t)$  as  $\epsilon = \max_{\mathbf{b} \in \Delta} |\tilde{\Phi}_t V^{\bar{B}}(\mathbf{b}) - \Phi_t V^{\bar{B}}(\mathbf{b})|$ . Note that the backup at time  $t$  will use  $\bar{B}[; t]$  as the input vector of beliefs. Define the density  $\delta_{\bar{B}[t]}$  of a set of points  $\bar{B}[t]$  to be the maximum distance from any belief in the simplex  $\Delta$  to a belief in the set  $\bar{B}[t]$ .

$$\delta_{\bar{B}[t]} = \max_{\mathbf{b}' \in \Delta} \min_{\mathbf{b} \in \bar{B}[t]} \|\mathbf{b} - \mathbf{b}'\|_1 \quad (6.28)$$

We now compute the maximum value  $\epsilon$ . Let  $\mathbf{b}' \in \Delta$  be the point where proposed procedure makes the largest error and let  $\mathbf{b} \in \bar{B}[t]$  be the closest 1-norm sampled belief to  $\mathbf{b}'$ . Let  $\boldsymbol{\alpha}$  be the vector maximal at  $\mathbf{b}$  (this vector is generated by the backup at  $\mathbf{b}$  because we assume that the value function in the future time slot computed  $\mathbf{b}$  is known thus there is no error at  $\mathbf{b}$ ) and let  $\boldsymbol{\alpha}'$  be the vector maximal at  $\mathbf{b}'$ . Therefore,

$$\begin{aligned} \epsilon &\leq [\boldsymbol{\alpha}']^* \mathbf{b}' - [\boldsymbol{\alpha}]^* \mathbf{b}' \\ &= [\boldsymbol{\alpha}']^* \mathbf{b}' - [\boldsymbol{\alpha}]^* \mathbf{b}' + [\boldsymbol{\alpha}]^* \mathbf{b} - [\boldsymbol{\alpha}]^* \mathbf{b} \\ &\leq [\boldsymbol{\alpha}']^* \mathbf{b}' - [\boldsymbol{\alpha}]^* \mathbf{b}' + [\boldsymbol{\alpha}]^* \mathbf{b} - [\boldsymbol{\alpha}]^* \mathbf{b} \\ &= [(\boldsymbol{\alpha}' - \boldsymbol{\alpha})]^* (\mathbf{b} - \mathbf{b}') \\ &\leq \|[(\boldsymbol{\alpha}' - \boldsymbol{\alpha})]\|_\infty \|(\mathbf{b} - \mathbf{b}')\|_1 \end{aligned}$$

In the last equation above, we use Holder's inequality. Note that  $\|[(\boldsymbol{\alpha}' - \boldsymbol{\alpha})]\|_\infty$  represents

the maximum difference in the costs that are achieved starting from a certain state and is given as  $\zeta$ . Note that  $\zeta < \infty$  because the total number of time slots is finite and the costs in each decision epoch are bounded. Thus we can write the above inequality as

$$\epsilon \leq \zeta \delta_{\bar{B}[t]} \quad (6.29)$$

We now proceed to the overall error introduced by the Algorithm.

$$\begin{aligned} \epsilon_t &= \|V^{\bar{B}[t]}(\cdot, t) - V(\cdot, t)\|_\infty \\ &= \|\tilde{\Phi}_t V^{\bar{B}[t+1]}(\cdot, t+1) - \Phi V(\cdot, t)\|_\infty \\ &= \|\tilde{\Phi}_t V^{\bar{B}[t+1]}(\cdot, t+1) - \Phi_t V^{\bar{B}[t+1]}(\cdot, t+1) + \Phi_t V^{\bar{B}[t+1]}(\cdot, t+1) - \Phi V(\cdot, t)\|_\infty \\ &\leq \|\tilde{\Phi}_t V^{\bar{B}[t+1]}(\cdot, t+1) - \Phi_t V^{\bar{B}[t+1]}(\cdot, t+1)\|_\infty + \|\Phi_t V^{\bar{B}[t+1]}(\cdot, t+1) - \Phi V(\cdot, t)\|_\infty \\ &\leq \zeta \delta_{\bar{B}[t]} + \delta \epsilon_{t+1} \\ &= \zeta \delta_{\bar{B}[t]} + \delta \zeta \delta_{\bar{B}[t+1]} + \delta^2 \epsilon_{t+2} \\ &= \zeta \Omega(\bar{B}) \frac{1}{1-\delta} \end{aligned}$$

The above result can be extended to the finite horizon case. If  $T$  is sufficiently large, then the value function achieved by the proposed policy and the exact policy will be close to  $V^{\bar{B}[t]}(\cdot, t)$  and  $V(\cdot, t)$  respectively. If  $\delta_{\bar{B}}$  is sufficiently small, then the proposed and the exact optimal policy will achieve very similar value function.

If the approximation error goes to zero, then the value function of the proposed and the exact optimal policy are the same. Suppose that there is a unique optimal solution to (6.3), we can see that the proposed and the exact optimal policies will also be identical. ■

### 6.8.6 Appendix F

**Proof for Theorem 16.** In Lemma 3 we showed that  $V(\mathbf{b}, t)$  can be written as  $\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^* \mathbf{b}$ . Since we are considering the space of models in this Theorem, we will define the value function on the space  $\mathbf{L}$ . Following Lemma 3 we can write

$$V((\mathbf{b}, \mathbf{m}), t) = \max_k \boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b} \quad (6.30)$$

where  $(\mathbf{b}, \mathbf{m}) \in \mathbf{L}$  and  $\boldsymbol{\alpha}(k, \mathbf{m})$  is the  $k^{th}$  alpha vector for model  $\mathbf{m}$ . We can assume that the same indexing is used for the alpha vectors across all the models. (Also, based on the definition of alpha vectors the total number of alpha vectors is the same across all the models.)

Consider a fixed belief  $\mathbf{b}$  and a fixed model  $\mathbf{m}$ .  $a^*((\mathbf{b}, \mathbf{m}), t)$  is the unique maximizer (except for a set of measure zero of models  $\mathbf{m}$ ; this is based on the assumption). Let  $k^*(\mathbf{m}, \mathbf{b})$  correspond to the index of the corresponding optimal alpha vector. (Here we have assumed that the index  $k^*(\mathbf{m}, \mathbf{b})$  is unique, but this assumption can be relaxed.) Based on the assumption that for a fixed  $\mathbf{m} \mathbf{b}$ ,  $a^*((\mathbf{b}, \mathbf{m}), t)$  is a unique maximizer (except for a set of measure zero of models  $\mathbf{m}$ ), we can conclude that the  $\boldsymbol{\alpha}(k^*(\mathbf{m}, \mathbf{b}), \mathbf{m})^* \mathbf{b}$  is strictly better than other  $[\boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b}, ; \forall k \neq k^*(\mathbf{m}, \mathbf{b})]$ . Note that there may exist  $\mathbf{m}, \mathbf{b}$  for which the maximizer  $k^*(\mathbf{m}, \mathbf{b})$  is not unique. The measure of such a set is zero (as it will amount to finding  $\mathbf{m}, \mathbf{b}$  such that  $\boldsymbol{\alpha}(k^*(\mathbf{m}, \mathbf{b}), \mathbf{m})^* \mathbf{b} = \boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b}$  for some  $k \neq k^*(\mathbf{m}, \mathbf{b})$ ), thus we can exclude these points.

If all the probability distributions defined in the model are continuous in  $\mathbf{m}$ , then  $\boldsymbol{\alpha}(k, \mathbf{m})$  is a continuous vector valued function of  $\mathbf{m}$  for all  $k$  as well. Therefore, the condition  $\boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b}$  has to be strictly better than  $\boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b}$ ,  $\forall k \neq k^*(\mathbf{m}, \mathbf{b})$  in a neighborhood of  $\mathbf{m}$ . In fact, due to the continuity of  $\boldsymbol{\alpha}(k, \mathbf{m})^* \mathbf{b}$  in  $\mathbf{b}$ , this has to hold true for a neighborhood in the joint space  $\mathbf{L}$ . This implies that the optimal action  $a^*$  stays fixed in the neighborhood as well. This proves the result. ■

# CHAPTER 7

## Optimal Piecewise Local-Linear Approximations

### 7.1 Introduction

From this chapter on, we move to the second part of the dissertation, where we focus on developing optimization methods for machine learning applications. In recent years, data-driven decision-making systems are increasingly becoming common in daily practice. These data-driven models that drive the decision making are often “black-boxes” and hard to interpret. Therefore, interpreting the decisions made by such systems is hard. In this chapter, we build a framework to help interpret such black-box systems. Our framework is based on approximating a black-box model with a piecewise local-linear function. In general, computing an exactly optimal approximation of the black-box function is computationally intractable because of the large number of possible ways to partition the data. Our approach is approximate, computationally tractable, and has provable performance guarantees. This chapter is based on my work in [AZS18].

Machine learning algorithms have proved hugely successful for a wide variety of supervised learning problems. However, in some domains, adoption of these algorithms has been hindered because the “black-box” nature of these algorithms makes their predictions difficult for potential users to interpret. This issue is especially important in the medical domain and security applications [CLG15]. The European Union’s Law on Data Regulation [GF16] makes it mandatory for “black-box” models to explain how they arrive at the predictions before implementing them in practice.

The problem of interpretation has received substantial attention in the literature recently [RSG16] [SGK17] [BKB17] [CSW18]. These papers have approached the problem of

interpreting the black-box model by approximating it with local models (e.g., linear models) in a neighborhood of each data point (See the justification in [LL17]).<sup>1</sup> These papers provide instance-wise explanations of the predictions made by the model and they only interpret the local nature of the model. These papers do not provide insights into the global behavior of the model. There are relatively fewer works [BKB17] that interpret the global model behavior.

While linear models are common to use as local models, there are no global explanation frameworks that partition the feature space into “homogeneous” regions and fit a local-linear model to predict the black-box function behavior. In this chapter, we provide a framework to build such piecewise local-linear approximations of the model.

### 7.1.1 Contributions

We are given a set of data points and the black-box function values computed for those data points. Our goal is to construct a partition of the feature space into subsets and to assign a simple local model to each subset. We propose a dynamic programming based approach to find the partition of the dataset and the set of local models. We prove that the output of our method, which we refer to as piecewise local-linear interpreter (PLLI), is approximately optimal in several different cases, i.e., it probably approximately correct (PAC) learns the black-box function. We use several real and synthetic datasets to establish the utility of the proposed approach.

Our framework is very general and has different applications as described below.

- **Region-wise feature importances:** We broadly categorize the works on feature importance scoring into two categories: a) Global feature importance scoring: Methods such as importance scoring based on tree based models [AK08] fall in this category. These methods identify the factors that the model finds important overall when making the predictions across all the data points. b) Instancewise feature importance scoring: Methods such as [CSW18] fall in this category. These methods identify the factors

---

<sup>1</sup>We define piecewise models later.

that the model finds important when making predictions for a certain data instance. Instancewise methods provide a more refined understanding of the model at an instance level while the global methods give an aggregate understanding of the model. Using an instance based approach across all the data points can be impractical thus it is important to find a middle ground between the global and instance based approaches. Our method helps achieve this task. We divide the feature space into regions and fit local-linear models and assign importance scores to different features in the different regions.

- **Global explanations through instancewise explanations:** Many works on instancewise explanations try to provide global explanation to model behavior in terms of the best  $K$  representative data instances and constructing explanations for them. In [RSG16] authors proposed a method to identify such representative data instances. Our framework divides the feature space into different regions and thus it naturally provides a method to identify different examples of data points. We further elaborate on the utility of our approach in comparison to [RSG16] in the Experiments Section.
- At the end, we leverage ideas from our framework to identify an algorithm that achieves optimal clustering for one-dimensional data in polynomial time. To the best of our knowledge, this is the first proof that a method can achieve optimal clustering for one-dimensional data.

## 7.2 Related Works

The related works on model interpretation can be broadly categorized into two categories. The first kind provide local explanations to a model near a given set of points; the second kind provides a more global explanation. We give the most representative works in each category but by no means this list is exhaustive.

### 7.2.1 Local Frameworks

These frameworks [RSG16] [SGK17] [LL17] [KL17] produce a linear approximation of the black-box model in some neighborhood of a given point. The coefficients of the linear approximation represent the importance associated with different features as predicted by the black-box model in that neighborhood.

We provide more specific details for these different frameworks below.

#### 7.2.1.1 LIME

In the LIME framework [RSG16], the linear approximation of the black-box at each input data point is computed using weighted least squares, with weights coming from an exponential kernel. The authors also extend the framework to identify a set of points at which to provide local-linear approximations; this helps with global model explanation. The points that are identified are not guaranteed to be diverse enough (further explained in the Experiments Section), i.e., they may all come from similar regions in the feature space and/or the range space of the black-box function.

#### 7.2.1.2 Relevance propagation

Layerwise relevance propagation (LRP) [BBM15] and DeepLIFT [SGK17] are relevance propagation frameworks commonly used for interpretation of neural networks. In DeepLIFT, the importance of a feature is determined by perturbing the feature of a data point from a reference value and tracking the gradient/change in the outcome/prediction. DeepLIFT is particularly designed to handle the problems that arise in neural networks such as gradient saturation that are not handled by others [SDV16].

#### 7.2.1.3 SHAP

The SHAP framework [LL17] shows that many of the existing frameworks can be understood as variants of a common linear interpretive model. The authors argue that the coefficients

of a good linear interpretive model should satisfy a particular set of axioms and show that the coefficients derived from Shapley regression provide the unique solution to these axioms.

#### 7.2.1.4 Influence functions

In [KL17], the authors used influence functions to understand the impact of training data points on the predictions of the model. For each new test data point, it identifies the training points that most influence the prediction of the model. This method is also used to determine the importance assigned by a model to the different feature dimensions at a given data point/neighborhood.

### 7.2.2 Global Frameworks

#### 7.2.2.1 Tree based approximations

In [BKB17], the authors develop an approach to approximate the entire black-box function. The objective of their work is to find the best tree that approximates the black-box model. The authors show that their approach learns the exact greedy decision tree. In our experiments, we show that the trees constructed in this greedy manner can provide very poor approximations of the true black-box model.

#### 7.2.2.2 Decision set based approximation

Decision sets (sets of if-then rules) based black-box approximations have been used in [LKC17]. (The framework in [LKC17] is meant for classification problems only; the present chapter focuses on regression problems but is easily adapted to classification problems.) The framework in [LKC17] optimizes an objective that balances the ambiguity between the classifier and the decision rule against the interpretability of the decision set. The approach of [LKC17] does not provide fidelity guarantees; our approach does.

### 7.2.3 Relationship of Piecewise Linear and Piecewise Constant Models to Existing Works

Existing local frameworks described in the previous section construct a local model for each data instance. In most cases, these local models are linear [LL17]. Since there is one model per data instance these models qualify as piecewise linear models, where every different model corresponds to a different piece. For a new data instance, we find the nearest data point in the training set and use the corresponding local-linear model to find the interpretation for the new data instance. In general, it is impractical to have a different model to interpret every instance and inspect each of those interpretations. It is more practical to have a model that interprets a group of data points instead of one data point thus keeping the total number of pieces required to interpret the model below a reasonable value. Existing global frameworks [LKC17] [BKB17] use regression trees or decision sets. These models represent piecewise constant models as there is a fixed value assigned by the model to each decision set or a leaf in the tree.

## 7.3 Problem Formulation

### 7.3.1 Toy Example

In this section, we begin by describing a toy example to illustrate the input and the output from the proposed algorithm. In Fig. 7.1, we show the example of a one dimensional black-box function  $f$ . We input the data  $D = \{(x_i, f(x_i))\}_{i=1}^n$  to the proposed algorithm, where  $(x_i, f(x_i)) \in \mathcal{V}$ . Our goal is to partition the space  $\mathcal{V}$  into homogeneous regions – similar in terms of black-box function values and similar in terms of their features. We first partition the function  $f$ 's range into three intervals as shown in the figure (later we explain the methodology used to decide the intervals).

Next we want to partition the inverse mapping of these intervals. Consider the interval  $[a, b]$  and its inverse  $f^{-1}[a, b] = \{[\alpha, \beta]\} \cup \{[\gamma, \infty)\}$ . The inverse map  $f^{-1}[a, b]$  is not a connected set. Hence, it is natural to partition  $f^{-1}[a, b]$  into two separate connected regions

$[\alpha, \beta]$  and  $[\gamma, \infty]$  (a natural approach to partition the data in  $f^{-1}[a, b]$  is to use k-means clustering with  $k=2$ ). For general functions, also  $f^{-1}[a, b]$  will not be a connected set and thus it is natural to use k-means clustering to separate the data in  $f^{-1}[a, b]$ . The choice of  $k$  depends on number of disconnected sets, which is not known apriori. Hence, a natural approach is to select  $k$  using cross-validation as we discuss later. The Table in Fig. 7.1 summarizes the partition in terms of the  $y$  interval and  $x$  interval and the coefficients for local-linear models and the dotted line shows the piecewise local-linear approximation.

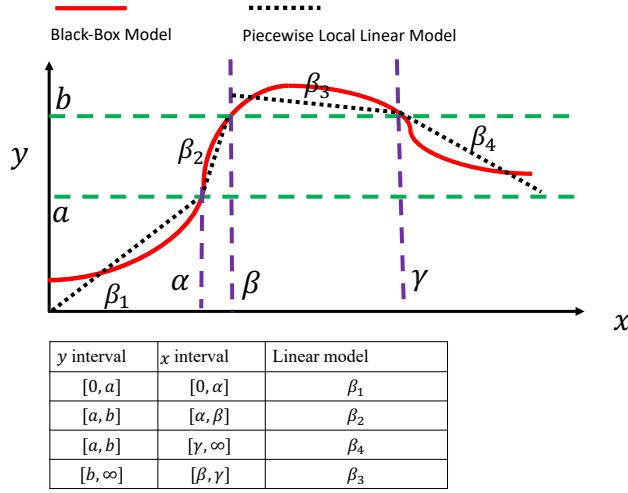


Figure 7.1: Comparison of the black-box model versus the piecewise approximation.

### 7.3.2 Interpretive models

We are given a space  $\mathcal{X}$  of *features* and a space  $\mathcal{Y} = [0, 1]$  of *labels*. We are given a predictive model  $f : \mathcal{X} \rightarrow \mathcal{Y}$  (say a random forest based model or a deep neural network model). The data is distributed according to some *true distribution*  $\mathcal{D}$  (typically unknown) on  $\mathcal{X}$ .<sup>2</sup> Our objective is to interpret  $f$  in terms of *interpretive models*, which are defined below. We seek to find a interpretive model  $g$  that approximates  $f$ .

The interpretive models we consider here represent the most commonly used models in

---

<sup>2</sup>We assume that the cumulative distribution function of  $f$  is a continuous function.

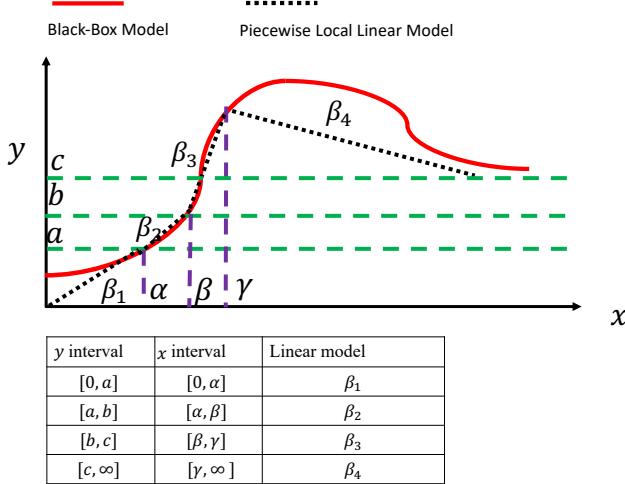


Figure 7.2: Comparison of the black-box model versus the piecewise approximation.

literature on model interpretation [RSG16] [LL17]. The interpretive models we consider are defined by partitioning  $\mathcal{X}$  into a finite number of disjoint sets and assigning a simple model (linear or constant model) to each set of the partition.

To make this precise, recall that a (finite) partition of a subset  $A \subset \mathcal{X}$  is a family  $\mathcal{Z} = \{Z_1, Z_2, \dots, Z_K\}$  of subsets of  $\mathcal{X}$  such that  $\bigcup_{i=1}^K Z_i = A$  and  $Z_i \cap Z_j = \emptyset$  if  $i \neq j$ . Given a partition  $\mathcal{Z}$  of  $A$  and an instance  $a \in A$  we write  $\mathcal{Z}(a)$  for the index of the unique element of the partition  $\mathcal{Z}$  to which  $a$  belongs. Write  $\mathcal{P}(A)$  for the set of all (finite) partitions of  $A$  and  $\mathcal{P}_K(A)$  for the set of partitions having  $K$  elements. We define  $\mathcal{M} = \{M_1, \dots, M_K\}$  a set of local models, where model  $M_j$  corresponds to the local model for points in  $Z_j$ . Each local model  $M_j$  belongs to a set  $\mathcal{H}$  of models, where  $\mathcal{H}$  can be from the family of linear models ( $M_j(x) = b^t x + c$ , where  $b \in \mathbb{R}^{|\mathcal{X}|}$  and  $c \in \mathbb{R}$ ). Given a partition  $\mathcal{Z}$  of  $\mathcal{X}$  and the corresponding set of local models  $\mathcal{M}$ , we define a *interpretive model*  $g_{\mathcal{M}, \mathcal{Z}} : \mathcal{X} \rightarrow Y$  by  $g_{\mathcal{M}, \mathcal{Z}}(x) = M_{\mathcal{Z}(x)}(x)$ .

We define the search space of our partitions next and note that we will closely follow the type of partitions described in the Toy example in Section 7.3.1. Partition the the range of function  $f$  into  $H$  intervals given as  $\{a_r\}_{r=1}^{H-1}$ , where  $a_1 < a_2, \dots < a_{H-1}$ . Next we partition the inverse mapping of these intervals, i.e.,  $f^{-1}[a_r, a_{r+1}] \in \mathcal{X}$ , into  $S_r$  regions. There can be many methods to partition the data into  $S_r$  regions such as k-means clustering, hierarchical

clustering etc.. The choice of the clustering algorithm is a hyperparameter and our method allows us to choose any of these clustering algorithms. However, in this chapter, for ease of exposition, we would only use k-means clustering algorithms to partition the inverse images of the intervals. We characterize region  $i$  among these  $S_r$  regions by its centroid  $\mu_i^r$  (centroid is defined as the mean of the data points in the region). All the points in region  $i$  are closer to the centroid  $\mu_i^r$  than to the centroids of the rest of the regions. A region  $Z_k$  of the partition  $\mathcal{Z}$  is defined as  $Z_k = \{x, \text{ s.t. } x \in f^{-1}[a_r, a_{r+1}], \|x - \mu_i^r\| \leq \|x - \mu_j^r\|, \forall j\}$ . We already showed a 1-D partition in Fig. 7.1, we give another example in Fig. 7.3 of a 2-D partition to illustrate more complicated shapes of the regions in the partition. The partition in Fig. 7.3 has six regions.

To summarize, we follow the approach of dividing the range of the function  $f$  and then further dividing the inverse image of the  $f$  into regions characterized by their centroids and we fit a linear model to interpret the predictions in each region. The partition  $\mathcal{Z}$  is characterized by the interval values  $\{a_r\}_{r=1}^{H-1}, \{\{\mu_i^r\}_{i=1}^{S_r}\}_{r=1}^{H-1}$ . The total number of regions in the partition is  $\sum_{r=1}^{H-1} S_r$ .

### 7.3.2.1 Why this type of partitions?

Our main purpose when constructing the partition is to find “homogeneous” regions, i.e., regions where the data features  $x$  are close to each other and the corresponding predictions  $f(x)$  are also close to each other. To achieve the first task, i.e., the function values are close, we first partition the range of function  $f$ . However, just partitioning the range of  $f$  does not guarantee that the inverse image of the intervals consists of data instances that are also close to each other. See  $f^{-1}[a, b]$  in the Toy example in Fig. 7.1, it consists of two disconnected regions. This is the reason why we partition inverse image of each interval  $f^{-1}[a_r, a_{r+1}]$  so that any disconnected regions are separated into different regions.

### 7.3.2.2 Comment on choice of the number of intervals and number of regions

How many intervals  $H$  should we divide the function's range into? How many regions should we further subdivide those intervals into? In Fig. 7.1 and Fig. 7.2, we divide the feature space  $\mathcal{X}$  into four regions with different choices of  $H$  and  $\{S_r\}_{r=1}^H$ . We compare and select between the two choices using cross-validation. In general, if we want to construct  $K$  regions there can be many possible choices for  $H$  and  $\{S_r\}_{r=1}^H$ , where  $\sum_{r=1}^H S_r = K$ . In this chapter, we will carry out simulations assuming that  $S_r = S_t = W$  for any  $r \neq t$ . The more general case when  $S_r \neq S_t$  will be a part of future work.

Before we describe the main algorithm, we fix some hyperparameters. We assume that  $K$  is given (provided as input by the expert or a parameter that can be tuned using cross-validation). Suppose that we want to have  $K$  regions in the partition. We assume that we will divide the function's range space into  $H$  intervals and each interval into  $W$  regions. Therefore,  $H \times W = K$ .<sup>3</sup> Hence, the partition  $\mathcal{Z}$  is characterized by the interval values  $\{a_r\}_{r=1}^{H-1}, \{\{\mu_i^r\}_{i=1}^W\}_{r=1}^{H-1}$  where each region  $Z_k$  in  $\mathcal{Z}$  is  $Z_k = \{x, \text{ s.t. } x \in f^{-1}[a_r, a_{r+1}], \|x - \mu_i^r\| \leq \|x - \mu_j^r\|, \forall j\}$ . The set of all such partitions  $\mathcal{Z}$  is  $\mathcal{P}_K(\mathcal{X})^\dagger$ .

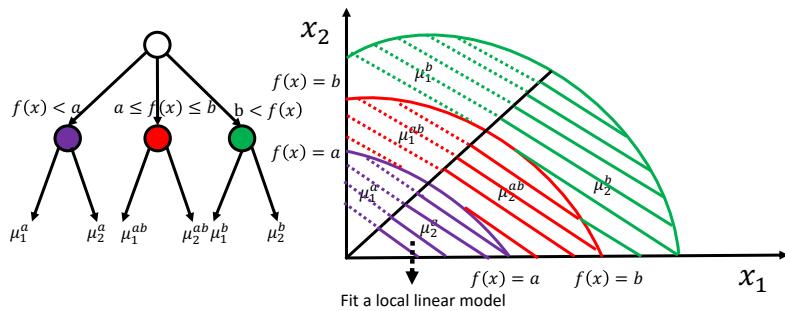


Figure 7.3: Example of a 2-D partition.

---

<sup>3</sup>The maximum number of choices for  $(H, W)$  are  $K$ .

### 7.3.3 Loss functions

We measure the goodness of fit of a proposed interpretation  $g$  for  $f$  in terms of a given *loss function* (for e.g., mean squared error). We assume  $\ell : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is a continuous, strictly increasing, strictly convex function such that  $\ell(0) = 0$ . We define the risk achieved by model  $g_{\mathcal{M}, \mathcal{Z}}$  as follows

$$R(f, g_{\mathcal{M}, \mathcal{Z}}; \mathcal{D}) = E_{\mathcal{D}}[l(\|f(X) - g_{\mathcal{M}, \mathcal{Z}}(X)\|_s)] \quad (7.1)$$

where  $X$  is a feature from the distribution  $\mathcal{D}$ , the expectation is taken over the distribution  $\mathcal{D}$ , and  $\|\cdot\|_s$  is the s-norm.

### 7.3.4 Risk Minimization

Our objective is to find a partition  $\mathcal{Z}$  and a map  $M : \mathcal{Z} \mapsto \mathcal{Y}$  (where each local model is drawn from  $\mathcal{H}$ ) to minimize the true risk subject to the constraint that the size of the partition, i.e.,  $|\mathcal{Z}| = K$ .

$$(\mathcal{M}^*, \mathcal{Z}^*) = \operatorname{argmin}_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} R(f, g_{\mathcal{M}, \mathcal{Z}}; \mathcal{D}) \quad (7.2)$$

$g_{\mathcal{M}^*, \mathcal{Z}^*}$  is the best piecewise model that minimizes the above risk.

### 7.3.5 Empirical Risk Minimization

In practice, we do not know the true distribution  $\mathcal{D}$  so we cannot minimize the true risk; instead, we see only a finite dataset (training set)  $D = \{(x_i, f(x_i))\}_{i=1}^N$  drawn from the true distribution. For given  $\mathcal{M}, \mathcal{Z}$  the empirical risk is

$$\hat{R}(\mathcal{M}, \mathcal{Z}; D) = \frac{1}{n} \sum_{(x_i, f(x_i)) \in D} \left[ l(\|f(x_i) - g_{\mathcal{M}, \mathcal{Z}}(x_i)\|_s) \right] \quad (7.3)$$

The spirit of Probably Approximately Correct (PAC) learning [SB14] suggests that we should minimize the empirical risk:

$$(\mathcal{M}^\dagger, \mathcal{Z}^\dagger) = \operatorname{argmin}_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D) \quad (7.4)$$

Later we will show that solving the above empirical risk minimization problem is PAC solution to the actual risk minimization problem in (7.2). We cannot solve the above problem using brute force search because it requires searching among  $\mathcal{O}(|D|^K)$  partitions, which becomes intractable very quickly with increase in  $|D|$  and  $K$ . In the next section, we propose an efficient Algorithm to solve the above problem.

## 7.4 Piecewise Local-Linear Interpreter

In this section, we develop the Piecewise Local-Linear Interpreter (PLLI) to solve the problem discussed above. Without loss of generality we assume that all the data points  $x_i$  in  $D$  are sorted in the increasing order of  $f(x_i)$ .

We give a summary of the working of the Algoirthm next. We give a detailed analysis of the Algorithm later. There are two parts to the Algorithm (Algorithm 5 and 6). In the first part, the Algorithm partitions the data  $D$  into subsets and finds an optimal local model corresponding to each subset. The division of the dataset into these subsets relies on dynamic programming. Suppose that the Algorithm wants to divide the first  $p$  points into  $q$  intervals. Also, suppose that the Algorithm has already constructed a partition to divide the first  $m$  points into  $k$  intervals for all  $m \leq p - 1$  and for all  $k \leq q - 1$ . The risk achieved by partition of  $m$  points where the size of the partition is  $k$  is defined as  $V(m, k)$ . For each  $x_i, x_j$  in the dataset, where  $i \leq j$ , define a subset of the data as follows.

$$D(i, j) = \{x : x \in D \& f(x_i) \leq f(x) < f(x_j)\}$$

Note that the dataset  $D(i, j)$  can be distributed in different regions of the feature space as shown in the Toy example in Section 7.3.1. Therefore, we divide the dataset  $D(i, j)$  into  $W$  regions using k-means clustering <sup>4</sup> with  $k = W$ , where the regions are given as  $\{S_1^{ij}, \dots, S_W^{ij}\}$ . We fit a linear model in each of these regions. We define the risk achieved over the dataset  $D(i, j)$  by these  $W$  local models as in (7.5).

---

<sup>4</sup>We can use other clustering methods as well instead.

$$G(i, j) = \sum_u \min_{h \in \mathcal{H}} \sum_{x_r \in S_u^{ij}} \left[ l(\|f(x_r) - h(x_r)\|_s) \right] \quad (7.5)$$

The Algorithm constructs a partition to divide  $p$  points into  $q$  intervals as follows

$$V(p, q) = \min_{n' \in \{1, \dots, p-1\}} \left[ V(n', q-1) + G(n'+1, p) \right]$$

$$\Phi(p, q) = \operatorname{argmin}_{n' \in \{1, \dots, p-1\}} \left[ V(n', q-1) + G(n'+1, p) \right]$$

where  $\Phi(p, q)$  is the index of the first data point in the  $q^{th}$  subset. The subset  $q$  consists of all the points indexed  $\{\Phi(p, q), \dots, p\}$ . The data subset defined as  $D(\Phi(p, q), p)$  and it is partitioned into  $W$  subsets each fitted with its own local model.

Similarly, the next subset, i.e., the  $(q-1)^{th}$  subset can be computed recursively as  $\{\Phi(\Phi(p, q), q-1), \dots, \Phi(p, q)-1\}$  and so on. In the first part of the Algorithm, we construct a partition of  $D$  and the corresponding set of local models. In the second part of the Algorithm, we extend this partition from the dataset  $D$  to the set  $\mathcal{X}$ . We write the function that is output by the Algorithm 6 as  $g_{M^\#, \mathcal{Z}^\#}$ .

## 7.5 Main Results

Our goal in this section is to show that the output of the Algorithm PAC learns  $f$ . We assume the special case when we only partition the range of  $f$  and do not further sub divide the intervals, i.e.,  $H = K$  and  $W = 1$ .

### 7.5.1 PAC learnability of Algorithms 1 and 2:

In this section, we discuss whether the outcome of Algorithm 5 and 6, i.e.  $g_{M^\#, \mathcal{Z}^\#}$  PAC learns  $f$ . We consider the case when we only partition the function's range into  $K$  intervals, i.e.,  $H = K$  and  $W = 1$ . In Fig. 7.2, we show example of such a case with  $H = 4$  and  $W = 1$ .

---

**Algorithm 5** Computing value and index functions

---

**Input:** Dataset  $D$ , Number of subsets  $K$

**Initialize:** Define  $V'(1, k) = 0, \forall k \in \{1, \dots, K\}$ .

For each  $x_i \in D, x_j \in D$  such that  $i \leq j$ , define  $D(i, j) = \{x : x \in D \text{ and } \|f(x_i)\| \leq \|f(x)\| \leq \|f(x_j)\|\}$

$\{S_1^{ij}, \dots, S_W^{ij}\} = \text{Kmeans}(D(h_l, h_u))$

$$G(i, j) = \sum_u \min_{h \in \mathcal{H}} \sum_{x_r \in S_u^{ij}} [l(\|f(x_r) - h(x_r)\|_s)]$$

$$M'(S_u^{ij}) = \min_{h \in \mathcal{H}} \sum_{x_r \in S_u^{ij}} [l(\|f(x_r) - h(x_r)\|_s)]$$

**for**  $n \in \{2, \dots, |D|\}$  **do**

**for**  $k \in \{1, \dots, K\}$  **do**

$$V'(n, k) = \min_{n' \in \{1, \dots, n-1\}} [V'(n', k-1) + G(n'+1, n)] \quad (7.6)$$

$$\Phi(n, k) = \operatorname{argmin}_{n' \in \{1, \dots, n-1\}} [V'(n', k-1) + G(n'+1, n)] \quad (7.7)$$

**Output:** Value function  $V'$ , Index function  $\Phi$

---

To show PAC learnability, we will first show that the outcome of Algorithms 5 and 6 achieves the minimum empirical risk.

**Proposition 4** *The output of the Algorithm 5 and 6 achieves the minimum risk value equal to  $\min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D)$ .*

The proofs to all the propositions and theorems are in the Appendix Section at the end of the chapter. The proof of Proposition 4 is given in the Appendix Section. We give a brief proof sketch next.

**Theorem 17**  $\forall \epsilon > 0, \delta \in (0, 1), \exists n^*(\epsilon, \delta)$  such that if  $D$  is drawn i.i.d. from  $\mathcal{D}$  and  $|D| \geq n^*(\epsilon, \delta)$ , then with probability at least  $1 - \delta$ ,

$$|R(f, g_{\mathcal{M}^\#, \mathcal{Z}^\#}; \mathcal{D}) - R(f, g_{\mathcal{M}^*, \mathcal{Z}^*}; \mathcal{D})| \leq \epsilon$$

The proof of Theorem 17 is in the Appendix Section at the end of the chapter.

---

**Algorithm 6** Computing partitions using the index function

---

- 1: **Input:** Index function  $\Phi$ , black-box predictive model  $f$
- 2: **Initialization:**  $h_u = |D|, r = 1$
- 3: **for**  $k \in \{1, \dots, H\}$  **do**
- 4:      $h_l = \Phi(h_u, K - k + 1)$
- 5:      $\{\mu_i^k\}_{i=1}^W = \text{Kmeans}(D(h_l, h_u))$
- 6:      $\sum_{x_r \in D(i,j), x_r \in W_s} [l(\|f(x_r) - h(x_r)\|_s)]$
- 7:     **for**  $u \in \{1, \dots, W\}$  **do**
- 8:          $Z_{K-r+1} = \{x : f(x_{h_l}) < f(x) \leq f(x_{h_u}), \|x - \mu_r^k\| \leq \|x - \mu_j^k\|\}$
- 9:          $M_{K-r+1} = M'(S_u^{h_l h_u})$
- 10:          $h_u = h_l$
- 11:          $r = r + 1$
- 12:
- 13: **Output:**  $\mathcal{Z}^\# = \{Z_1, \dots, Z_K\},$
- 14:  $\mathcal{M}^\# = \{M_1, \dots, M_K\}$

---

## 7.6 Experiments

In this section, we describe the experiments conducted on synthetic and real datasets. We will cover regression problems in this experiments section. The proposed method can also be applied to classification problems. All the simulations were conducted in Python in Google Colab.

### 7.6.1 Metrics

We will use two metrics to measure the performance of the different methods. We denote mean squared error as MSE. We use MSE to measure the performance of the model on the labelled data (squared of the norm of the difference between the predictions and labels), which is denoted as MSE-p. We also use MSE to measure the fidelity (squared of the norm of the difference between the predictions and black-box function values), i.e., how close is the model to the black-box model, which is denoted as MSE-f. We use  $R^2$ , i.e., the coefficient of determination, to measure the fit of the model.

### 7.6.2 Synthetic Dataset

We begin by describing a synthetic dataset that we use to illustrate the performance of the method before going into more complicated real data setting. We assume that each data point is of the form  $(x_1, x_2, y)$ , where  $x_1$  and  $x_2$  are the features and  $y = (x_1 + x_2)^2$ . We assume that  $x_1$  and  $x_2$  are independent and are drawn from  $\mathcal{N}(0, 1)$ . We sample 1000 data points.

#### 7.6.2.1 Black-Box Model

We split the data randomly into 80 percent training and 20 percent testing. We fit a random forest regressor to predict the target variable. In the Table 7.1, we compare the performance of the RF regressor, which is a black-box method, with other more interpretable methods such as regression tree and linear model. We observe that the RF regressor has a much

smaller mean squared error (MSE) in comparison to a linear model or a regression tree. We do not report  $R^2$  for linear model and regression tree as the models are so poor fit that we obtained a negative  $R^2$ .

### 7.6.2.2 Piecewise Local-Linear Interpreter

There are several possible configurations for the piecewise local-linear interpreter. We will fix  $K = 4$  (we fix a small value as the dataset is small). We have three parameter configurations possible.  $(H = 4, W = 1)$ ,  $(H = 1, W = 4)$ ,  $(H = 2, W = 2)$ . Instead of using the dynamic programming procedure described in Algorithm 5, we can alternatively use a simpler procedure to partition the function's range. We first order the dataset in terms of the black-box predictions and divide the dataset into  $H$  equal quantiles. We use k-means clustering for data in each quantile to divide the data into  $W$  clusters and fit a local-linear model to it. We refer to this procedure as EQ-PLLI, where EQ stands for equal quantile. On the other hand, we refer to the procedure from Algorithm 5 and 6 as OP-PLLI, where OP stands for optimal. In Table 7.2, we compare these configurations in terms of MSE-f and MSE-p. Based on MSE-f and MSE-p we select the OP-PLLI ( $H = 2, W = 2$ ). We also compare with regression tree (with four leaves as  $K = 4$ ) and linear model fitted to predict the black-box model.

### 7.6.2.3 Model Summary

In Figs. 7.4, 7.5, we show the partitions constructed under the different models shown in Table 7.2. In Fig. 7.4 a) and Fig. 7.5 a), we show the different partitions from equal quantile and optimal approach. We can observe that the regions in each partition in Fig. 7.4 a) and Fig. 7.5 a) are not contiguous. For instance, the orange region is located in two separate regions of the 2-D feature space. In Fig. 7.4 b) and c) the regions in each partition are contiguous. In Table 7.3, we give the model summary constructed based on the piecewise interpreter ( $W = 2, H = 2$ , OP). Each region characterized by the centroid and the function range and the corresponding coefficients of  $x_1$  and  $x_2$  are shown in Table 7.3.

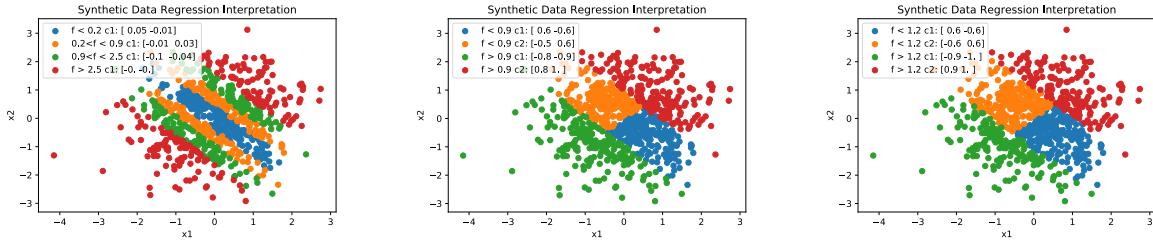


Figure 7.4: Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 4, W = 1$ ) EQ, b) ( $H = 2, W = 2$ ) EQ, c) ( $H = 2, W = 2$ ) OP.

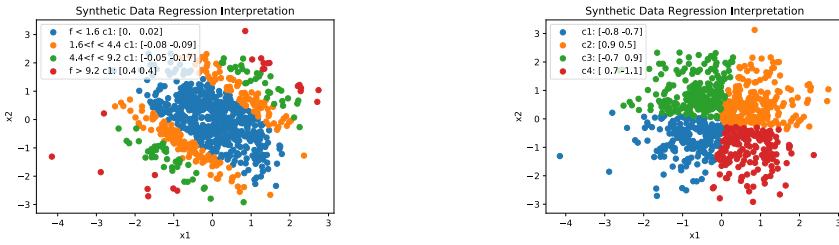


Figure 7.5: Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a) ( $H = 4, W = 1$ ) OP, b) ( $H = 1, W = 4$ ) OP.

Table 7.1: Comparison of RF Regressor with other methods.

Model	MSE-p	R <sup>2</sup>
RF Regressor	0.11	0.97
Regression tree	6.30	0.00
Linear model	5.15	0.05
Constant model	5.43	—

### 7.6.3 Interpret RF regression on Boston Housing Dataset

We use the Boston Housing Dataset from UCI repository. The dataset consists of information about the house prices and other attributes about where the house is located. The attributes

Table 7.2: Comparison of RF Regressor with other methods.

Model	MSE-f	MSE-p
EQ-PLLI ( $H = 4, W = 1$ )	1.19	1.52
OP-PLLI ( $H = 4, W = 1$ )	0.54	0.73
PLLI ( $H = 1, W = 4$ )	0.69	0.70
EQ-PLLI ( $H = 2, W = 2$ )	0.18	0.12
OP-PLLI ( $H = 2, W = 2$ )	0.18	0.11
Linear model	5.75	7.57
Regression tree	4.34	5.15

Table 7.3: Model Summary of RF Regressor: Synthetic Data.

Region	x1	x2
$f < 1.2, [0.6, -0.6]$	0.104	0.002
$f < 1.2, [-0.6, 0.6]$	0.070	0.004
$f > 1.2, [-0.9, -1.0]$	4.230	4.230
$f > 1.2, [0.9, 1.0]$	4.230	4.230

with their abbreviations and descriptions are described below.

- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940

- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per 10,000
- PTRATIO pupil-teacher ratio by town
- $B 1000(Bk - 0.63)^2$  where  $Bk$  is the proportion of blacks by town
- LSTAT lower status of the population

The total number of instances in the dataset is 506.

#### 7.6.3.1 Black-Box Model

We split the data randomly into 80 percent training and 20 percent testing. We fit a random forest regressor to predict the target variable, i.e., the price of the house based on the attributes described above. In Table 7.4, we compare the MSE-p of the random forest method and compare it with other interpretable methods such as a linear model and a regression tree model. We observe that the RF regressor has a much smaller mean squared error (MSE) in comparison to a linear model or a regression tree. However, the improvement in the performance comes at the cost that the new model is harder to interpret. In the next section, we use the proposed procedure to get insights into the behavior of this random forest regressor model.

Table 7.4: Comparison of RF Regressor with other methods.

Model	MSE-p	R <sup>2</sup>
RF Regressor	8.217	0.88
Regression tree	37.23	0.60
Linear model	21.78	0.78
Constant model	81.54	0.78

Table 7.5: Comparison of PI interpreter different hyperparameter configurations.

Model	MSE-f	MSE-p
EQ-PLLI ( $H = 4, W = 1$ )	5.76	15.10
OP-PLLI ( $H = 4, W = 1$ )	3.40	10.05
PLLI ( $H = 1, W = 4$ )	8.80	11.32
EQ-PLLI ( $H = 2, W = 2$ )	5.83	13.01
OP-PLLI ( $H = 2, W = 2$ )	6.40	12.40
Linear model	16.79	21.78
Regression tree	21.17	37.27

### 7.6.3.2 Piecewise Local-Linear Interpreter

There are several possible configurations to use for the piecewise interpreter. We fix  $K = 4$  (as the dataset is small). We have three parameter configurations possible  $H = 4, W = 1$ ,  $H = 1, W = 4$ ,  $H = 2, W = 2$ . We compare the fidelity (how well does the piecewise model represent the black-box) and MSE of these models (how well does the piecewise model perform when making the predictions of the labels) in the Table 7.5. We select the model based on the fidelity value. We select the OP-PLLI model with  $H = 4, W = 1$ .

### 7.6.3.3 Black-Box Model Summary

We present the model summary in Tables 7.6 and 7.7. The table's different rows shows the different regions in the partition and the importance associated with different features. In Figs. 7.6 and 7.7, we show the different regions in the partitions. We use the two components of the PCA to represent the features. Based on the different parameters we get different partitions. In Figs. 7.6 and 7.7, the regions in the partitions are contiguous. The partition in Fig. 7.6 b) and Fig. 7.7 b) offers the additional advantage that the different data points in the partition do not overlap in the two dimensional space. If the data points do not

overlap in the two dimensional space, then that implies that each region in the partition can be simply described in terms of the two PCA components only and does not need to use black-box output  $f$  to define the region.

Table 7.6: Comparison of RF Regressor with other methods.

<b>Region</b>	<b>CRIM</b>	<b>ZN</b>	<b>INDUS</b>	<b>CHAS</b>	<b>NOX</b>	<b>RM</b>	<b>AGE</b>
$f < 19.0, [2.0, 0.0]$	0.63	0.79	0.68	0.41	1.37	0.14	0.05
$19 < f < 26.0, [-0.8, -0.2]$	0.40	0.17	0.11	0.22	0.08	1.38	0.61
$26.0 < f < 35.0, [-2.0, 0.3]$	0.0	0.62	1.39	0.10	0.16	0.95	0.57
$f > 35.0, [-2.0, 0.3]$	6.61	0.03	2.51	0.08	4.69	2.66	1.34

Table 7.7: Comparison of RF Regressor with other methods.

<b>Region</b>	<b>DIS</b>	<b>RAD</b>	<b>TAX</b>	<b>PTR</b>	<b>B</b>	<b>LSTAT</b>
$f < 19.0, [2.0, 0.0]$	0.31	0.73	1.46	1.04	0.18	1.59
$19 < f < 26.0, [-0.8, -0.2]$	0.63	0.94	0.72	0.58	0.28	1.43
$26.0 < f < 35.0, [-2.0, 0.3]$	0.94	2.75	1.57	0.99	0.0	2.27
$f > 35.0, [-2.0, 0.3]$	2.71	7.74	3.04	3.02	0.0	2.71

#### 7.6.3.4 Submodular Pick

In [RSG16], the authors proposed a method to identify candidate data points to provide local instance based explanations. The proposed method is called submodular pick. The method tries to ensure that data points that are selected present a diverse set of feature importance. However, the selected data points do not necessarily represent a diversity in terms of the feature distribution or the predicted-value distribution. Another approach to select the data points is to select them randomly.

Our method provides a natural way to select the candidate data points. The data points selected by our method are the centroids of each region in the partition identified by the

PLLI. Suppose we want to identify  $K$  candidate data points. In this case, the size of partition we use is  $K$  using PLLI.

We measure how well spread are the data points identified spread in the space as follows. For each selected point compute the distance from the nearest neighbor. We define coverage as the average minimum distance from the neighbors , i.e.,  $\frac{1}{K} \sum_i \min_{j \neq i} \|x_i - x_j\|$ . We measure the coverage for the feature importance, the coverage for the feature vectors, and the coverage for function values. We compare the proposed procedure (for  $K = 4$ ) with random method (averaged over 10 runs) and the submodular pick method. In Table 7.8, we compare the various methods. We observe that the proposed method is better at giving a larger coverage in terms of feature values, predicted values and importance values. In Fig. 7.8 we show the different data points (predicted values, explanations and features) identified by the proposed method.

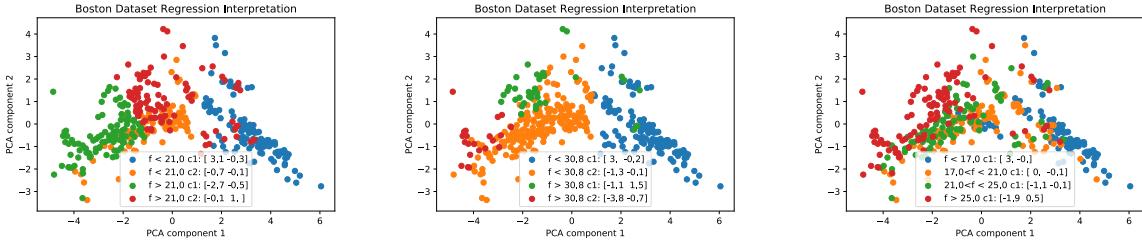


Figure 7.6: Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a)  $(H = 2, W = 2)$  EQ, b)  $(H = 2, W = 2)$  OP, c)  $(H = 4, W = 1)$  EQ.

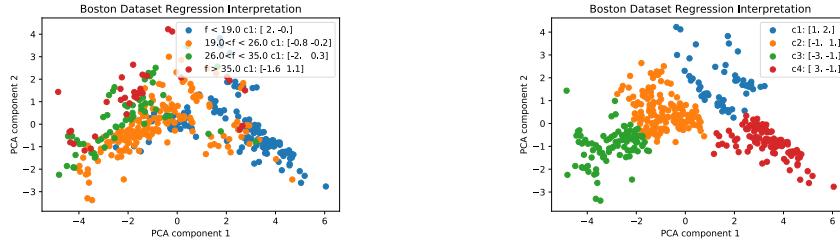


Figure 7.7: Comparison of PLLI for different hyperparameter configurations. Figures above from left to right have the following parameter configurations a)  $(H = 4, W = 1)$  OP, b=  $(H = 1, W = 4)$  OP.

Table 7.8: Comparison of coverage of various selection methods.

Algorithm	Coverage importance	Coverage predictions	Coverage features
PI	0.68	8.69	3.80
Submodular pick	0.63	4.38	3.20
Random	0.60	3.87	3.75

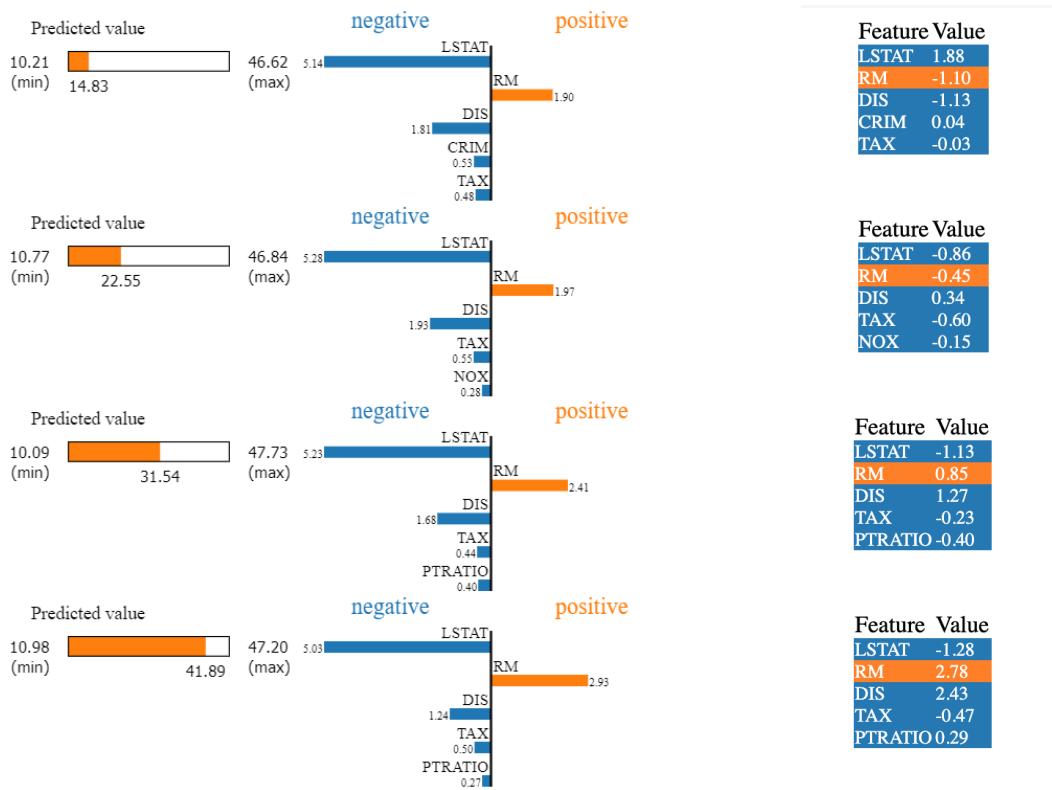


Figure 7.8: Data points selected based on PLLI and corresponding explanations (for the top five features.)

#### 7.6.4 Large dataset

The computational complexity of the proposed approach is large (See Proposition 5). It is easy to approximate the PLLI procedure to allow it to scale to large datasets (see the description in the Appendix). In this section, we show that an approximate version of PLLI can scale well for large datasets as the experiments in the previous sections were done on

datasets that were moderately small (500-1000 datapoints). In this section, we use California Housing Dataset from StatLib library below. It consists of 20,640 data points with 8 features. The 8 features are described as

- MedInc median income in block
- HouseAge median house age in block
- AveRooms average number of rooms
- AveBedrms average number of bedrooms
- Population block population
- AveOccup average house occupancy
- Latitude house block latitude
- Longitude house block longitude.

The target variable is the median house value for California districts.

We compare the performance of EQ PLLI ( $H = 4, W = 1$ ) with approximate OP PLLI ( $H = 4, W = 1$ ). The goal is to show that with a reasonable computation time the proposed approximation approach performs well. In Table 7.9, we compare the RF regressor with more interpretable methods. In Table 7.10, we show the comparison of PLLI method. Note that approximate OP-PLLI took 700 seconds to train, while the exact OP-PLLI would take 3 days to train.

## 7.7 Connection with K-means clustering

In this section, we begin by drawing a connection between the equation (7.4) and the general problem of clustering.

Table 7.9: Comparison of RF Regressor with other methods.

Model	MSE-p	R <sup>2</sup>
RF Regressor	0.24	0.80
Regression tree	0.74	0.58
Linear model	0.53	0.60

Table 7.10: Comparison of PLLI interpreter different hyperparameter configurations.

Model	MSE-f	MSE-p
EQ-PLLI ( $H = 4, W = 1$ )	0.084	0.33
Approximate-OP-PLLI ( $H = 4, W = 1$ )	0.076	0.26

### 7.7.1 Ordered Partitions

We say that the partition  $\mathcal{Z}$  of  $A \subset \mathcal{X}$  is *ordered* if for every  $Z, Z' \in \mathcal{Z}$  with  $Z \neq Z'$ , either

- (i) for all  $z \in Z, z' \in Z'$  we have  $f(z) < f(z')$ , or
- (ii) for all  $z \in Z, z' \in Z'$  we have  $f(z) > f(z')$

We consider the same setting as in the Section 7.5, where  $H = K$  and  $W = 1$ , i.e., we only want to optimize how to divide the function's range. Hence, we only search in the space of  $\{a_r\}_{r=1}^{H-1}$ , which characterize the different intervals that are possible. Recall that we define the set of partitions that we search in as  $\mathcal{P}_K(\mathcal{X})^\dagger$ . Consider any two regions  $Z_i = f^{-1}[a_m, a_n]$  and  $Z_j = f^{-1}[a_p, a_q]$  in partition  $\mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger$ .  $[a_m, a_n]$  and  $[a_p, a_q]$  are non-overlapping intervals by construction. Hence, the two regions  $Z_i$  and  $Z_j$  are ordered. Therefore,  $\mathcal{P}_K(\mathcal{X})^\dagger$  is the set of all the ordered partitions of size  $K$ .

Suppose we set all the coefficients in the linear model except for the intercept to zero, then we get a constant model. We reformulate the problem (7.4). We expand the search to the space of all the partitions of size  $K$ ,  $\mathcal{P}_K(\mathcal{X})$ , instead of just ordered partitions and restrict the search to local-constant models.

$$(\mathcal{M}^\dagger, \mathcal{Z}^\dagger) = \operatorname{argmin}_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})} \hat{R}(\mathcal{M}, \mathcal{Z}; D) \quad (7.8)$$

We simplify the above problem. Suppose  $\{Z_1, \dots, Z_K\}$  are the regions in the partition and  $\{c_1, \dots, c_k\}$  are the corresponding locally constant model values. We write the loss as follows

$$\hat{R}(\mathcal{M}, \mathcal{Z}; D) = \frac{1}{n} \sum_{k=1}^K \sum_{x_i \in Z_k} [l(\|f(x_i) - c_k\|_s)] \quad (7.9)$$

Let  $f(x_i) = y_i$ . Suppose  $\mathcal{I} = \{I_1, \dots, I_K\}$  is a partition of  $\{y_1, \dots, y_n\}$  in  $K$  regions. We rewrite (7.9) as

$$\hat{R}(\mathcal{M}, \mathcal{I}; D) = \frac{1}{n} \sum_{k=1}^K \sum_{y_i \in I_k} [l(\|y_i - c_k\|_s)] \quad (7.10)$$

We formulate risk minimization with (7.10) as objective

$$(\mathcal{M}^\dagger, \mathcal{I}^\dagger) = \operatorname{argmin}_{\mathcal{M} \in \mathcal{H}^K, \mathcal{I} \in \mathcal{P}_K(\mathcal{Y})} \hat{R}(\mathcal{M}, \mathcal{I}; D) \quad (7.11)$$

Observe that the two optimization problems (7.8) and (7.11) are equivalent. It is easy to show this using contradiction.

Also, observe that the optimization problem (7.11) is a general clustering problem for one-dimensional data  $\{y_1, \dots, y_n\}$ . Next we prove that the output of Algorithm 5 and 6 achieves the optimal clustering in polynomial time. We first begin by getting worst case complexity bounds for the Algorithm 5 and 6.

**Proposition 5** *If the loss function is mean squared error and the local model is from constant model family, then the computational complexity of Algorithm 5 and 6 together is  $\mathcal{O}(|D|^3 K d)$ .*

If we set  $l$  to be a squared function and the norm  $s = 2$  in (7.3), we obtain a MSE minimization problem. The classic k-means clustering method also tries to minimize the same objective. In the next proposition, we state that the output of Algorithm 5 and 6 achieves the optimal clustering.

**Proposition 6** *If the loss function is mean squared error and the local model is from constant model family, then the output of the Algorithm 5 and 6 achieves optimal clustering, i.e.,  $\min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{I} \in \mathcal{P}_K(\mathcal{Y})} \hat{R}(\mathcal{M}, \mathcal{I}; D) = \hat{R}(\mathcal{M}^\#, \mathcal{Z}^\#; D)$*

Note that we have established that the Algorithm 5 and 6 achieves optimal clustering in polynomial time. Methods in the literature such as k-means clustering are not guaranteed to achieve the optimal clustering even for the one-dimensional case that we described above. In the next proposition we discuss the general clustering problem, when we only require  $l$  to be strictly convex and the norm can be any norm  $s \geq 1$ .

**Proposition 7** *If the local model is from constant model family, then for every  $\epsilon, \delta > 0$  and every  $K$  there is some  $m^*(\epsilon, \delta, K)$  such that if the training set  $D$  is drawn i.i.d. from the distribution  $\mathcal{D}$  and  $|D| \geq m^*(\epsilon, \delta, K)$ , then with probability at least  $1 - \delta$  we have*

$$\left| \min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{I} \in \mathcal{P}_K(\mathcal{Y})} \hat{R}(\mathcal{M}, \mathcal{I}; D) - \hat{R}(\mathcal{M}^\#, \mathcal{Z}^\#; D) \right| < \epsilon$$

## 7.8 Conclusion

This chapter provides a novel way to construct piecewise approximations of a black-box model. Our approach uses dynamic programming to partition the feature space into regions and then assigns a simple local model within each region. We carry out experiments show that the proposed approach achieves a smaller loss and better reflects the black-box model compared to other approaches. We also prove that the proposed approach can also be applied to the problem of clustering. We provide a first proof that the proposed approach achieves optimal clustering in polynomial time when the data is one-dimensional.

## 7.9 Appendix

### 7.9.1 Appendix A

**Proof of Proposition 4.** Throughout this Appendix, we will assume that the partitions are purely constructed based on the division of the range of  $f$  in intervals (Recall the assumption

$H = K$  and  $W = 1$ ).

**Bellman Principle** Let  $\mathcal{Z}$  be an ordered partition of  $\mathcal{X}$  and assume that  $R(\mathcal{Z}, D)$  minimizes the risk among all ordered partitions of  $\mathcal{X}$  with at most  $|\mathcal{Z}|$  elements. If  $\mathcal{Z}', \mathcal{Z}'' \subset \mathcal{Z}$  is a partition of  $\mathcal{Z}$ , then  $R(\mathcal{Z}', D)$  minimizes the risk among all ordered partitions of  $A'$  with at most  $|\mathcal{Z}'|$  elements. (If this were not true then we could find another ordered partition  $\mathcal{Z}^*$  of  $A'$  with lower risk. But then  $\mathcal{Z}^* \cup \mathcal{Z}''$  would be an ordered partition of  $\mathcal{X}$  with lower risk than  $\mathcal{Z}$ , which would be a contradiction.)

Suppose that the first  $n$  points are to be divided into  $k$  regions. The minimum risk achieved by the optimal partition of first  $n$  points into  $k$  regions satisfies the Bellman equation given as.

$$V(n+1, k) = \min_{n' \in \{1, \dots, n\}} \{V(n', k-1) + G(n'+1, n)\}$$

Note that  $V(1, k) = 0$  for all  $1 \leq k \leq K$  (the model  $M$ 's output is equal to the one data point itself)

To prove this proposition, we first state a lemma in the next section.

### 7.9.1.1 Appendix B

**Lemma 4** *The value function output by the Algorithm 5, which is  $V'$ , is the same as the true value function  $V$ , i.e.,  $V' = V$  thus  $V'(n, K) = \min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D)$*

**Proof of Lemma 4.** We use induction in the number of data points  $n$ .

We start with the base case  $n = 1$ . For  $n = 1$  and  $1 \leq k \leq K$ , we know that the  $V'(1, k) = 0$  (from the initialization of the Algorithm). We also know that  $V'(1, k) = 0$  for all  $1 \leq k \leq K$  (the model  $M$ 's output is equal to the one data point itself). Hence, the claim is true for  $n = 1$  and for all  $1 \leq k \leq K$ .

Suppose that the Algorithm outputs optimal value functions for all  $s \leq n$  and for all  $k \leq K$ .

Consider the data point  $n + 1$  and the constraint on the number of partitions is  $k$ . From the Algorithm 5 we know that

$$V'(n + 1, k) = \min_{n' \in \{1, \dots, n\}} \{V'(n', k - 1) + G(n' + 1, n)\}$$

Let us assume that  $V'(n + 1, k)$  is not optimal, i.e.,

$$V'(n + 1, k) > V(n + 1, k)$$

We use the Bellman principle to write the value function  $V$  as follows

$$V(n + 1, k) = V(n^*, k - 1) + G(n^* + 1, n)$$

We use the above two equations to write

$$V'(n + 1, k) > V(n^*, k - 1) + G(n^* + 1, n)$$

We also know that

$$V'(n + 1, k) < V'(n^*, k - 1) + G(n^* + 1, n)$$

Therefore, we can write

$$\begin{aligned} V'(n^*, k - 1) + G(n^* + 1, n) &> V(n^*, k - 1) + G(n^* + 1, n) \\ \implies V(n^*, k - 1) &> V(n^*, k - 1) \end{aligned}$$

This contradicts the assumption that  $V(n^*, k - 1) = V'(n^*, k - 1)$ . Hence, the assumption  $V'(n + 1, k) > V(n + 1, k)$  cannot be true, which implies  $V'(n + 1, k) \leq V(n + 1, k)$ . Thus we can say that  $V'(n + 1, k) = V(n + 1, k)$  ( $V'(n + 1, k) < V(n + 1, k)$  can't be true since  $V$  is optimal value function). ■

In Lemma 4, we showed that  $V' = V$ . To complete the proof of Proposition 4, we need to show that the partition output by the Algorithm 6 achieves  $V$ .

Recall the computation of value function from Algorithm 5

$$V'(n + 1, k) = \min_{n' \in \{1, \dots, n\}} \{V'(n', k - 1) + G(n' + 1, n)\}$$

From Algorithm 5, we also know that

$$\Phi(n+1, k) = \underset{n' \in \{1, \dots, n\}}{\operatorname{argmin}} \{V'(n', k-1) + G(n'+1, n)\}$$

We can write

$$V'(n+1, k) = V'(\Phi(n, k), k-1) + G(\Phi(n, k) + 1, n)$$

The subset of the data upto data point  $n$  is written as  $D_n$ . The optimal partition with  $n+1$  points and at most  $k$  regions induces a partition of the dataset  $D_{n+1}$ . We write the last region of the induced partition on  $D_{n+1}$  as  $S_k$ . We know that  $S_k = \{\Phi(n, k) + 1, \dots, n\}$ .

We can repeat this procedure recursively and define  $S_{k-1}$  and so on. The set of points  $S_{k-1} = \{\Phi(\Phi(n, k), k-1), \dots, \Phi(n, k)\}$  is the set of points that belong to the region  $k-1$  and so on. This induced partition achieves the risk value of  $V'(n+1, k)$

We require that the partition constructed in Algorithm 6 also divides the points in the dataset in the exact same manner as described above.

Consider the region  $Z_K$  output by the Algorithm 6.

$$Z_K = \{x : f(x_{\Phi(|D|, K)}) < f(x) \leq f(x_{|D|})\}$$

The points  $\{\Phi(|D|, K) + 1, \dots, |D|\}$  are ordered and thus all of these belong to  $Z_K$ . The same argument applies to the set  $Z_{K-1}$  (given below) and the set of points  $\{\Phi(\Phi(|D|, K), K-1) + 1, \dots, \Phi(|D|, K)\}$  and so on.

$$Z_{K-1} = \{x : ||f(x_{\Phi(\Phi(|D|, K), K-1)})|| < ||f(x)|| \leq ||f(x_{\Phi(|D|, K)})||\}$$

Observe that the division of the points is the same as prescribed by the value function  $V'$ . Hence, the output of Algorithm 6 achieves  $V'$  and from Lemma 4 we know it is equal to the minimum risk. ■

### 7.9.2 Appendix C

**Proof of Theorem 17.** From Proposition 4, we know that Algorithm 5 and 6 combined solve the empirical risk minimization problem. In this theorem, we claim that the empirical risk minimization (ERM) actually leads to successful agnostic PAC learning.

Consider the MSE loss. The feature space is  $\mathcal{X}$  and the label space is  $\mathcal{Y}$ . We consider a discretization of the label space  $\mathcal{Y}_d$ . This discretization trick is fairly common see [SB14].

If  $\mathcal{Y} = [0, 1]$ , then a quantization of  $\mathcal{Y}$  into steps of length  $\Delta$  is given as  $\mathcal{Y}_d = \{0, \Delta, 2\Delta, \dots, 1\}$ . Suppose the partition of the dataset is  $\{D_1, \dots, D_K\}$  and the corresponding set of local-linear models are  $\{M_1, \dots, M_K\}$ . Each  $M_k : \mathcal{X} \rightarrow \mathcal{Y}$  (If we use a local-linear model we can clip the linear function and keep it bounded between  $[0, 1]$ ). We write the MSE loss for the continuous labels/black-box predicted values as

$$MSE_c = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - M_k(x))^2$$

where  $y = f(x)$ . Suppose the discretization of a label  $M_k(x)$  is  $y_d \in \mathcal{Y}_d$  (we discretize a point by finding the closest point from  $\mathcal{Y}_d$ ). We write the discretized MSE as

$$MSE_d = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - y_d)^2$$

We compare these MSEs

$$|MSE_c - MSE_d| = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} |(2y - y_d - M_k(x))| |M_k(x) - y_d| \leq 4\Delta \quad (7.12)$$

The same applies to the expected MSE as well.

$$|E[MSE_c] - E[MSE_d]| \leq 4\Delta \quad (7.13)$$

For the sake of clearer exposition, let us assume that the hypothesis class that we want to optimize over is of constant models  $M_k(x) = c$  instead of linear models in (7.2). We will show at the end how to extend the proof to linear class. The MSE loss  $MSE_c$  is minimized when  $\bar{y}_k = \frac{1}{|D_k|} \sum_{x \in D_k} f(x)$  and we write the loss w.r.t.  $\bar{y}_k$ .

$$MSE_c^* = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - \bar{y}_k)^2$$

Suppose we restrict the search of the minimizer  $\bar{y}_k$  to discretized values as well. In that case, we take the discretization of the sample mean  $\bar{y}_k$ , which we denote as  $\bar{y}_k^d \in \mathcal{Y}_d$  as the discretized optimal solution. We write the mean square error with  $\bar{y}_k^d$  as

$$MSE_c^\dagger = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - \bar{y}_k^d)^2$$

The difference between the MSEs is given as follows.

$$|MSE_c^\dagger - MSE_c^*| = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} |(2y - \bar{y}_k - \bar{y}_k^d)| |\bar{y}_k^d - \bar{y}_k| \leq 4\Delta \quad (7.14)$$

If the discretization level  $\Delta$  is sufficiently small, finding the minimizer in the discrete space is almost as good as the minimization in the continuous space.

We also want to discretize the partitions. Next, we consider the effect of discretization.

In the above expression (7.14) each set  $D_k$  is obtained based on the partition of the continuous label space  $\mathcal{Y} = [0, 1]$ . If we limit the search to the partitions based on the discreteized label space  $\mathcal{Y}_d$ , then also it is easy to show that the difference due to the quantization can be made arbitrarily small. Let us consider a region  $D_k$  defined as follows. If  $x \in D$  and  $f(x) \in [a, b]$ , then  $x \in D_k$ . Let us consider the discretized version of  $D_k$  given as  $D_k^d$  defined as follows. If  $f(x) \in [a_d, b_d]$ , then  $x \in D_k^d$ , where  $a_d, b_d$  are discretizations of  $a, b$ . Define the difference the total errors for the two regions as follows.

$$\begin{aligned} SE(D_k) - SE(D_k^d) &= \sum_{x \in D_k} (f(x) - \mu)^2 - \sum_{x \in D_k^d} (f(x) - \mu)^2 = \\ &\quad \sum_{x \in D_k \cap [D_k^d]^c} (f(x) - \mu)^2 - \sum_{x \in D_k^c \cap D_k^d} (f(x) - \mu)^2 \end{aligned} \quad (7.15)$$

where  $A^c$  is the complement of set  $A$ . The difference between the loss defined over these partitions will vary the most when these partitions are maximally different, which happens when  $a = a + \Delta$  and  $b = b - \Delta$ .

$$SE(D_k) - SE(D_k^d) = \sum_{f(x) \in [a, a + \Delta] \cup [b - \Delta, b], x \in D} (f(x) - \mu)^2$$

Since  $f(x) \in [0, 1], \mu \in [0, 1] \implies (f(x) - \mu)^2 \leq 1$ . Therefore,  $SE(D_k) - SE(D_k^d)$  is bounded above by the number of points in  $[a, a + \Delta] \cup [b - \Delta, b]$ . The total number of points in  $[a, a + \Delta] \cup [b - \Delta, b]$  as the data becomes large is  $\approx \Pr(f(x) \in [a, a + \Delta] \cup [b - \Delta, b])|D|$ .

If  $\Delta$  is very small and if we assume that the cumulative distribution function (c.d.f) of  $f$  is continuous, then it follows that  $Pr(f(x) \in [a, a + \Delta] \cup [b - \Delta, b])|D| = v(a)\Delta|D| + v(b)\Delta|D|$ , where  $v$  is the probability density function (p.d.f) of  $f$ . Therefore,

$$SE(D_k) - SE(D_k^d) \leq (v(a) + v(b))\Delta|D| \quad (7.16)$$

From the above (7.16) it follows that the difference in the MSEs can be made arbitrarily small. Observe that the objective function in (7.2) only consists of the values of  $f$ . Also, observe that the partition of  $\mathcal{X}$  creates a partition of the domain of  $f$ , i.e.,  $\mathcal{Y}$ . To solve (7.2) one can equivalently search over all the ways to partition  $\mathcal{Y}$  into  $K$  intervals. Hence (7.2) can be restated as

$$\min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{Y})^\dagger} R(f, g_{M, \mathcal{Z}}; \mathcal{D}) \quad (7.17)$$

where  $\mathcal{P}_K(\mathcal{Y})^\dagger$  is the set of all the partitions of  $\mathcal{Y}$  into  $K$  regions where each region is an interval. If we restrict the boundary of each interval to be from the set  $\mathcal{Y}_d$ , then the set of all the discretized partitions  $\mathcal{P}_K^d(\mathcal{Y})^\dagger$ .

We reformulate (7.2) in terms of discretized partitions and discretized hypothesis class as follows.

$$\min_{\mathcal{M} \in \mathcal{H}_d^K, \mathcal{Z} \in \mathcal{P}_K^d(\mathcal{Y})^\dagger} R(f, g_{M, \mathcal{Z}}; \mathcal{D}) \quad (7.18)$$

where  $\mathcal{H}_d$  is the set of discretized constant models. From the equation (7.13), (7.14) and (7.16) the minimizer of (7.18) will be very close to the minimizer of (7.2).

We also reformulate the empirical risk minimization (7.4) in terms of the discrete space below.

$$\min_{\mathcal{M} \in \mathcal{H}_d^K, \mathcal{Z} \in \mathcal{P}_K^d(\mathcal{Y})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D) \quad (7.19)$$

We know that the hypothesis class for the above problem is finite. We use Corollary 4.6 in [SB14] to arrive at the conclusion that empirical risk minimization leads to successful PAC learning of the hypothesis class of all the discrete partitions. Hence, solving (7.19) (Algorithm 5 and 6's output leads to a solution close to (7.19)) leads to solving (7.18).

Let us extend the proof to linear hypothesis class. Each linear function  $\beta^t x$  is characterized by a vector  $\beta$  and we also assume that  $\|\beta\|$  is bounded above by a value  $M$ . Suppose we discretize each component of this vector and search for the optimal solution in this discretized space. The vector  $\beta$  has a bounded norm  $\implies$  the discretized space of linear functions consists is a finite set. The MSE loss  $MSE_c$  is minimized when  $\bar{y}_k(x) = (\beta^*)^t x$  and we write the loss w.r.t.  $\bar{y}_k(x)$ . The discretization of  $\beta^*$  is given as  $[\beta^*]_d$ . Therefore,  $\bar{y}_k^d(x) = [\beta^*]_d^t x$

$$MSE_{c,l}^* = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - \bar{y}_k(x))^2$$

$$MSE_{c,l}^\dagger = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} (y - \bar{y}_k^d(x))^2$$

The difference between the MSEs is given as follows.

$$|MSE_{c,l}^\dagger - MSE_{c,l}^*| = \frac{1}{|D|} \sum_{k=1}^K \sum_{x \in D_k} |(2y - \bar{y}_k(x) - \bar{y}_k^d(x))| |\bar{y}_k^d(x) - \bar{y}_k(x)| \quad (7.20)$$

The difference  $|\bar{y}_k^d(x) - \bar{y}_k(x)|$  can be made arbitrarily small by making the discretization of each component of  $\beta$  small. The rest of the proof follows the exact same steps as in the case of constant hypothesis class as shown above.

### 7.9.3 Appendix D

**Proof of Proposition 5.** For this proposition, we assume that the loss function is MSE and interpretive model belong to piecewise constant class. In this proposition, we need to show that the complexity of the Algorithm 5 and 6 is  $\mathcal{O}(|D|^3 K d)$ .

From Algorithm 5 we know that the main step that is executed inside the for loops is

$$V'(n+1, k) = \min_{n' \in \{1, \dots, n-1\}} \{V'(n', k-1) + G(n'+1, n)\}$$

Let us compute the complexity of the above step. Note that the terms inside the above expression depends on  $V'(n', k-1)$  and  $G(n'+1, n)$ .  $V'(n', k-1)$  is stored already from

the previous  $n$  iterations. The computation of  $G(n' + 1, n)$  takes  $\mathcal{O}(nd)$  steps at most if the loss is MSE. We need to compare  $n$  of these values. Therefore, the total time for this step is  $\mathcal{O}(n^2d)$  steps. For a fixed  $n$  the inner for loop will take  $\mathcal{O}(n^2Kd)$  steps.

We can bound the steps for the outer for loop as  $C \sum_{n=1}^{|D|} n^2 Kd$  steps, which grows as  $\mathcal{O}(|D|^3 Kd)$  steps. The complexity of Algorithm 6 is  $\mathcal{O}(K)$  (as there are  $K$  calls to the function  $\Phi$ ). ■

#### 7.9.4 Appendix E

**Proof of Proposition 6.** We already showed that the optimization problems in equations (7.8) and (7.11) are equivalent.

Next we will show that

$$\min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})} \hat{R}(\mathcal{M}, \mathcal{Z}; D) = \min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D)$$

We now state a property that is used to construct the optimal ordered partition that is as good as the optimal partition.

**Ordering Property:** Consider a partition  $\mathcal{Z}$  of the feature space. Consider any two regions of the partition say  $A$  and  $B$ . We refer to the sets of points in the dataset that belong to  $A$  as  $\tilde{A}$  and  $B$  as  $\tilde{B}$ . Define  $f(\tilde{A}) = \{f(a), \forall a \in \tilde{A}\}$ . The set of predictions for sets  $\tilde{A}$  and  $\tilde{B}$  are  $f(\tilde{A})$  and  $f(\tilde{B})$ . The sample means for the predictions at the points in  $\tilde{A}$  and  $\tilde{B}$  are  $\bar{M}(\tilde{A})$  and  $\bar{M}(\tilde{B})$  respectively. Without loss of generality assume that  $\bar{M}(\tilde{A}) < \bar{M}(\tilde{B})$ . The property states that for all the points in  $\tilde{A}$  their corresponding black-box predictions  $f(x) < \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$  and for all the points in  $\tilde{B}$  their corresponding black-box predictions  $f(x) > \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$ . If this property holds for every pair of regions in the partition, then it automatically implies that the partition is ordered.

**Idea.** We will show that if a partition does not satisfy the ordering property then it can always be modified to construct a partition that is ordered and is at least as good as the partition that we start with.

We start with the partition  $\mathcal{Z}$  (we are interested in partitions with atleast two regions in

them.) that is optimal. Suppose that  $\mathcal{Z}$  does not satisfy the ordering property.

If the ordering property is not satisfied, then there are two possibilities:

1. For some two regions  $A$  and  $B$  in the partition (and corresponding induced sets  $\tilde{A}$  and  $\tilde{B}$ ) there exists a point  $x_s \in \tilde{A}$  such that  $f(x_s) > \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$
2. For some two regions  $A$  and  $B$  in the partition (and corresponding induced sets  $\tilde{A}$  and  $\tilde{B}$ ) there exists a point  $x_s \in \tilde{B}$  such that  $f(x_s) < \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$

For the rest of the proof we will assume that the first case is true. The analysis for the second case would be similar as well. We will show that we can modify the partition  $\mathcal{Z}$  to  $\mathcal{Z}'$  in a simple way such that the MSE for  $\mathcal{Z}'$  is in fact less than or equal to the MSE of  $\mathcal{Z}$ .

We modify the set  $\tilde{B}$  by adding  $x_s$  to it from the region  $\tilde{A}$ . We call these new regions as  $\tilde{B}'$  and  $\tilde{A}'$  respectively. We express the difference between the losses before and after moving the  $x_s$  below. Let  $y_s = f(x_s)$ .

$$\begin{aligned} L_{diff} &= \sum_{y \in f(\tilde{A})} [y - \bar{M}(\tilde{A})]^2 + \sum_{y \in f(\tilde{B})} [y - \bar{M}(\tilde{B})]^2 - \sum_{y \in f(\tilde{A}')} [y - \bar{M}(\tilde{A}')]^2 - \sum_{y \in f(\tilde{B}')} [y - \bar{M}(\tilde{B}')]^2 \\ &= (|\tilde{A}| - 1) \bar{M}^2(\tilde{A}') + (|\tilde{B}| + 1) \bar{M}^2(\tilde{B}') - |\tilde{A}| \bar{M}^2(\tilde{A}) - |\tilde{B}| \bar{M}^2(\tilde{B}) \end{aligned} \tag{7.21}$$

Our objective is to show that  $L_{diff} \geq 0$ .

We express  $\bar{M}(\tilde{A}')$  and  $\bar{M}(\tilde{B}')$  in terms of  $\bar{M}(\tilde{A})$  and  $\bar{M}(\tilde{B})$  respectively as follows.

$$\begin{aligned} \bar{M}(\tilde{A}') &= \frac{\bar{M}(\tilde{A})|\tilde{A}| - y_s}{|\tilde{A}| - 1} \\ \bar{M}(\tilde{B}') &= \frac{\bar{M}(\tilde{B})|\tilde{B}| + y_s}{|\tilde{B}| + 1} \end{aligned} \tag{7.22}$$

$$\bar{M}(\tilde{A}')^2 = \left( \frac{\bar{M}(\tilde{A})|\tilde{A}| - y_s}{|\tilde{A}| - 1} \right)^2 = \frac{\bar{M}(\tilde{A})^2|\tilde{A}|^2 + y_s^2 - 2y_s\bar{M}(\tilde{A})}{(|\tilde{A}| - 1)^2} \tag{7.23}$$

$$\bar{M}(\tilde{B}')^2 = \left( \frac{\bar{M}(\tilde{B})|\tilde{B}| + y_s}{|\tilde{B}| + 1} \right)^2 = \frac{\bar{M}(\tilde{B})^2|\tilde{B}|^2 + y_s^2 + 2y_s\bar{M}(\tilde{B})}{(|\tilde{B}| + 1)^2} \quad (7.24)$$

We substitute (7.23) and (7.24) into (7.21) to obtain the following.

$$L_{diff} = \frac{\bar{M}^2(\tilde{A})|\tilde{A}| + y_s^2 - 2y_s\bar{M}(\tilde{A})|\tilde{A}|}{(|\tilde{A}| - 1)} + \frac{-\bar{M}^2(\tilde{B})|\tilde{B}| + y_s^2 + 2y_s\bar{M}(\tilde{B})|\tilde{B}|}{(|\tilde{B}| + 1)} \quad (7.25)$$

The expression in the above equation (7.25) is a quadratic function of  $y_s$ . We call it  $L_{diff}(y_s)$ . We want to analyze the behavior of the above function in the region  $y_s > \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$ . Our objective is to show that the above function is greater than zero when  $y_s > \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$ . We compute the gradient of the above function at  $y_s = \frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$  as (7.26).

$$\begin{aligned} \frac{dL_{diff}(y_s)}{dy_s} \Big|_{y_s=\frac{\bar{M}(\tilde{A})+\bar{M}(\tilde{B})}{2}} &= \left[ \frac{2y_s}{|\tilde{A}| - 1} + \frac{2y_s}{|\tilde{B}| + 1} - 2\bar{M}(\tilde{A})\frac{|\tilde{A}|}{|\tilde{A}| - 1} + 2\bar{M}(\tilde{B})\frac{|\tilde{B}|}{|\tilde{B}| + 1} \right]_{y_s=\frac{\bar{M}(\tilde{A})+\bar{M}(\tilde{B})}{2}} \\ &= \left[ 2y_s(|\tilde{A}| + |\tilde{B}|)\frac{1}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} - 2\bar{M}(\tilde{A})\frac{|\tilde{A}|}{|\tilde{A}| - 1} + 2\bar{M}(\tilde{B})\frac{|\tilde{B}|}{|\tilde{B}| + 1} \right]_{y_s=\frac{\bar{M}(\tilde{A})+\bar{M}(\tilde{B})}{2}} \\ &= \left[ \frac{2y_s(|\tilde{A}| + |\tilde{B}|) + 2|\tilde{A}||\tilde{B}|(\bar{M}(\tilde{B}) - \bar{M}(\tilde{A})) - 2\bar{M}(\tilde{A})|\tilde{A}| - 2\bar{M}(\tilde{B})|\tilde{B}|}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} \right]_{y_s=\frac{\bar{M}(\tilde{A})+\bar{M}(\tilde{B})}{2}} \\ &= \bar{M}(\tilde{A})(|\tilde{B}| - |\tilde{A}|) + \bar{M}(\tilde{B})(|\tilde{A}| - |\tilde{B}|) + 2|\tilde{A}||\tilde{B}|(\bar{M}(\tilde{B}) - \bar{M}(\tilde{A})) \\ &= (|\tilde{A}| - |\tilde{B}|)(\bar{M}(\tilde{B}) - \bar{M}(\tilde{A})) + 2|\tilde{A}||\tilde{B}|(\bar{M}(\tilde{B}) - \bar{M}(\tilde{A})) \\ &= (\bar{M}(\tilde{B}) - \bar{M}(\tilde{A}))(|\tilde{A}| - |\tilde{B}| + 2|\tilde{A}||\tilde{B}|) \\ &= (\bar{M}(\tilde{B}) - \bar{M}(\tilde{A}))(|\tilde{A}| + |\tilde{B}|(2|\tilde{A}| - 1)) \end{aligned} \quad (7.26)$$

Since  $\bar{M}(\tilde{B}) \geq \bar{M}(\tilde{A})$  and  $|\tilde{A}| \geq 1$  the expression in (7.26) is greater than zero. Note that

$$\frac{d^2L_{diff}(y_s)}{dy_s^2} = \frac{2}{|\tilde{A}| - 1} + \frac{2}{|\tilde{B}| + 1} \geq 0$$

Therefore, the gradient of the above expression in (7.26) will be greater than zero at all the points greater than  $\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}$ . Hence, we get

$$\min_{y_s \in [\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}, \infty)} L_{diff}(y_s) = L_{diff}\left(\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}\right)$$

Next, we compute  $L_{diff}\left(\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}\right)$  in (7.27) and we see that the expression is always greater than or equal to zero.

$$\begin{aligned}
L_{diff}\left(\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}\right) &= \\
&\left[ \frac{\bar{M}(\tilde{A})^2|\tilde{A}| + y_s^2 - 2y_s\bar{M}(\tilde{A})|\tilde{A}|}{(|\tilde{A}| - 1)} + \frac{-\bar{M}(\tilde{B})^2|\tilde{B}| + y_s^2 + 2y_s\bar{M}(\tilde{B})|\tilde{B}|}{(|\tilde{B}| + 1)} \right]_{y_s=\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}} \\
&= \frac{\bar{M}^2(\tilde{A})|\tilde{A}| + [\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}]^2 - [\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})]\bar{M}(\tilde{A})|\tilde{A}|}{(|\tilde{A}| - 1)} + \\
&\quad - \frac{-\bar{M}(\tilde{B})^2|\tilde{B}| + [\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}]^2 + [\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})]\bar{M}(\tilde{B})|\tilde{B}|}{(|\tilde{B}| + 1)} \\
&= \frac{[\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}]^2 - \bar{M}(\tilde{B})\bar{M}(\tilde{A})|\tilde{A}|}{(|\tilde{A}| - 1)} + \frac{[\frac{\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})}{2}]^2 + \bar{M}(\tilde{A})\bar{M}(\tilde{B})|\tilde{B}|}{(|\tilde{B}| + 1)} \\
&= \frac{[\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})]^2}{4} \frac{|\tilde{B}| + |\tilde{A}|}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} - \bar{M}(\tilde{A})\bar{M}(\tilde{B}) \frac{|\tilde{A}| + |\tilde{B}|}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} \\
&= \frac{|\tilde{B}| + |\tilde{A}|}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} \left[ \frac{[\bar{M}(\tilde{A}) + \bar{M}(\tilde{B})]^2}{4} - \bar{M}(\tilde{A})\bar{M}(\tilde{B}) \right] \\
&= \frac{|\tilde{B}| + |\tilde{A}|}{(|\tilde{A}| - 1)(|\tilde{B}| + 1)} [\bar{M}(\tilde{B}) - \bar{M}(\tilde{A})]^2 / 4
\end{aligned} \tag{7.27}$$

If  $\bar{M}(\tilde{B}) > \bar{M}(\tilde{A})$ , then this contradicts the optimality of the partition  $\mathcal{Z}$ . If  $\bar{M}(\tilde{B}) = \bar{M}(\tilde{A})$ , then the partition  $\mathcal{Z}'$  is as good as  $\mathcal{Z}$ . The partition  $\mathcal{Z}'$  may not be ordered. We can repeat the above argument starting with  $\mathcal{Z}'$  until we have an ordered partition that has at least the same loss as  $\mathcal{Z}$ . Note that  $\tilde{A}$  has to have at least two points for the setup to make sense. If  $\tilde{A}$  only had one point then shifting the point would reduce the number of regions in the partition. In the case when  $|\tilde{A}| = 1$ , we do not shift the point from  $\tilde{A}$  to  $\tilde{B}$  but instead we swap the point  $y_s$  with a point from  $\tilde{B}$  which has a lower value than  $y_s$ . The same conclusion follows for this case as well. ■

### 7.9.5 Appendix F

**Proof of Proposition 7.** We need to show that

$$\left| \min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})} \hat{R}(\mathcal{M}, \mathcal{Z}; D) - \min_{\mathcal{M} \in \mathcal{H}^K, \mathcal{Z} \in \mathcal{P}_K(\mathcal{X})^\dagger} \hat{R}(\mathcal{M}, \mathcal{Z}; D) \right| \leq \epsilon$$

We denote  $l(\|x - y\|_s)$  as  $L(x, y)$ . The optimal value of the constant cluster mapping only depends on the data points in that cluster/region in the partition and it is computed as follows. For cluster  $Z$  we write the local optimal value as  $y_Z^*$ .

**Dense Partitions:** We first show that it is sufficient to consider a certain type of partitions to guarantee approximate optimality, which we call dense partitions. The idea behind a dense partition is described as follows. As the dataset becomes large, each set in the induced partition should also become large. For ease of explanation, we will use the induced partitions on the dataset only.

Suppose the total number of data points is  $n = |D|$ .

**Definition.** Consider a partition  $\mathcal{Z} = \{Z_1, \dots, Z_K\}$ . The set of points that belong to  $Z_j$  are given as  $\tilde{Z}_j$  and let  $n_j = |\tilde{Z}_j|$ . We refer to  $\mathcal{Z}$  as dense if the data size grows to infinity, then the size of each induced region should also increase to infinity, i.e., as  $n \rightarrow \infty \implies n_j \rightarrow \infty, \forall j \in \{1, \dots, K\}$ .

Let  $\mathcal{P}_K^d(\mathcal{X})$  be the set of all the dense partitions of  $\mathcal{X}$ . We argue that it is sufficient to search in the space of dense partitions provided the dataset is large enough. Suppose that there is a partition  $\mathcal{Z} \in \mathcal{P}_K^d(\mathcal{X})^c$ , which is not dense. If a partition is not dense, then it can be argued that there exists a certain region  $Z_k$  such that  $n_k \leq N_k$ , where  $N_k$  is the upper bound on the size of  $Z_k$ . We construct a new partition from  $\mathcal{Z}$ . We transfer the points in  $Z_k$  to one of the remaining regions. The maximum change in the loss can be bounded by  $c \frac{N_k}{n}$  for some  $c > 0$ . If the data size is large enough, then the change in loss can be bounded less than  $\epsilon/K$ . We can repeat this argument for all the regions that have a bounded number of points. The final partition we get as a result will be a dense partition and its loss will be close to the original partition. Therefore, for the rest of the proof we restrict our attention to dense partitions.

**Ordering property for general loss function:** We now state a property that is very similar to the property (basically a generalization) we stated for MSE, used to construct the optimal ordered partition. Consider a partition  $\mathcal{Z}$ . Consider any two regions of the partition say  $A$  and  $B$  (with induced sets on the data given as  $\tilde{A}$  and  $\tilde{B}$ ) and the corresponding optimal predicted values assigned by the model  $M$  are  $y_{\tilde{A}}^*$  and  $y_{\tilde{B}}^*$  respectively. Without loss of generality assume that  $y_{\tilde{A}}^* < y_{\tilde{B}}^*$ . The property states that for all the points in  $\tilde{A}$  their corresponding black-box predictions  $f(x) < \frac{y_{\tilde{A}}^* + y_{\tilde{B}}^*}{2}$  and for all the points in  $\tilde{B}$  their corresponding black-box predictions  $f(x) > \frac{y_{\tilde{A}}^* + y_{\tilde{B}}^*}{2}$ . Note that if this property holds for every pair of regions in the partition, then it automatically implies that the partition is ordered.

We start with the partition  $\mathcal{Z}$  (we are interested in partitions with at least two regions in them) that is optimal and does not satisfy the ordering property.

There are two possibilities:

1. For some two regions  $\tilde{A}$  and  $\tilde{B}$  in the partition there exists a point  $x_s \in \tilde{A}$  such that  $f(x_s) > \frac{y_{\tilde{A}}^* + y_{\tilde{B}}^*}{2}$
2. For some two regions  $\tilde{A}$  and  $\tilde{B}$  in the partition there exists a point  $x_s \in \tilde{B}$  such that  $f(x_s) < \frac{y_{\tilde{A}}^* + y_{\tilde{B}}^*}{2}$

For the rest of the proof we will assume that the first case is true. The analysis for the second case would be similar as well.

**Idea.1** We will show that we can modify the partition  $\mathcal{Z}$  to  $\mathcal{Z}'$  in a simple way such that the loss for  $\mathcal{Z}'$  is in fact lower than the loss of  $\mathcal{Z}$ . We modify the region  $\tilde{B}$  of by adding  $x_s$  to it from the region  $\tilde{A}$ . We call these new regions as  $\tilde{B}'$  and  $\tilde{A}'$  respectively.

**Idea 2.** In this case since we deal with general loss functions the optimal value  $y_{\tilde{A}}^*$  does not have a closed form unlike the case of MSE. This makes it difficult to track the change in  $y_{\tilde{A}}^*$  when we construct the new regions  $\tilde{A}'$ . However, we can track the changes using influence functions [CW82] provided the number of data points is sufficiently large.

We express the difference between the losses before and after moving the  $x_s$ . Let  $y_s = f(x_s)$  Since  $y_s > \frac{y_{\tilde{A}}^* + y_{\tilde{B}}^*}{2}$  we get  $L(y_s, y_{\tilde{A}}^*) > L(y_s, y_{\tilde{B}}^*)$ .

We define the loss for the sets  $\tilde{A}$  and  $\tilde{B}$  as

$$L^{total}(\tilde{A}) = \sum_{y \in \tilde{A}} L(y, y_{\tilde{A}}^*)$$

$$L^{total}(\tilde{B}) = \sum_{y \in \tilde{B}} L(y, y_{\tilde{B}}^*)$$

We write the change in the loss function for the two sets  $A$  and  $B$  as follows.

$$L^{total}(\tilde{A}) - L^{total}(\tilde{A}') = \sum_{y \in \tilde{A}, y \neq y_s} L(y, y_{\tilde{A}}^*) - \sum_{y \in \tilde{A}, y \neq y_s} L(y, y_{\tilde{A}'}^*) + L(y_s, y_{\tilde{A}}^*) \quad (7.28)$$

$$L^{total}(\tilde{B}) - L^{total}(\tilde{B}') = \sum_{y \in \tilde{B}, y \neq y_s} L(y, y_{\tilde{B}}^*) - \sum_{y \in \tilde{B}, y \neq y_s} L(y, y_{\tilde{B}'}^*) - L(y_s, y_{\tilde{B}'}^*) \quad (7.29)$$

We track the change in  $y_{\tilde{A}}^*$  to  $y_{\tilde{A}'}^*$  and  $y_{\tilde{B}}^*$  to  $y_{\tilde{B}'}^*$  using influence functions [CW82].

We can express the difference  $y_{\tilde{A}'}^* - y_{\tilde{A}}^*$  using [CW82].

$$y_{\tilde{A}'}^* - y_{\tilde{A}}^* \approx \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{A}}^*} \right] \frac{1}{\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*}} \frac{1}{|\tilde{A}|}$$

$$L(y, y_{\tilde{A}}^*) - L(y, y_{\tilde{A}'}^*) \approx \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{A}}^*} (y_{\tilde{A}}^* - y_{\tilde{A}'}^*) \quad (7.30)$$

We substitute  $y_{\tilde{A}'}^* - y_{\tilde{A}}^*$  from above in (7.30).

$$L(y, y_{\tilde{A}}^*) - L(y, y_{\tilde{A}'}^*) \approx - \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{A}}^*} \right]^2 \frac{1}{\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*}} \frac{1}{|\tilde{A}|} \quad (7.31)$$

We track the change in  $y_{\tilde{B}}^*$  to  $y_{\tilde{B}'}^*$ . The final expressions for the change are given as follows.

$$L(y, y_{\tilde{A}}^*) - L(y, y_{\tilde{A}'}^*) \approx - \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{A}}^*} \right]^2 \frac{1}{\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*}} \frac{1}{|\tilde{A}|} \quad (7.32)$$

$$L(y, y_{\tilde{B}}^*) - L(y, y_{\tilde{B}'}^*) \approx \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{B}}^*} \right]^2 \frac{1}{\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*}} \frac{1}{|\tilde{B}|} \quad (7.33)$$

The function  $\frac{\partial L(y, c)}{\partial c}$  is continuous almost everywhere. The set  $\mathcal{Y}$  is an interval. Hence, the above function  $\frac{\partial L(y, c)}{\partial c}$  on the interval  $\mathcal{Y}$  is bounded.

Consider  $\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*}$ . We claim that if  $|\tilde{B}|$  is sufficiently large, then

$$\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*} = \Omega(|\tilde{B}|) \text{ with a high probability,}$$

i.e. there exists a  $k$  and  $n_0$  such that  $\forall |\tilde{B}| > n_0$ ,  $\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*} \geq k|\tilde{B}|$  with a high probability. We know that  $L$  is strictly convex. The term  $\frac{1}{|\tilde{B}|} \sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*}$  will take a fixed positive value in limit as  $\tilde{B}$  grows large (from strong law of large numbers). We can set  $k$  to be anything smaller than the limit of this term to establish the claim. Note that since we are using dense partitions it is safe to assume that as the data will grow large so will the size of the regions. Similarly  $\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*} = \Omega(|\tilde{A}|)$ .

Therefore, we can substitute the lower bounds on  $\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*}$  and  $\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*}$  in (7.32) and (7.33) to obtain

$$|L(y, y_{\tilde{A}}^*) - L(y, y_{\tilde{A}'}^*)| \leq \frac{L_{\tilde{A}}}{|\tilde{A}|^2} \quad (7.34)$$

$$|L(y, y_{\tilde{B}}^*) - L(y, y_{\tilde{B}'}^*)| \leq \frac{L_{\tilde{B}}}{|\tilde{B}|^2} \quad (7.35)$$

We add (7.28) and (7.29) to obtain  $L_{diff}$  given as follows. In the simplification below we use (7.32) and (7.33).

$$\begin{aligned}
L_{diff} &= L^{total}(\tilde{A}) - L^{total}(\tilde{A}') + L^{total}(\tilde{B}) - L^{total}(\tilde{B}') = \sum_{y \in \tilde{A}, y \neq y_s} L(y, y_{\tilde{A}}^*) - \sum_{y \in \tilde{A}, y \neq y_s} L(y, y_{\tilde{A}'}^*) \\
&+ L(y_s, y_{\tilde{A}}^*) + \sum_{y \in \tilde{B}, y \neq y_s} L(y, y_{\tilde{B}}^*) - \sum_{y \in \tilde{B}, y \neq y_s} L(y, y_{\tilde{B}'}^*) - L(y_s, y_{\tilde{B}'}^*) \\
&\approx \left[ L(y_s, y_{\tilde{A}}^*) - L(y_s, y_{\tilde{B}}^*) \right] - \sum_{y \in \tilde{A}, y \neq y_s} \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{A}}^*} \right]^2 \frac{1}{\sum_{y \in \tilde{A}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{A}}^*}} \frac{1}{|\tilde{A}|} \\
&+ \sum_{y \in \tilde{B}, y \neq y_s} \left[ \frac{\partial L(y, c)}{\partial c} \Big|_{c=y_{\tilde{B}}^*} \right]^2 \frac{1}{\sum_{y \in \tilde{B}} \frac{\partial^2 L(y, c)}{\partial c^2} \Big|_{c=y_{\tilde{B}}^*}} \frac{1}{|\tilde{B}|}
\end{aligned} \tag{7.36}$$

The first term in the above expression, i.e.,  $\left[ L(y_s, y_{\tilde{A}}^*) - L(y_s, y_{\tilde{B}}^*) \right] > 0$ . Therefore,  $\exists \epsilon' > 0$  such that

$$\left[ L(y_s, y_{\tilde{A}}^*) - L(y_s, y_{\tilde{B}}^*) \right] \geq \epsilon' \tag{7.37}$$

The rest of the terms in the above expression are bounded as well. We use the expressions in (7.34) and (7.35) to arrive at a lower bound on  $L_{diff}$  given as

$$L_{diff} \geq \left[ L(y_s, y_{\tilde{A}}^*) - L(y_s, y_{\tilde{B}}^*) \right] - \frac{L_{\tilde{A}}}{|\tilde{A}|} - \frac{L_B}{|\tilde{B}|} \tag{7.38}$$

Suppose  $\frac{L_{\tilde{A}}}{|\tilde{A}|} > \frac{L_{\tilde{B}}}{|\tilde{B}|}$  without loss of generality.

$$L_{diff} \geq \left[ L(y_s, y_{\tilde{A}}^*) - L(y_s, y_{\tilde{B}}^*) \right] - 2 \frac{L_{\tilde{A}}}{|\tilde{A}|} \tag{7.39}$$

Also, from (7.37) we have

$$L_{diff} \geq \epsilon' - 2 \frac{L_{\tilde{A}}}{|\tilde{A}|} \tag{7.40}$$

Suppose  $|\tilde{A}| \geq 4\frac{1}{\epsilon'} L_{\tilde{A}}$ . Then  $L_{diff} \geq \frac{\epsilon'}{2}$ . (Note that we are only considering dense partitions. If the data set is large enough the size of  $\tilde{A}$  will satisfy the required assumption.)

Therefore,  $L_{diff} > 0$ , which means that the loss improves by shifting the data points. This contradicts the optimality of the partition among the dense partitions. ■

### 7.9.6 Appendix G

---

**Algorithm 7** Approximate computation value and index functions

---

- 1: **Input:** Dataset  $D$ , Number of intervals  $H$  and number of regions to divide each interval  $W$ ,  $\delta$  is an integer, larger the value of  $\delta$  the higher the approximation factor
- 2: **Initialize:** Define  $V'(1, k) = 0, \forall k \in \{1, \dots, K\}$ .
- 3: For each  $x_i \in D, x_j \in D$  such that  $i \leq j$ , define  $D(i, j) = \{x : x \in D \text{ and } \|f(x_i)\| \leq \|f(x)\| \leq \|f(x_j)\|\}$
- 4:  $\{S_1^{ij}, \dots, S_W^{ij}\} = \text{Kmeans}(D(h_l, h_u))$
- 5:  $G(i, j) = \sum_u \min_{h \in \mathcal{H}} \sum_{x_r \in S_u^{ij}} l(|f(x_r) - h(x_r)|)$
- 6:  $M(S_u^{ij}) = \arg \min_{h \in \mathcal{H}} \sum_{x_r \in S_u^{ij}} l(|f(x_r) - h(x_r)|)$
- 7: **for**  $n \in \{2, \dots, |D|\}$  **do**
- 8:     **for**  $k \in \{1, \dots, K\}$  **do**
- 9:

$$V'(n, k) = \min_{n' \in \{1, \delta+1, 2\delta+1, \dots, \lfloor \frac{n-1}{\delta} \rfloor \delta\}} [V'(n', k-1) + G(n'+1, n)] \quad (7.41)$$

10:

$$\Phi(n, k) = \arg \min_{n' \in \{1, \delta+1, 2\delta+1, \dots, \lfloor \frac{n-1}{\delta} \rfloor \delta\}} [V'(n', k-1) + G(n'+1, n)] \quad (7.42)$$

11: **Output:** Value function  $V'$ , Index function  $\Phi$

---

# CHAPTER 8

## Optimization Based Approach to Estimating Kullback-Leibler Divergence

### 8.1 Introduction

Kullback-Leibler (KL) divergence is one of the fundamental quantities in statistics and machine learning. It is used to measure the distance between two probability distributions. Mutual information, which is another fundamental quantity, is a special case of KL divergence. It measures the information shared between two random variables and is equal to the KL divergence between the joint and product distributions of the two random variables. It is used in several applications such as feature selection [PLD05], clustering [RBN14], and representation learning [CDH16]. Estimation of KL divergence and mutual information is a challenging task and developing estimators for these quantities continue to be an active area of research.

Recently a method called Mutual Information Neural Estimation (MINE) [BBR18] has been proposed to estimate the KL divergence between two distributions. The key ideas in MINE are explained as follows:

- Use the Donkser-Varadhan (DV) [DV83] representation to express the KL divergence.
- Use a family of functions characterized by neural networks in the DV representation to build the estimator.

The authors in [BBR18] used MINE to estimate the mutual information and showed that their estimator is better than the estimators in the literature [KSG04] [Per08] in terms of the

bias in many cases. MINE is a general purpose estimator as it estimates the KL divergence and not just mutual information. However, the estimator constructed in [BBR18] using the main algorithm is not guaranteed to be consistent (explained later). In this work, we propose a new estimator of KL divergence to address this issue. We also rely on the Donsker-Varadhan representation to build our estimator. If we estimate the KL divergence using the DV representation, then we do not need to estimate the probability distributions directly unlike the standard estimators [KSG04]. Instead of searching in the space of neural network families (as in [BBR18]) we set the search space as a Reproducing Kernel Hilbert Space (RKHS) and hence we name the estimator as the Kernel KL divergence estimator (KKLE). We are able to show that the search in RKHS reduces to solving a convex learning problem. This enables us to prove that the estimator we derive is consistent.

In the experiments section, we compare the proposed KKLE with MINE estimator. We carry out simulations over large datasets to show that the performances of both MINE and KKLE are comparable. We also compare the two estimators for small datasets and we find that the KKLE estimator is better than the MINE estimator. We also provide insights to explain why KKLE is expected to perform well.

## 8.2 Problem Formulation and Approach

We first give a brief background. KL divergence is a quantity that is used to measure the distance between two probability distributions  $\mathbb{P}$  and  $\mathbb{Q}$ . It is defined as

$$\text{KL}(\mathbb{P} \parallel \mathbb{Q}) := \mathbb{E}_{\mathbb{P}}[\log \frac{d\mathbb{P}}{d\mathbb{Q}}]$$

where  $\frac{d\mathbb{P}}{d\mathbb{Q}}$  is the Radon-Nikodym derivative of  $\mathbb{P}$  with respect to  $\mathbb{Q}$ . . The Shannon entropy of a random variable is the amount of information contained in  $X$  and is defined as  $H(X) := \mathbb{E}_{\mathbb{P}_X}[-\log d\mathbb{P}_X]$ , where  $\mathbb{P}_X$  is the distribution of  $X$ . Mutual information between two random variables  $X, Y$  is defined as

$$I(X; Y) := H(X) - H(X \mid Y)$$

where  $H(X)$  is the Shannon entropy of  $X$ ,  $H(X | Y)$  is the Shannon entropy of  $X$  conditional on  $Y$ . Let the joint probability distribution of  $X$  and  $Y$  be  $\mathbb{P}_{XY}$  and the product of the marginal distributions be  $\mathbb{P}_X \otimes \mathbb{P}_Y$ . The mutual information between two random variables can also be expressed in terms of the KL divergence as follows.  $I(X; Y) = \text{KL}(\mathbb{P}_{XY} || \mathbb{P}_X \otimes \mathbb{P}_Y)$ , where  $\text{KL}$  is the KL divergence between the two input distributions. We describe the Donsker-Varadhan representation for KL divergence next.

### 8.2.1 The Donsker-Varadhan Representation

The Donsker Varadhan (DV) representation [DV83] for KL divergence between two distributions  $\mathbb{P}$  and  $\mathbb{Q}$  is given as follows. The sample space for the distributions  $\mathbb{P}$  and  $\mathbb{Q}$  is the same set  $\Omega$ . For simplicity, we assume that  $\Omega$  is a compact subset of  $\mathbb{R}$ . Suppose  $T$  is a mapping from the sample space  $\Omega$  to  $\mathbb{R}$ , i.e.,  $T : \Omega \rightarrow \mathbb{R}$ .

$$\text{KL}(\mathbb{P} || \mathbb{Q}) = \sup_{T \in \mathcal{M}} \left[ \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T]) \right] \quad (8.1)$$

where  $\mathcal{M}$  is the space of mappings where both the expectations  $\mathbb{E}_{\mathbb{P}}[T]$  and  $\log(\mathbb{E}_{\mathbb{Q}}[e^T])$  are finite. Recall that if  $\mathbb{P} = \mathbb{P}_{XY}$  and  $\mathbb{Q} = \mathbb{P}_X \otimes \mathbb{P}_Y$ , then we obtain the mutual information  $I(X; Y)$ . Since our work is closely related to MINE [BBR18] we explain the approach briefly in the next section.

### 8.2.2 MINE

We are given a set of parameters  $\Theta$  that define the family of neural networks. Each member  $\theta$  of the family characterizes a function  $T_\theta$  and the set of all the functions is defined as  $\mathcal{F} = \{T_\theta; \theta \in \Theta\}$ . The neural measure of KL divergence is defined as

$$\text{KL}_\Theta(\mathbb{P} || \mathbb{Q}) = \sup_{\theta \in \Theta} \left[ \mathbb{E}_{\mathbb{P}}[T_\theta] - \log(\mathbb{E}_{\mathbb{Q}}[e^{T_\theta}]) \right] \quad (8.2)$$

From (8.1) and (8.2), we can see that

$$\text{KL}(\mathbb{P} || \mathbb{Q}) \geq \text{KL}_\Theta(\mathbb{P} || \mathbb{Q})$$

Define  $\hat{\mathbb{P}}^n$  and  $\hat{\mathbb{Q}}^m$  as the empirical distribution of  $\mathbb{P}$  and  $\mathbb{Q}$  respectively with  $n$  and  $m$  i.i.d. samples given as  $\mathbf{X} = \{x_i\}_{i=1}^n$  and  $\mathbf{Y} = \{y_j\}_{j=1}^m$  respectively. Let  $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ . We write  $\mathbf{Z} = \{z_k, \forall k \in \{1, \dots, n+m\}\}$ , where  $z_k = x_k, \forall k \in \{1, \dots, n\}$  and  $z_{n+k} = y_k, \forall k \in \{1, \dots, m\}$ . The MINE estimator for KL divergence is given as

$$\hat{\text{KL}}_\Theta(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) = \sup_{\theta \in \Theta} \left[ \mathbb{E}_{\hat{\mathbb{P}}^n} [T_\theta] - \log \left( \mathbb{E}_{\hat{\mathbb{Q}}^m} [e^{T_\theta}] \right) \right] \quad (8.3)$$

### 8.2.2.1 Limitations of MINE

In [BBR18], it was shown that  $\hat{\text{KL}}_\Theta(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m)$  is a consistent estimator of the KL divergence. The algorithm in [BBR18] tries to maximize the loss function  $\mathbb{E}_{\hat{\mathbb{P}}^n} [T_\theta] - \log \left( \mathbb{E}_{\hat{\mathbb{Q}}^m} [e^{T_\theta}] \right)$  to get as close as possible to (8.3). Stochastic gradient descent (SGD) is used to search for the optimal neural network parameters  $\theta$  in  $\Theta$ . For the estimator in (8.3) to be consistent the family of neural networks has to consist of at least one hidden layer [BBR18] [Hor91]. As a result, the loss function that the algorithm tries to optimize is non-convex. Since the loss is non-convex it is not guaranteed to converge to the MINE estimator defined in equation (8.3). Also, since the loss function is non-convex the optimization can lead to poor local minima, which are worse than the other minima or have poor generalization properties.

### 8.2.3 KKLE: Kernel Based KL Divergence Estimation

In this section, we build an approach that overcomes the limitations that were highlighted in the previous section. Consider a RKHS  $\mathcal{H}$  over  $\mathbb{R}$  with a kernel  $k : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . We assume that the kernel is a continuously differentiable function. The norm of a function  $T$  in  $\mathcal{H}$  is given as  $\|T\|_{\mathcal{H}}^2 = \langle T, T \rangle_{\mathcal{H}}$ , where  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  is the inner product defined in the Hilbert Space. In [BBR18], it was assumed that the function  $T_\theta$  is bounded. We also limit our search over the space of bounded functions, i.e., we assume that the  $\|T\|_{\mathcal{H}} \leq M$ . This is a reasonable assumption to make because (8.1) assumes the two expectation terms are finite, which is only possible if  $T$  is bounded almost everywhere. We define the kernel measure of

KL divergence as follows

$$\mathsf{KL}_{\mathcal{H}}(\mathbb{P} \parallel \mathbb{Q}) = \sup_{T \in \mathcal{H}, \|T\|_{\mathcal{H}} \leq M} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T]) \quad (8.4)$$

From (8.4) and (8.1), we can also deduce that

$$\mathsf{KL}(\mathbb{P} \parallel \mathbb{Q}) \geq \mathsf{KL}_{\mathcal{H}}(\mathbb{P} \parallel \mathbb{Q})$$

We define the empirical estimator of the kernel measure below.

$$\hat{\mathsf{KL}}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) = \sup_{T \in \mathcal{H}, \|T\|_{\mathcal{H}} \leq M} \left[ \mathbb{E}_{\hat{\mathbb{P}}^n}[T] - \log(\mathbb{E}_{\hat{\mathbb{Q}}^m}[e^T]) \right] \quad (8.5)$$

We define a matrix  $\mathbf{K}$ , which we call the kernel matrix, such that for every  $z_i \in \mathbf{Z}$ ,  $z_j \in \mathbf{Z}$ ,  $\mathbf{K}[z_i, z_j] = k(z_i, z_j)$ . For the rest of the discussion, we assume that the maximum exists and hence, the supremum and maximum are interchangeable. Let

$$g(\boldsymbol{\alpha}) = \log\left(\frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\boldsymbol{\alpha}^t \mathbf{K}[y_j, :]}\right) - \frac{1}{n} \sum_{x_i \in \mathbf{X}} \boldsymbol{\alpha}^t \mathbf{K}[x_i, :]$$

where  $\boldsymbol{\alpha} \in \mathbb{R}^{m+n}$  and  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{m+n}]$ . In the next proposition, we show that we can compute  $\hat{\mathsf{KL}}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m)$  by minimizing  $g(\boldsymbol{\alpha})$ .

**Proposition 8** *For any  $\epsilon > 0$ ,  $\exists t > 0$  such that the optimal  $T$  that solves (8.5) is  $T^*(z) = \sum_{i=1}^{n+m} \alpha_i^* k(z_i, z)$ , where  $\boldsymbol{\alpha}^*$  is*

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \leq M^2} g(\boldsymbol{\alpha}) + \frac{1}{t} \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \quad (8.6)$$

and

$$|\hat{\mathsf{KL}}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) + g(\boldsymbol{\alpha}^*)| \leq \epsilon$$

**Proof.** We rewrite the objective in (8.5) as a penalized objective as follows.

$$\log(\mathbb{E}_{\hat{\mathbb{Q}}^m}[e^T]) - \mathbb{E}_{\hat{\mathbb{P}}^n}[T] + \frac{1}{t} \|T\|_{\mathcal{H}}^2 \quad (8.7)$$

Suppose that  $t$  is large enough, i.e.,  $t \geq M^2/\epsilon$ . Therefore, the penalty term is bounded by a small value  $\epsilon$ . In such a case, the negative of the penalized objective in (8.7) is very close to

the objective in (8.5). Therefore, solving the problem below should give an  $\epsilon$  approximate solution to (8.5).

$$\min_{T, \|T\|_{\mathcal{H}} \leq M} \log \left( \mathbb{E}_{\hat{\mathbb{Q}}^m} [e^T] \right) - \mathbb{E}_{\hat{\mathbb{P}}^n} [T] + \frac{1}{t} \|T\|_{\mathcal{H}}^2 \quad (8.8)$$

We use Representer Theorem (See [SS01]) to infer that the optimal  $T$  for (8.8) that achieves the minimum above can be written as a linear combination

$$T^*(.) = \sum_{i=1}^{n+m} \alpha_i k(z_i, .) \quad (8.9)$$

where  $z_i = x_i$ ,  $\forall i \in \{1, \dots, n\}$  and  $z_{n+j} = y_j$ ,  $\forall j \in \{1, \dots, m\}$ . We substitute the above expressions from (8.9) in (8.8) to obtain the following equivalent optimization problem.

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \leq M^2} \log \left( \frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\boldsymbol{\alpha}^t \mathbf{K}[y_j, :]} \right) - \frac{1}{n} \sum_{x_i \in \mathbf{X}} \boldsymbol{\alpha}^t \mathbf{K}[x_i, :] + \frac{1}{t} \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \quad (8.10)$$

Hence, (8.10) is equivalent to (8.8), which gives the  $\epsilon$  approximate optimal solution to (8.5). This completes the proof.  $\blacksquare$

In Proposition 8, we showed that , i.e.,  $\hat{\mathsf{KL}}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) \approx -g(\boldsymbol{\alpha}^*)$ . Next we discuss how to solve for  $\hat{\mathsf{KL}}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m)$  efficiently. We solve (8.6) using SGD. See Algorithm 8 for a detailed description.

**Proposition 9** • *The optimization problem in (8.6) is a convex optimization problem.*

- *Algorithm 8 converges to the optimal solution of (8.6).*

**Proof.** The first term in the objective in (8.6) is log of sum of exponentials, which is a convex function (See [BV04]). The second term in (8.6) is linear. Therefore, the objective in (8.6) is a convex function. The matrix  $\mathbf{K}$  is positive definite (See [SS01]). Hence, the function  $\boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha}$  is convex. Therefore, the set of  $\boldsymbol{\alpha}$  to be searched, i.e.,  $\boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \leq M^2$  is a convex set. This establishes that (8.6) is a convex optimization problem.

If the objective function (8.6) is Lipschitz continuous and convex and bounded, then the stochastic gradient descent based procedure would converge to the minimum (See Chapter

---

**Algorithm 8** KKLE algorithm to estimate KL divergence

---

**Input:**  $\mathbf{X} = \{x_i\}_{i=1}^n \sim \mathbb{P}$  and  $\mathbf{Y} = \{y_j\}_{j=1}^m \sim \mathbb{Q}$ ,  $\gamma$  (distance from minimum),  $\max_{\text{iter}}$  (maximum number of steps),  $\eta$  (step size),  $k$  (minibatch size)

**Output:** KL divergence estimate

- 1: **Initialization:** Initialize  $\boldsymbol{\alpha}$  randomly,  $n_{\text{iter}} = 0$ , Convergence = False
  - 2: **While** ( $n_{\text{iter}} \leq \max_{\text{iter}}$  and Convergence == False)
  - 3: **Minibatch sampling:** Sample  $k$  samples from  $\mathbf{X}$  and  $k$  samples from  $\mathbf{Y}$
  - 4:
$$\hat{\text{KL}}(\boldsymbol{\alpha})_p = -\log\left(\frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\boldsymbol{\alpha}^t \mathbf{K}[y_j, :]}\right) + \frac{1}{n} \sum_{x_i \in \mathbf{X}} \boldsymbol{\alpha}^t \mathbf{K}[x_i, :] + \frac{1}{t} \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha}$$
  - 5:  $\alpha = \alpha + \eta \nabla \hat{\text{KL}}(\boldsymbol{\alpha})_p$
  - 6:
$$\hat{\text{KL}}(\boldsymbol{\alpha})_c = -\log\left(\frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\boldsymbol{\alpha}^t \mathbf{K}[y_j, :]}\right) + \frac{1}{n} \sum_{x_i \in \mathbf{X}} \boldsymbol{\alpha}^t \mathbf{K}[x_i, :] + \frac{1}{t} \boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha}$$
  - 7: **If**  $|\hat{\text{KL}}(\boldsymbol{\alpha})_c - \hat{\text{KL}}(\boldsymbol{\alpha})_p| \leq \gamma$
  - 8: Convergence = True
  - 9:  $n_{\text{iter}} = n_{\text{iter}} + 1$  **return**  $\hat{\text{KL}}(\boldsymbol{\alpha})_c$
- 

14 in [SB14]). We want to show that  $g(\boldsymbol{\alpha}) = \log\left(\frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\boldsymbol{\alpha}^t \mathbf{K}[y_j, :]}\right) - \frac{1}{n} \sum_{x_i \in \mathbf{X}} \boldsymbol{\alpha}^t \mathbf{K}[x_i, :]$  is Lipschitz continuous in  $\boldsymbol{\alpha}$ . It is sufficient to show that the gradient of the function  $g$  w.r.t  $\alpha$  is bounded. Define a function

$$h(t) = g(x + t(y - x))$$

and  $h'(t) = dh(t)/dt$ . Observe that  $h(0) = g(x)$  and  $h(1) = g(y)$ . Using chain rule we can write  $h'(t) = \nabla_z g(z)^t|_{z=x+t(y-x)}(y - x)$

$$\begin{aligned} g(y) - g(x) &= \int_0^1 h'(t) dt \\ &= \int_0^1 \nabla_z g(z)^t|_{z=x+t(y-x)}(y - x) dt \leq \|\nabla_z g(z)\| \|y - x\| \end{aligned} \tag{8.11}$$

We write the partial derivative of  $g$  w.r.t. each component of  $\boldsymbol{\alpha}$  as follows  $\frac{\partial g(\boldsymbol{\alpha})}{\alpha_j} = \frac{\sum_{i=1}^{n+m} e^{\alpha_j \mathbf{K}[z_i, z_j]} \mathbf{K}[z_i, z_j]}{\sum_{i=1}^{n+m} e^{\alpha^t \mathbf{K}[z_i, :]}}$ . We want to derive a loose upper bound on  $\|\nabla g\|_1$ . To do that we first make the following observation about the matrix  $\mathbf{K}$ . We assumed that the samples  $x_i$  and  $y_j$

that are drawn from the distributions  $\mathbb{P}$  and  $\mathbb{Q}$  come from a set  $\Omega$ , which is a compact subset of  $\mathbb{R}$ . Since the kernel  $k$  is a continuously differentiable function and  $\Omega$  is a compact subset we can infer that all the elements in  $\mathbf{K}$  are bounded. For simplicity, we assume that  $K$  is bounded above by 1 and bounded below by zero. Since all the terms in  $\frac{\partial g(\alpha)}{\alpha_j}$  are positive we can say the following

$$\begin{aligned} \|\nabla g\|_1 &= \frac{\sum_{j=1}^{n+m} \sum_{i=1}^{n+m} e^{\alpha_j \mathbf{K}[z_i, z_j]} \mathbf{K}[z_i, z_j]}{\sum_{i=1}^n e^{\alpha^t \mathbf{K}[z_i, :]}} \leq \\ \frac{\sum_{j=1}^n \sum_{i=1}^n e^{\alpha_j \mathbf{K}[z_i, z_j]} \mathbf{K}[z_i, z_j]}{n} &\leq \frac{\sum_{j=1}^n \sum_{i=1}^n e^{\alpha_j}}{n} \leq \max_{\alpha, \alpha^t \mathbf{K} \alpha \leq M} \sum_{j=1}^n e^{\alpha_j} \end{aligned} \quad (8.12)$$

Since  $\sum_{j=1}^n e^{\alpha_j}$  is bounded above in the search space. Therefore, the maximum in (8.12) has to be finite. Since  $\|\nabla g\|_2 \leq \|\nabla g\|_1$ . Hence  $\|\nabla g\|_2$  is bounded above and from (8.11) we can see that the function  $g$  is Lipschitz continuous in  $\alpha$ . Lastly, it is easy to see that  $g$  itself is bounded because  $\mathbf{K}$  is bounded and  $\alpha$  also takes value in a compact set. We also need to show that the second term in (8.6) is also Lipschitz continuous. The gradient of the second term is  $2\mathbf{K}\alpha$ . Let us try to bound the norm of the gradient. Before that since we know that  $\mathbf{K}$  is positive definite and symmetric, we can write the eigendecomposition of  $\mathbf{K}$  as  $\mathbf{K} = U\Lambda U^t$ , where  $U$  is an orthonormal matrix comprised of the eigenvectors of  $\mathbf{K}$ ,  $\Lambda = \text{diag}[\lambda_1, \dots, \lambda_{m+n}]$  is the diagonal matrix of the set of eigenvalues  $\{\lambda_i\}_{i=1}^{m+n}$ .

$$\|\mathbf{K}\alpha\|^2 = \alpha^t \mathbf{K}^t \mathbf{K} \alpha = \alpha^t U^t \Lambda^2 U \alpha = v^t \Lambda^2 v \leq \sum_i \lambda_i^2 \|v\|^2 = \sum_i \lambda_i^2 \|\alpha\|^2 \quad (8.13)$$

In the last simplification on RHS in the above we use the following.  $v = U\alpha$  and  $\|U\alpha\| = \|\alpha\|$  ( $U$  is an orthonormal matrix).  $\alpha^t \mathbf{K} \alpha$  is bounded  $\implies \|\alpha\|$  is also bounded. Hence,  $\|\mathbf{K}\alpha\|^2$  is also bounded. We have now shown that the objective in (8.6) is Lipschitz continuous. From Corollary 14.2 in [SB14], we know that the procedure in Algorithm 8<sup>1</sup> converges to the minimum of the problem (8.6). ■

---

<sup>1</sup>For the proof we are assuming that we use the entire data in one minibatch and follow gradient descent.

### 8.2.4 Analyzing the Consistency of KKLE

**Definition 14** *Strong Consistency:* For all  $\eta > 0$ , if there exists a kernel  $k$  and an  $N$  such that  $\forall n \geq N, m \geq N$  such that  $|\hat{KL}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) - KL(\mathbb{P} \parallel \mathbb{Q})| \leq \eta$  then  $\hat{KL}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m)$  is a strongly consistent estimator of  $KL(\mathbb{P} \parallel \mathbb{Q})$

**Theorem 18**  $\hat{KL}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m)$  is a strongly consistent estimator of  $KL(\mathbb{P} \parallel \mathbb{Q})$

**Proof.** The proof of this theorem follows the same steps as the Proof in [BBR18]. Since we are in a setting where the consistency depends on the expressiveness of the Hilbert Space, which is different from the setting in [BBR18], we have to redo the proof for this case. We divide the proof into two parts.

For simplicity, we assume that the Hilbert space  $\mathcal{H}$  has a finite dimensional basis  $\Phi$ . Hence, every function in  $\mathcal{H}$  can be written as  $T(z) = \beta^t \Phi(z)$ . We substitute this form of function in (8.5) to obtain

$$\hat{KL}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) = - \min_{\beta, \|\beta\| \leq M} \left[ \log \left( \frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\beta^t \Phi(y_j)} \right) - \frac{1}{n} \sum_{x_i \in \mathbf{X}} \beta^t \Phi(x_i) \right] \quad (8.14)$$

Note that the assumption will not limit us from extending the proof to infinite basis (We can approximate an infinite radial basis function kernel with a finite radial basis [RR08]). Next we show that the estimator from (8.14) is a consistent estimator of (8.4).

We use the triangle inequality to arrive at the following.

$$\begin{aligned} |\hat{KL}_{\mathcal{H}}(\hat{\mathbb{P}}^n \parallel \hat{\mathbb{Q}}^m) - KL_{\mathcal{H}}(\mathbb{P} \parallel \mathbb{Q})| &\leq \max_{\beta, \|\beta\| \leq M} \left( \left| \frac{1}{n} \left[ \sum_{x_i \in \mathbf{X}} \beta^t \Phi(x_i) \right] - \mathbb{E}[\beta^t \Phi(x_i)] \right| \right) \\ &\quad + \max_{\beta, \|\beta\| \leq M} \left| \log \left( \frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\beta^t \Phi(y_j)} \right) - \log(\mathbb{E}[e^{\beta^t \Phi(y_j)}]) \right| \end{aligned} \quad (8.15)$$

$\Phi$  is a continuous function and since the outcomes are drawn from  $\Omega$ , a compact subset in  $\mathbb{R}$ ,  $\Phi$  is bounded.  $\beta^t \Phi$  is bounded over the set  $\|\beta\| \leq M$ . The space of parameters  $\beta$  is compact because the norm of  $\|\beta\|$  is bounded. These observations allow us to use [GG00] to show the following for a sufficiently large  $N$  and  $n \geq N$

$$\max_{\beta, \|\beta\| \leq M} \left( \left| \frac{1}{n} \left[ \sum_{x_i \in \mathbf{X}} \beta^t \Phi(x_i) \right] - \mathbb{E}[\beta^t \Phi(x_i)] \right| \right) \leq \eta/2$$

Similarly  $\log(\mathbb{E}[e^{\beta^t \Phi_i}])$  is also bounded in  $\|\beta\| \leq M$ .

Similarly, for a sufficiently large  $N$  and  $m \geq N$  we have

$$\max_{\beta, \|\beta\| \leq M} \left| \log\left(\frac{1}{m} \sum_{y_j \in \mathbf{Y}} e^{\beta^t \Phi(y_j)}\right) - \log(\mathbb{E}[e^{\beta^t \Phi(y_j)}]) \right| \leq \eta/2$$

The next question we are interested in is if there exists a finite basis that is good enough. We use radial basis functions (Gaussian radial basis in particular) with finite number of centers [WWZ12]. Suppose we use a weighted sum of the radial basis functions to learn the mutual information. In [Buh03] [PS91] [WWZ12], it is shown that finite radial basis functions can approximate arbitrary functions. We assume that the function that achieves optimal for (8.1) is smooth (This assumption is also made in [Hor91] and [BBR18]).

Let  $T^* = \log \frac{d\mathbb{P}}{d\mathbb{Q}}$ . By construction  $T^*$  satisfies

$\mathbb{E}_{\mathbb{P}}[T^*] = \text{KL}(\mathbb{P} \parallel \mathbb{Q})$  and  $\mathbb{E}_{\mathbb{Q}}[e^{T^*}] = 1$ . Suppose we allow for  $\eta$  tolerance on the error on the function we want to approximate. For a fixed  $\eta$ , we can derive a finite basis which can approximate any smooth function as shown in [PS91]. Suppose a finite radial basis function spans the RKHS and let  $T$  be the function that achieves the maximizer for (8.5). For a function  $T$  we can write the gap between the KL divergence and KL divergence achieved by  $T$  as follows.

$$\text{KL}(\mathbb{P} \parallel \mathbb{Q}) - \text{KL}_{\mathcal{H}}(\mathbb{P} \parallel \mathbb{Q}) = \mathbb{E}_{\mathbb{P}}[T^* - T] + \mathbb{E}_{\mathbb{Q}}[e^{T^*} - e^T]$$

We can select a large enough radial basis (Theorem 1 in [PS91]) such that

$$\mathbb{E}_{\mathbb{P}}[T^* - T] \leq \eta/2$$

$$\mathbb{E}_{\mathbb{Q}}[e^{T^*} - e^T] \leq \eta/2$$

Both the above conditions hold simultaneously because  $e^x$  is Lipschitz continuous and  $T$  is bounded  $\|T\|_{\mathcal{H}} \leq M$ . ■

We established that the proposed estimator is strongly consistent. In the next section, we analyze the complexity and convergence properties of KKLE.

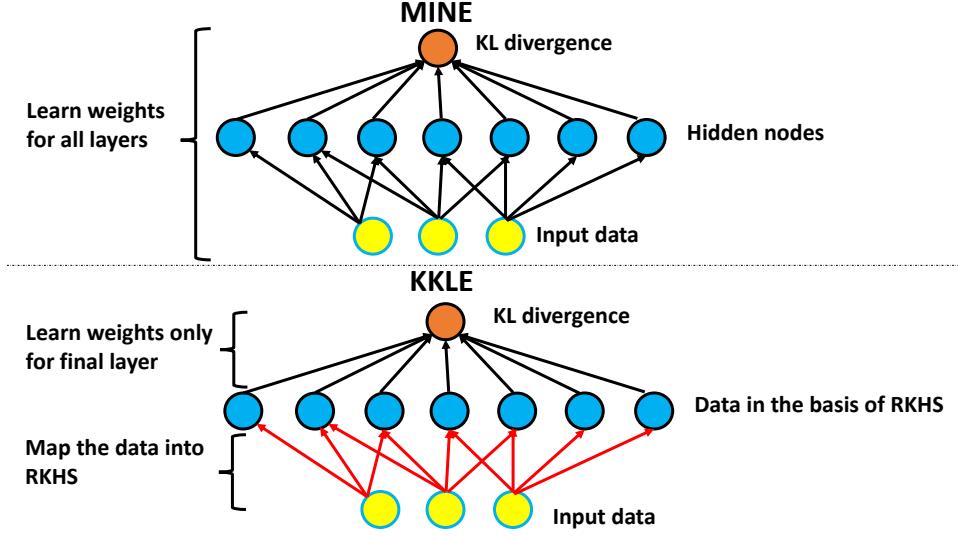


Figure 8.1: Compare KKLE vs MINE when using a finite basis for Hilbert Space.

### 8.2.5 Convergence and Complexity

The approach in Algorithm 8 optimizes the objective in (8.6). The number of steps before which the algorithm is guaranteed to converge is computed using [SB14]. The number steps grow as  $\mathcal{O}(\frac{\rho^2}{\epsilon^2})$ , where  $\rho$  is the Lipschitz constant for the loss function and  $\gamma$  is the tolerance in maximum distance from the minimum value of the loss (also defined in Algorithm 8).

The dimension of  $\boldsymbol{\alpha}$  vector is  $n + m$  and the dimension of the kernel matrix  $\mathbf{K}$  is  $m + n \times m + n$ . Computing and storing this matrix can be a problem if the data is too large. The time complexity of the algorithm is given as  $\mathcal{O}(\max_{\text{iter}}(m + n)^2)$ , where  $\max_{\text{iter}}$  is the maximum number of steps in the Algorithm 8 and  $(m + n)^2$  is the computational cost per step.

If the size of the data is large, then solving the above problem can be slow. We use [RR08] to improve the computational speed. In [RR08], the authors derive an approximation in terms of a lower  $d$  dimensional mapping  $\phi$  to approximately reproduce the kernel  $k$ . The complexity with this approximation drops to  $\mathcal{O}(\max_{\text{iter}}(m + n)d)$ . In the experiments section, we use this trick to improve the complexity.

Before going to experiments, we conclude this section with an illustrative comparison of KKLE with MINE. In Fig. 8.1, we compare the two estimators (KKLE and MINE) for the

case when RKHS is finite dimensional. For MINE all the layers of the neural network are trained to optimize the objective (8.3). For KKLE, the first layer projects the data into a higher dimensional basis of RKHS. The second and the final layer is trained to optimize (8.5).

## 8.3 Experiments

### 8.3.1 Comparisons

#### 8.3.1.1 Setup

We use the same setting as in [POO18] [BBR18]. We compare MINE estimator with KKLE estimator on the task of estimating mutual information, which as described earlier can also be represented in terms of the KL divergence. There are two random vectors  $\mathbf{X} \in \mathbb{R}^D$  and  $\mathbf{Y} \in \mathbb{R}^D$ , where  $X_k$  and  $Y_k$  are the  $k^{th}$  components of  $\mathbf{X}$  and  $\mathbf{Y}$  respectively.  $(X_k, Y_k)$  is drawn from a 2-dimensional Gaussian distribution with 0 mean and  $\rho$  correlation. The true mutual information in this case can be analytically computed and is given as  $-\frac{D}{2} \log(1 - \rho^2)$ . We are given a dataset with  $N$  i.i.d. samples from the distribution of  $(\mathbf{X}, \mathbf{Y})$ . In the next section, we compare the performance of the proposed KKLE estimator with MINE estimator in terms of the following metrics: Bias of the estimator, root mean squared error in the estimation (RMSE), variance in the estimator values. All the simulations were done on a 2.2GHz Intel Core i7 processor, with 16 GB memory using Tensorflow in Python. We use [RR08] to map the features and reduce the computational costs. The comparisons are done for two scenarios, when the dataset is very large, and when the dataset is small.

#### 8.3.1.2 Comparisons for large data

In this section, our goal is to compare the two estimators for a sufficiently large dataset ( $N = 10^5$ ) to show both the estimators are consistent. We sample  $N = 10^5$   $(\mathbf{X}, \mathbf{Y})$  from the distribution described above for  $D = 1$  and  $D = 5$ . We compare the bias, RMSE, and variance of the proposed KKLE estimator with the MINE estimator. The minibatch size

Table 8.1: KKLE vs MINE estimator for large data.

Estimator	D	Bias	RMSE	Variance	Correlation	Mutual Information
MINE	1	-0.009442	0.011378	0.000040	0.2	0.020411
MINE	1	-0.025266	0.030608	0.000299	0.5	0.143841
MINE	1	-0.060696	0.075414	0.002003	0.9	0.830366
KKLE	1	-0.009221	0.010990	0.000036	0.2	0.020411
KKLE	1	-0.025688	0.030982	0.000300	0.5	0.143841
KKLE	1	-0.065784	0.079743	0.002031	0.9	0.830366
MINE	5	-0.020874	0.024841	0.000181	0.2	0.102055
MINE	5	-0.072369	0.09106	0.003055	0.5	0.719205
MINE	5	-0.415350	0.758026	0.402088	0.9	4.151828
KKLE	5	-0.006116	0.038716	0.00146	0.2	0.102055
KKLE	5	-0.046382	0.116801	0.011491	0.5	0.719205
KKLE	5	-0.622219	0.979745	0.572215	0.9	4.151828

for the gradient descent is 5000. In each step a minibatch is sampled and a gradient step is taken. The total number of steps is 1000. In Table 8.1, we provide the comparisons for  $D = 1$  and  $D = 5$ . The results in the Table 8.1 are averaged over 100 trials. We observe that the performance of both the estimators are similar. Note that both the estimators degrade in the setting when dimensionality of the data becomes large and the variables are very correlated.

### 8.3.1.3 Comparison for small data

In this section, our goal is to compare the two estimators for a small dataset ( $N = 100$ ). We compare the bias, RMSE, and the variance of the KKLE estimator with the MINE estimator. Since the size of the data is small using minibatches did not help. Hence, we use the whole data and run the simulation for 100 gradient steps. We average the results for 100 trials

Table 8.2: KKLE vs MINE estimator for small data.

Estimator	Bias	RMSE	Variance	Correlation	Mutual Information
MINE	0.0939	0.1044	0.0021	0.2	0.0204
MINE	0.0681	0.1128	0.0081	0.5	0.1438
MINE	-0.2910	0.5123	0.1777	0.9	0.8303
KKLE	0.04999	0.0733	0.000288	0.2	0.0204
KKLE	0.06152	0.1254	0.01195	0.5	0.1438
KKLE	0.00855	0.1833	0.03357	0.9	0.8303

and report the comparisons in Table 8.2. We compare the estimators for  $D = 1$  scenario. We find that the KKLE estimator has a much lower bias, variance, and RMSE value. For  $D = 5$  scenario both the estimators are not reliable for the small dataset setting. Hence, the comparisons in this setting did not provide any insights and are not reported.

### 8.3.2 Explaining KKLE’s performance

We conclude that for smaller datasets and smaller dimensions the KKLE estimator performs better than the MINE estimator. When the datasets are very large both MINE and KKLE estimator perform well.

- The loss surface for MINE is non-convex in the parameters and thus different trials lead to different minima being achieved thus leading to a higher variance than KKLE, which searches over a convex loss surface.
- Hypothetically assume that the search space for KKLE is the same as MINE. In such a case, the optimizer for KKLE is likely to have a lower bias as it will always find the best minima, which is not true for MINE.

### 8.3.3 Application to Metrics for Fairness

There are many applications for mutual information. In this section, we propose another application that can directly benefit from the proposed estimator. Machine learning methods are used in many daily life applications. In many of these applications such as deciding whether to give a loan, hiring decisions, it is very important that the algorithm be fair. There are many definitions of fairness that have been proposed in the literature [SHG18]. We discuss the three most commonly used definitions of fairness here.

- **Demographic Parity.** A predictor is said to satisfy demographic parity if the predictor is independent of the protected attribute (for instance, race, gender, etc.).
- **Equality of Odds.** A predictor satisfies equality of odds if the predictor and the protected attribute are independent conditional on the outcomes.
- **Equality of Opportunity** A predictor satisfies equality of opportunity with respect to a certain class if the predictor and the protected attribute are independent conditional on the class.

These definitions provide a condition to measure fairness. These conditions serve as a hard constraint and may not be satisfied by any algorithm. Hence, it is important to provide metrics that measure the extent to which these conditions are satisfied. Current works [BDH18] mainly implement these metrics for fairness when the protected attribute is a categorical variable. Extending these metrics to settings when the protected attribute is continuous (for instance, income level, etc.) is not obvious (See the future works mentioned in [DOB18]).

We propose to express these fairness criteria in terms of mutual information. Expressing it in terms of mutual information has two advantages: a) We can understand the extent to which the criterion is satisfied as the new definition won't be a mere hard constraint, and b) Dealing with protected attributes that are continuous (for e.g., income level) becomes more natural.

We give the mathematical formulation next. Suppose that the predictor random variable is given as  $Y^p$  (for instance, the prediction that the individual would default on the loan), the ground truth is  $Y$  (for instance, if the person actually defaults on the loan), and the protected attribute is given as  $A$  (for instance, race, income level etc.).

- **Demographic Parity**  $Y^p \perp A \Leftrightarrow I(Y^p; A) = 0$
- **Equality of Odds**  $Y^p \perp A | Y \Leftrightarrow I(Y^p; A | Y) = 0$
- **Equality of Opportunity**  $Y^p \perp A | Y = 1 \Leftrightarrow I(Y^p; A | Y = 1) = 0$

Therefore, for each of the above definitions, we require the appropriate value of mutual information to be low. Hence, we can compare the extent of fairness for different machine learning models in terms of the mutual information estimate. In each of the above definitions, we are only required to estimate the mutual information between two random variables, which is good as we know that mutual information estimation is reliable in lower dimensions. It would be interesting to investigate mutual information based fairness constraints. Further investigation of mutual information based metrics for fairness in machine learning is an interesting future work.

## 8.4 Conclusion

We propose a new estimator for KL divergence based on kernel machines. We prove that the proposed estimator is consistent. Empirically, we find that the proposed estimator can be more reliable than the existing estimator MINE in different settings. We also provide insights into when KKLE is expected to do better than MINE.

# CHAPTER 9

## Conclusions and Future Work

In this dissertation, we developed approximate optimization methods for intractable optimization problems that have provable performance guarantees achievable with reasonable computational resources. We focused on two different areas where we often find such problems: a) resource allocation, and b) machine learning.

In the first part of this dissertation, we developed optimization methods for resource allocation problems. In Chapters 3 and 4, we developed methods for multi-agent resource sharing. The methods that we presented were applied to the problem of interference management in wireless networks and were shown to help tremendously in comparison to the state-of-the-art methods. Moreover, we proved that our distributed approach achieves a constant approximation ratio w.r.t to the best possible solution computed in a centralized manner. These methods can also be applied to many other resource allocation problems such as task scheduling. These provable efficiency guarantees also extend to more general cases. It would be interesting to explore the application of our framework to other resource allocation problems. In our current work, we only provided provable guarantees for static environments with a fixed number of users and fixed environment conditions. It would be useful to extend these results to more dynamic scenarios with changing users and changing environments.

Next, we studied a particular type of resource allocation problem with strategic agents. We studied the problem of matching with strategic agents. We developed a dynamic matching mechanism that allows the two sides to be matched for instance, the clients and workers to interact and learn about each other and then arrive at final matching. The proposed mechanism has several nice properties: the equilibrium strategy for the workers and clients

is simple, coalitionally stable and guarantees truthful revelation. The mechanism under certain settings is also guaranteed to lead to the maximum possible revenue possible among all the mechanisms.

In Chapter 6, we studied the problem of screening. The problem of screening can be abstracted as follows. An agent has limited resources to monitor a stochastic process. The agent’s objective is to use the resources to its avail in the best possible manner to best track the evolution of the stochastic process. We proposed a general framework to solve the above problem and applied it to the problem of breast cancer screening to establish its utility. We provided performance guarantees that are achievable in polynomial-time. The framework can be potentially applied to sensor scheduling and stopping time problems. It would be interesting to adapt and apply this framework to other applications in the future.

In the second part of our dissertation, we turned our attention to developing optimization methods for machine learning applications. In Chapter 7, we developed a method to interpret “black-box” models. Our method can be used to construct piecewise local-linear approximations of machine learning models. By constructing such approximations it becomes easier for expert auditing of the model as linear models are easier to understand. We established the utility of our approach through various experiments on real datasets. We applied our approach to regression problems on datasets with a moderate number of dimensions. Extending the proposed approach to high dimensional datasets and applying it on image datasets is a very interesting future work. We also applied the proposed approach to the problem of clustering. We gave a first proof that the proposed algorithm leads to a polynomial time solution to the problem of clustering one-dimensional data.

At the end in Chapter 8, we developed an optimization-based approach to estimate the KL divergence. The approach is inspired from recent methods that use optimization based on neural networks to estimate the KL divergence. We showed the utility of our proposed approach and established that it can be more useful than the recently proposed approach in certain scenarios. We proved that the proposed estimator is consistent unlike the recently proposed estimator.

## REFERENCES

- [AAE10] Oguzhan Alagoz, Turgay Ayer, and Fatih Safa Erenay. “Operations research models for cancer screening.” *Wiley encyclopedia of operations research and management science*, 2010.
- [AAS11] Vaneet Aggarwal, Amir Salman Avestimehr, and Ashutosh Sabharwal. “On achieving local view capacity via maximal independent graph scheduling.” *IEEE Transactions on Information Theory*, **57**(5):2711–2729, 2011.
- [AAS12] Turgay Ayer, Oguzhan Alagoz, and Natasha K Stout. “OR Forum—A POMDP approach to personalize mammography screening decisions.” *Operations Research*, **60**(5):1019–1034, 2012.
- [ACD12] Jeffrey G Andrews, Holger Claussen, Mischa Dohler, Sundeep Rangan, and Mark C Reed. “Femtocells: Past, present, and future.” *IEEE Journal on Selected Areas in Communications*, **30**(3):497–508, 2012.
- [Ahu19] Kartik Ahuja. “Estimating Kullback-Leibler Divergence Using Kernel Machines.” *arXiv preprint arXiv:1905.00586*, 2019.
- [AJK14] Nick Arnosti, Ramesh Johari, and Yash Kanoria. “Managing congestion in decentralized matching markets.” In *Proceedings of the fifteenth ACM conference on Economics and computation*, pp. 451–451. ACM, 2014.
- [AK08] Kellie J Archer and Ryan V Kimes. “Empirical characterization of random forest variable importance measures.” *Computational Statistics & Data Analysis*, **52**(4):2249–2260, 2008.
- [AKL17] Itai Ashlagi, Yash Kanoria, and Jacob D Leshno. “Unbalanced random matching markets: The stark effect of competition.” *Journal of Political Economy*, **125**(1):69–98, 2017.
- [AS16] Kartik Ahuja and Mihaela van der Schaar. “Dynamic matching and allocation of tasks.” *arXiv preprint arXiv:1602.02439*, 2016.
- [AV16] Ahmed M Alaa and Mihaela Van Der Schaar. “Balancing suspense and surprise: Timely decision making with endogenous information acquisition.” In *Advances in Neural Information Processing Systems*, pp. 2910–2918, 2016.
- [AXS15a] Kartik Ahuja, Yuanzhang Xiao, and Mihaela van der Schaar. “Distributed interference management policies for heterogeneous small cell networks.” *IEEE Journal on Selected Areas in Communications*, **33**(6):1112–1126, 2015.
- [AXS15b] Kartik Ahuja, Yuanzhang Xiao, and Mihaela van der Schaar. “Efficient interference management policies for femtocell networks.” *IEEE Transactions on Wireless Communications*, **14**(9):4879–4893, 2015.

- [AZS17] Kartik Ahuja, William Zame, and Mihaela van der Schaar. “Dpscreen: Dynamic personalized screening.” In *Advances in Neural Information Processing Systems*, pp. 1321–1332, 2017.
- [AZS18] Kartik Ahuja, William R Zame, and Mihaela van der Schaar. “Optimal Piecewise Local-Linear Approximations.” *arXiv preprint arXiv:1806.10270*, 2018.
- [BBB95] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.
- [BBM15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.” *PloS one*, **10**(7):e0130140, 2015.
- [BBR18] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. “Mine: mutual information neural estimation.” *arXiv preprint arXiv:1801.04062*, 2018.
- [BBS06] Gurashish Brar, Douglas M Blough, and Paolo Santi. “Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks.” In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 2–13. ACM, 2006.
- [BDH18] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. “AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias.” *arXiv preprint arXiv:1810.01943*, 2018.
- [Bec73] Gary S Becker. “A theory of marriage: Part I.” *Journal of Political economy*, **81**(4):813–846, 1973.
- [BKB17] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. “Interpreting blackbox models via model extraction.” *arXiv preprint arXiv:1705.08504*, 2017.
- [BP01] Nancy Baxter, Canadian Task Force on Preventive Health Care, et al. “Preventive health care, 2001 update: Should women be routinely taught breast self-examination to screen for breast cancer?” *Canadian Medical Association Journal*, **164**(13):1837–1846, 2001.
- [PPB13a] Mehdi Bennis, Samir M Perlaza, Pol Blasco, Zhu Han, and H Vincent Poor. “Self-Organization in Small Cell Networks: A Reinforcement Learning Approach.” *IEEE Transactions on Wireless Communications*, **12**(7):3202–3212, 2013.
- [PPB13b] Mehdi Bennis, Samir Medina Perlaza, Pol Blasco, Zhu Han, and H Vincent Poor. “Self-organization in small cell networks: A reinforcement learning approach.” *IEEE Transactions on Wireless Communications*, **12**(7):3202–3212, 2013.

- [BPC11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers.” *Foundations and Trends® in Machine learning*, **3**(1):1–122, 2011.
- [BT89] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.
- [Buh03] Martin D Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.
- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [CAM09] Vikram Chandrasekhar, Jeffrey G Andrews, Tarik Muharemovic, Zukang Shen, and Alan Gatherer. “Power control in two-tier femtocell networks.” *IEEE Transactions on Wireless Communications*, **8**(8):4316–4328, 2009.
- [CDH16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets.” In *Advances in neural information processing systems*, pp. 2172–2180, 2016.
- [CHH12] Chan-Byoung Chae, Insoo Hwang, Robert W Heath, and Vahid Tarokh. “Interference aware-coordinated beamforming in a multi-cell system.” *IEEE Transactions on Wireless Communications*, **11**(10):3692–3703, 2012.
- [CHH13] Marcia Irene Canto, Femme Harinck, Ralph H Hruban, George Johan Offerhaus, Jan-Werner Poley, Ihab Kamel, Yung Nio, Richard S Schulick, Claudio Bassi, Irma Kluijt, et al. “International Cancer of the Pancreas Screening (CAPS) Consortium summit on the management of patients with increased risk for familial pancreatic cancer.” *Gut*, **62**(3):339–347, 2013.
- [CLG15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission.” In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730. ACM, 2015.
- [CMK08] David Choi, Pooya Monajemi, Shinjae Kang, and John Villasenor. “Dealing with loud neighbors: The benefits and tradeoffs of adaptive femtocell access.” In *IEEE Global Telecommunications Conference*, pp. 1–5. IEEE, 2008.
- [Cox92] David R Cox. *Regression models and life-tables. Breakthroughs in statistics*. Springer New York, 1992.
- [Cro01] Martin J Crowder. *Classical competing risks*. Chapman and Hall/CRC, 2001.

- [CS89] Israel Cidon and Moshe Sidi. “Distributed assignment algorithms for multihop packet radio networks.” *IEEE Transactions on Computers*, **38**(10):1353–1361, 1989.
- [CSW18] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. “Learning to explain: An information-theoretic perspective on model interpretation.” *arXiv preprint arXiv:1802.07814*, 2018.
- [CTP07] Mung Chiang, Chee Wei Tan, Daniel P Palomar, Daniel O’Neill, and David Julian. “Power control by geometric programming.” *IEEE Transactions on Wireless Communications*, **6**(7):2640–2651, 2007.
- [CTW03] Dean Corbae, Ted Temzelides, and Randall Wright. “Directed matching and monetary exchange.” *Econometrica*, **71**(3):731–756, 2003.
- [CW82] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.
- [DCK11] Timothy J Daskivich, Karim Chamie, Lorna Kwan, Jessica Labo, Roland Palvolgyi, Atreya Dash, Sheldon Greenfield, and Mark S Litwin. “Overtreatment of men with low-risk prostate cancer and significant comorbidity.” *Cancer*, **117**(10):2058–2066, 2011.
- [DL05] Ettore Damiano and Ricky Lam. “Stability in dynamic matching markets.” *Games and Economic Behavior*, **52**(1):34–53, 2005.
- [DNG15] Pankaj Dayama, Balakrishnan Narayanaswamy, Dinesh Garg, and Y Narahari. “Truthful interval cover mechanisms for crowdsourcing applications.” In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1091–1099. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [DOB18] Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. “Empirical Risk Minimization under Fairness Constraints.” *arXiv preprint arXiv:1802.08626*, 2018.
- [Dov14] Laura Doval. “A theory of stability in dynamic matching markets.” Technical report, working paper, 2014.
- [DV83] Monroe D Donsker and SR Srinivasa Varadhan. “Asymptotic evaluation of certain Markov process expectations for large time. IV.” *Communications on Pure and Applied Mathematics*, **36**(2):183–212, 1983.
- [EAS14] Fatih Safa Erenay, Oguzhan Alagoz, and Adnan Said. “Optimizing colonoscopy screening for colorectal cancer prevention and surveillance.” *Manufacturing & Service Operations Management*, **16**(3):381–400, 2014.
- [EPT07] Raul Etkin, Abhay Parekh, and David Tse. “Spectrum sharing for unlicensed bands.” *IEEE Journal on Selected Areas in Communications*, **25**(3):517–528, 2007.

- [Erc13] Kayhan Erciyes. *Distributed Graph Algorithms for Computer Networks*. Springer, 2013.
- [ET90] Anthony Ephremides and Thuan V Truong. “Scheduling broadcasts in multihop radio networks.” *IEEE Transactions on Communications*, **38**(4):456–460, 1990.
- [FCF00] A Lindsay Frazier, Graham A Colditz, Charles S Fuchs, and Karen M Kuntz. “Cost-effectiveness of screening for colorectal cancer in the general population.” *Jama*, **284**(15):1954–1961, 2000.
- [Fis15] James CD Fisher. “Matching with continuous bidirectional investment.” *University of Technology Sydney Working Paper*, 2015.
- [FIT16] Daniel Fragiadakis, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo. “Strategyproof matching with minimum quotas.” *ACM Transactions on Economics and Computation*, **4**(1):6, 2016.
- [FM93] Gerard J Foschini and Zoran Miljanic. “A simple distributed autonomous power control algorithm and its convergence.” *IEEE Transactions on Vehicular Technology*, **42**(4):641–646, 1993.
- [GBB89] Mitchell H Gail, Louise A Brinton, David P Byar, Donald K Corle, Sylvan B Green, Catherine Schairer, and John J Mulvihill. “Projecting individualized probabilities of developing breast cancer for white females who are being examined annually.” *JNCI: Journal of the National Cancer Institute*, **81**(24):1879–1886, 1989.
- [GDD03] Caterina Guiot, Piero Giorgio Degiorgis, Pier Paolo Delsanto, Pietro Gabriele, and Thomas S Deisboeck. “Does tumor growth follow a “universal law”?” *Journal of theoretical biology*, **225**(2):147–151, 2003.
- [GF16] Bryce Goodman and Seth Flaxman. “European Union regulations on algorithmic decision-making and a right to explanation.” *arXiv preprint arXiv:1606.08813*, 2016.
- [GG00] Sara A van de Geer and Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.
- [GI10] Lorenza Giupponi and Christian Ibáñez. “Distributed interference control in OFDMA-based femtocells.” In *IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pp. 1201–1206. IEEE, 2010.
- [GKG07] David Gesbert, Saad Ghazanfar Kiani, Anders Gjendemsjø, et al. “Adaptation, coordination, and distributed resource allocation in interference-limited wireless networks.” *Proceedings of the IEEE*, **95**(12):2393–2409, 2007.
- [GMR12] Amitabha Ghosh, Nitin Mangalvedhe, Rapeepat Ratasuk, Bishwarup Mondal, Mark Cudak, Eugene Visotsky, Timothy A Thomas, Jeffrey G Andrews, Ping Xia, Han Shin Jo, et al. “Heterogeneous cellular networks: From theory to practice.” *IEEE Communications Magazine*, **50**(6):54–64, 2012.

- [Gol05] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [GS62] David Gale and Lloyd S Shapley. “College admissions and the stability of marriage.” *American mathematical monthly*, pp. 9–15, 1962.
- [Hal80] William K Hale. “Frequency assignment: Theory and applications.” *Proceedings of the IEEE*, **68**(12):1497–1514, 1980.
- [HBH06] Jianwei Huang, Randall A Berry, and Michael L Honig. “Distributed interference compensation for wireless networks.” *IEEE Journal on Selected Areas in Communications*, **24**(5):1074–1084, 2006.
- [HLN13] Ekram Hossain, Long Bao Le, and Dusit Niyato. “Self-Organizing Small Cell Networks.” In *Radio Resource Management in Multi-Tier Cellular Wireless Networks*. John Wiley Sons, Inc., Hoboken, NJ, USA, 2013.
- [Hol99] Bengt Holmström. “Managerial incentive problems: A dynamic perspective.” *The Review of Economic Studies*, **66**(1):169–182, 1999.
- [Hop12] Ed Hopkins. “Job market signaling of relative position, or Becker married to Spence.” *Journal of the European Economic Association*, **10**(2):290–322, 2012.
- [Hor91] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks.” *Neural networks*, **4**(2):251–257, 1991.
- [HS95] Mark L Huson and Arunabha Sen. “Broadcast scheduling algorithms for radio networks.” In *IEEE Military Communications Conference*, volume 2, pp. 647–651. IEEE, 1995.
- [HWK10] Kwanghun Han, Seungmin Woo, Duho Kang, and Sunghyun Choi. “Automatic neighboring BS list generation scheme for femtocell network.” In *Second International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 251–255. IEEE, 2010.
- [HYC09] Eun Jin Hong, Se Young Yun, and Dong-Ho Cho. “Decentralized power control scheme in femtocell networks: A game theoretic approach.” In *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 415–419, Sept 2009.
- [HZV12] Chien-Ju Ho, Yu Zhang, Jennifer Wortman Vaughan, and Mihaela Van Der Schaar. “Towards social norm design for crowdsourcing markets.” In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [IM15] Nicole Immorlica and Mohammad Mahdian. “Incentives in large random two-sided markets.” *ACM Transactions on Economics and Computation*, **3**(3):14, 2015.
- [JMM09] Han-Shin Jo, Cheol Mun, June Moon, and Jong-Gwan Yook. “Interference mitigation using uplink power control for two-tier femtocell networks.” *IEEE Transactions on Wireless Communications*, **8**(10):4906–4910, 2009.

- [Joh99] Öjvind Johansson. “Simple distributed  $\Delta + 1$  coloring of graphs.” *Information Processing Letters*, **70**(5):229–232, 1999.
- [JPP05] Kamal Jain, Jitendra Padhye, Venkata N Padmanabhan, and Lili Qiu. “Impact of interference on multi-hop wireless network performance.” *Wireless networks*, **11**(4):471–487, 2005.
- [JSX10] Ahmedin Jemal, Rebecca Siegel, Jiaquan Xu, and Elizabeth Ward. “Cancer statistics, 2010.” *CA: a cancer journal for clinicians*, **60**(5):277–300, 2010.
- [JYP88] David S Johnson, Mihalis Yannakakis, and Christos H Papadimitriou. “On generating all maximal independent sets.” *Information Processing Letters*, **27**(3):119–123, 1988.
- [Kar84] Narendra Karmarkar. “A new polynomial-time algorithm for linear programming.” In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311. ACM, 1984.
- [Kar94] Richard M Karp. “Probabilistic recurrence relations.” *Journal of the ACM*, **41**(6):1136–1150, 1994.
- [KBN07] Carrie N Klabunde, Rachel Ballard-Barbash, International Breast Cancer Screening Network, et al. “Evaluating population-based screening mammography programs internationally.” In *Seminars in breast disease*, volume 10, pp. 102–107. Elsevier, 2007.
- [KG08] Saad G Kiani and David Gesbert. “Optimal and distributed scheduling for multicell capacity maximization.” *IEEE Transactions on Wireless Communications*, **7**(1):288–297, 2008.
- [KK18] Sangram V Kadam and Maciej H Kotowski. “Multiperiod Matching.” *International Economic Review*, **59**(4):1927–1947, 2018.
- [KL08] Tae-Hwan Kim and Tae-Jin Lee. “Throughput enhancement of macro and femto networks by frequency reuse and pilot sensing.” In *IEEE International Performance, Computing and Communications Conference*, pp. 390–394. IEEE, 2008.
- [KL17] Pang Wei Koh and Percy Liang. “Understanding black-box predictions via influence functions.” *arXiv preprint arXiv:1703.04730*, 2017.
- [KLK11] Dongho Kim, Jaesong Lee, Kee-Eung Kim, and Pascal Poupart. “Point-based value iteration for constrained POMDPs.” In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [KMT14] John Kennes, Daniel Monte, and Norovsambuu Tumennasan. “The day care assignment: A dynamic matching problem.” *American Economic Journal: Microeconomics*, **6**(4):362–406, 2014.
- [Koc14] Yilmaz Kocer. “Dynamic Matching and Learning.” *working slides*, 2014.

- [KOS14] David R Karger, Sewoong Oh, and Devavrat Shah. “Budget-optimal task allocation for reliable crowdsourcing systems.” *Operations Research*, **62**(1):1–24, 2014.
- [Kri16] Vikram Krishnamurthy. *Partially observed Markov decision processes*. Cambridge University Press, 2016.
- [Kri17] Vikram Krishnamurthy. “POMDP Structural Results for Controlled Sensing.” *arXiv preprint arXiv:1701.00179*, 2017.
- [KSG04] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. “Estimating mutual information.” *Physical review E*, **69**(6):066138, 2004.
- [Kur09] Morimitsu Kurino. “Credibility, efficiency, and stability: A theory of dynamic matching markets.” *Jena economic research papers, JENA*, **7**:41, 2009.
- [LBO99] Franka Loeve, Rob Boer, Gerrit J van Oortmarsen, Marjolein van Ballegooijen, and J Dik F Habbema. “The MISCAN-COLON simulation model for the evaluation of colorectal cancer screening.” *Computers and Biomedical Research*, **32**(1):13–33, 1999.
- [LCV09] Jonathan Ling, Dmitry Chizhik, and Reinaldo Valenzuela. “On resource allocation in dense femto-deployments.” In *IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems*, pp. 1–6. IEEE, 2009.
- [LD17] Emiliya Lazarova and Dinko Dimitrov. “Paths to stability in two-sided matching under uncertainty.” *International Journal of Game Theory*, **46**(1):29–49, 2017.
- [LGD11] David Lopez-Perez, Ismail Guvenc, Guillaume De La Roche, Marios Kountouris, Tony QS Quek, and Jie Zhang. “Enhanced intercell interference coordination challenges in heterogeneous networks.” *IEEE Wireless Communications*, **18**(3):22–30, 2011.
- [Lie07] Michael N Liebman. “Personalized medicine: a perspective on the patient, disease and causal diagnostics.” 2007.
- [LKC17] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. “Interpretable & Explorable Approximations of Black Box Models.” *arXiv preprint arXiv:1707.01154*, 2017.
- [LL17] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions.” In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.
- [LLJ10] Poongup Lee, Taeyoung Lee, Jangkeun Jeong, and Jitae Shin. “Interference management in LTE femtocell systems using fractional frequency reuse.” In *The 12th International Conference on Advanced Communication Technology (ICACT)*, volume 2, pp. 1047–1051. IEEE, 2010.

- [LLJ11] A. Ladanyi, D. Lopez-Perez, A. Juttner, Xiaoli Chu, and Jie Zhang. “Distributed resource allocation for femtocell interference coordination via power minimisation.” In *IEEE GLOBECOM Workshops*, pp. 744–749, Dec 2011.
- [LS09] Robin S Lee and Michael Schwarz. “Interviewing in two-sided matching markets.” 2009.
- [lte] “Access, Evolved Universal Terrestrial Radio, Further advancements for E-UTRA physical layer aspects (Release 10).” *3GPP TS V9*.
- [LVD09] David López-Pérez, Alvaro Valcarce, Guillaume De La Roche, and Jie Zhang. “OFDMA femtocells: a roadmap on interference avoidance.” *IEEE Communications Magazine*, **47**(9):41–48, 2009.
- [LW03] Mei-Ling Ting Lee and George A Whitmore. “First hitting time models for lifetime data.” *Handbook of statistics*, **23**:537–543, 2003.
- [LW06] Mei-Ling Ting Lee, George A Whitmore, et al. “Threshold regression for survival analysis: modeling event times by a stochastic process reaching a boundary.” *Statistical Science*, **21**(4):501–513, 2006.
- [LXH10] Hongjia Li, Xiaodong Xu, Dan Hu, Xiqiang Qu, Xiaofeng Tao, and Ping Zhang. “Graph method based clustering strategy for femtocell interference management and spectrum efficiency improvement.” In *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pp. 1–5. IEEE, 2010.
- [LZ08] Zhi-Quan Luo and Shuzhong Zhang. “Dynamic spectrum management: Complexity and duality.” *IEEE Journal of Selected Topics in Signal Processing*, **2**(1):57–73, 2008.
- [MBH95] Madhav V Marathe, Heinz Breu, Harry B Hunt, Shankar S Ravi, and Daniel J Rosenkrantz. “Simple heuristics for unit disk graphs.” *Networks*, **25**(2):59–68, 1995.
- [MDC06] Luis Meira-Machado, Jacobo De Una-Alvarez, and Carmen Cadarso-Suarez. “Nonparametric estimation of transition probabilities in a non-Markov illness–death model.” *Lifetime Data Analysis*, **12**(3):325–344, 2006.
- [Mih16] Manea Mihai. “Game Theory.” *Massachusetts Institute of Technology: MIT OpenCourseWare*, 2016.
- [Mil11] Rupert G Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011.
- [MIR08] Lisa M Maillart, Julie Simmons Ivy, Scott Ransom, and Kathleen Diehl. “Assessing dynamic breast cancer screening policies.” *Operations Research*, **56**(6):1411–1427, 2008.
- [MS06a] George J Mailath and Larry Samuelson. “Repeated games and reputations: long-run relationships.” *OUP Catalogue*, 2006.

- [MS06b] GJ Mailath and L Samuelson. “Repeated games and reputations: long-run relationships.”, 2006.
- [MWM92] Jeanne S Mandelblatt, Mary E Wheat, Mark Monane, Rebecca D Moshief, James P Hollenberg, and Jian Tang. “Breast cancer screening for elderly women with and without comorbid conditions.” *Ann Intern Med*, **116**:722–730, 1992.
- [NBG10] Mohsin Nazir, Mehdi Bennis, Kaveh Ghaboosi, Allen B MacKenzie, and Matti Latva-aho. “Learning based mechanisms for interference mitigation in self-organized femtocell networks.” In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pp. 1886–1890. IEEE, 2010.
- [Nec08] Marc C Necker. “A graph-based scheme for distributed interference coordination in cellular OFDMA networks.” In *IEEE Vehicular Technology Conference*, pp. 713–718. IEEE, 2008.
- [NHK05] Tim Nieberg, Johann Hurink, and Walter Kern. “A robust PTAS for maximum weight independent sets in unit disk graphs.” In *Graph-theoretic concepts in computer science*, pp. 214–221. Springer, 2005.
- [NTN09] Heidi D Nelson, Kari Tyne, Arpana Naik, Christina Bougatsos, Benjamin K Chan, and Linda Humphrey. “Screening for breast cancer: an update for the US Preventive Services Task Force.” *Annals of internal medicine*, **151**(10):727–737, 2009.
- [OFE15] Kevin C Oeffinger, Elizabeth TH Fontham, Ruth Etzioni, Abbe Herzig, James S Michaelson, Ya-Chen Tina Shih, Louise C Walter, Timothy R Church, Christopher R Flowers, Samuel J LaMonte, et al. “Breast cancer screening for women at average risk: 2015 guideline update from the American Cancer Society.” *Jama*, **314**(15):1599–1614, 2015.
- [Ort01] Jaime Ortega. “Job rotation as a learning mechanism.” *Management Science*, **47**(10):1361–1370, 2001.
- [Per08] Fernando Pérez-Cruz. “Kullback-Leibler divergence estimation of continuous distributions.” In *2008 IEEE international symposium on information theory*, pp. 1666–1670. IEEE, 2008.
- [PGT03] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. “Point-based value iteration: An anytime algorithm for POMDPs.” In *IJCAI*, volume 3, pp. 1025–1032, 2003.
- [PK14] Lydia E Pace and Nancy L Keating. “A systematic assessment of benefits and risks to guide breast cancer screening decisions.” *Jama*, **311**(13):1327–1335, 2014.
- [PLD05] Hanchuan Peng, Fuhui Long, and Chris Ding. “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy.”

- IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8):1226–1238, 2005.
- [POO18] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A Alemi, and George Tucker. “On variational lower bounds of mutual information.” In *NeurIPS Workshop on Bayesian Deep Learning*, 2018.
- [PS91] Jooyoung Park and Irwin W Sandberg. “Universal approximation using radial-basis-function networks.” *Neural computation*, 3(2):246–257, 1991.
- [PST12] Emmanouil Pateromichelakis, Mehrdad Shariat, R Tafazolli, et al. “Dynamic graph-based multi-cell scheduling for femtocell networks.” In *IEEE Wireless Communications and Networking Conference Workshops*, pp. 98–102. IEEE, 2012.
- [RBN14] Simone Romano, James Bailey, Vinh Nguyen, and Karin Verspoor. “Standardized mutual information for clustering comparisons: one step further in adjustment for chance.” In *International Conference on Machine Learning*, pp. 1143–1151, 2014.
- [RCI13] Baharak Rastegari, Anne Condon, Nicole Immorlica, and Kevin Leyton-Brown. “Two-sided matching with partial information.” In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 733–750. ACM, 2013.
- [Riz11] Dimitris Rizopoulos. “Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data.” *Biometrics*, 67(3):819–829, 2011.
- [RKV03] Stephen J Rulyak, Michael B Kimmey, David L Veenstra, and Teresa A Brentnall. “Cost-effectiveness of pancreatic cancer screening in familial pancreatic cancer kindreds.” *Gastrointestinal endoscopy*, 57(1):23–29, 2003.
- [RL93] Subramanian Ramanathan and Errol L Lloyd. “Scheduling algorithms for multi-hop radio networks.” *IEEE/ACM Transactions on Networking (TON)*, 1(2):166–177, 1993.
- [Rob86] John Michael Robson. “Algorithms for maximum independent sets.” *Journal of Algorithms*, 7(3):425–440, 1986.
- [Rot82] Alvin E Roth. “The economics of matching: Stability and incentives.” *Mathematics of operations research*, 7(4):617–628, 1982.
- [Rot89] Alvin E Roth. “Two-sided matching with incomplete information about others’ preferences.” *Games and Economic Behavior*, 1(2):191–209, 1989.
- [RP89] Rajiv Ramaswami and Keshab K Parhi. “Distributed scheduling of broadcasts in a radio network.” In *Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging*, pp. 497–504. IEEE, 1989.

- [RR08] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines.” In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.
- [RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why should i trust you?: Explaining the predictions of any classifier.” In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 2016.
- [RTV15] Dimitris Rizopoulos, Jeremy MG Taylor, Joost Van Rosmalen, Ewout W Steyerberg, and Johanna JM Takkenberg. “Personalized screening intervals for biomarkers using joint models for longitudinal and survival data.” *Biostatistics*, **17**(1):149–164, 2015.
- [SA16] Peter Schulam and Raman Arora. “Disease trajectory maps.” In *Advances in Neural Information Processing Systems*, pp. 4709–4717, 2016.
- [SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [SDV16] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. “Grad-CAM: Why did you say that?” *arXiv preprint arXiv:1611.07450*, 2016.
- [SFM99] Stephen F Sener, Amy Fremgen, Herman R Menck, and David P Winchester. “Pancreatic cancer: a report of treatment and survival trends for 100,313 patients diagnosed from 1985–1995, using the National Cancer Database.” *Journal of the American College of Surgeons*, **189**(1):1–7, 1999.
- [SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences.” *arXiv preprint arXiv:1704.02685*, 2017.
- [SHG18] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. “A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual &Group Unfairness via Inequality Indices.” In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2239–2248. ACM, 2018.
- [SHL12] Nazmus Saquib, Ekram Hossain, Long Bao Le, and Dong In Kim. “Interference management in OFDMA femtocell networks: Issues and approaches.” *IEEE Wireless Communications*, **19**(3):86–95, 2012.
- [Siu16] Albert L Siu. “Screening for breast cancer: US Preventive Services Task Force recommendation statement.” *Annals of internal medicine*, **164**(4):279–296, 2016.
- [SMG01] Cem U. Saraydar, Narayan B Mandayam, and David Goodman. “Pricing and power control in a multicell wireless data network.” *IEEE Journal on Selected Areas in Communications*, **19**(10):1883–1892, 2001.

- [SR92] Scott Y Seidel and Theodore S Rappaport. “914 MHz path loss prediction models for indoor wireless communications in multifloored buildings.” *IEEE Transactions on Antennas and Propagation*, **40**(2):207–217, 1992.
- [SS71] Lloyd S Shapley and Martin Shubik. “The assignment game I: The core.” *International Journal of game theory*, **1**(1):111–130, 1971.
- [SS73] Richard D Smallwood and Edward J Sondik. “The optimal control of partially observable Markov processes over a finite horizon.” *Operations research*, **21**(5):1071–1088, 1973.
- [SS00] Robert Shimer and Lones Smith. “Assortative matching and search.” *Econometrica*, **68**(2):343–369, 2000.
- [SS01] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- [SWB09] Slawomir Stanczak, Marcin Wiczanowski, and Holger Boche. *Fundamentals of resource allocation in wireless networks: theory and algorithms*, volume 3. Springer, 2009.
- [SWH11] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. “Remaining useful life estimation—a review on the statistical data driven approaches.” *European journal of operational research*, **213**(1):1–14, 2011.
- [SXZ16] Mihaela van der Schaar, Yuanzhang Xiao, and William Zame. “Endogenous Matching in a Dynamic Assignment Model with Adverse Selection and Moral Hazard.” *submitted*, 2016.
- [TFL11] Chee Wei Tan, Shmuel Friedland, and Steven H Low. “Spectrum management in multiuser cognitive wireless networks: Optimality and algorithm.” *Selected Areas in Communications, IEEE Journal on*, **29**(2):421–430, 2011.
- [TGR02] David B Thomas, Dao Li Gao, Roberta M Ray, Wen Wan Wang, Charlene J Allison, Fan Liang Chen, Peggy Porter, Yong Wei Hu, Guan Lin Zhao, Lei Da Pan, et al. “Randomized trial of breast self-examination in Shanghai: final results.” *Journal of the national Cancer Institute*, **94**(19):1445–1457, 2002.
- [TKS16] Amy Trentham-Dietz, Karla Kerlikowske, Natasha K Stout, Diana L Miglioretti, Clyde B Schechter, Mehmet Ali Ergun, Jeroen J Van Den Broek, Oguzhan Alagoz, Brian L Sprague, Nicolien T Van Ravesteyn, et al. “Tailoring breast cancer screening intervals by breast density and risk for women aged 50 years or older: collaborative modeling of screening outcomes.” *Annals of internal medicine*, **165**(10):700–712, 2016.
- [TSR12] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. “Efficient crowdsourcing of unknown experts using multi-armed bandits.” pp. 768–773, 2012.

- [TSR14] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. “Efficient crowdsourcing of unknown experts using bounded multi-armed bandits.” *Artificial Intelligence*, **214**:89–111, 2014.
- [UAB11] Serkan Uygungelen, Gunther Auer, and Zubin Bharucha. “Graph-based dynamic frequency reuse in femtocell networks.” In *IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–6. IEEE, 2011.
- [VFC14] Ester Vilaprinyo, Carles Forne, Misericordia Carles, Maria Sala, Roger Pla, Xavier Castells, Laia Domingo, Montserrat Rue, et al. “Cost-effectiveness and harm-benefit analyses of risk-based screening strategies for breast cancer.” *PLoS one*, **9**(2):e86858, 2014.
- [VK92] Isabella Verdinelli and Joseph B Kadane. “Bayesian designs for maximizing information and outcome.” *Journal of the American Statistical Association*, **87**(418):510–515, 1992.
- [WJO68] James Maxwell Glover Wilson, Gunnar Jungner, World Health Organization, et al. “Principles and practice of screening for disease.” 1968.
- [WLL08] Evelyn P Whitlock, Jennifer S Lin, Elizabeth Liles, Tracy L Beil, and Rongwei Fu. “Screening for colorectal cancer: a targeted, updated systematic review for the US Preventive Services Task Force.” *Annals of internal medicine*, **149**(9):638–658, 2008.
- [WO13] Ermin Wei and Asuman Ozdaglar. “On the O (1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers.” *arXiv preprint arXiv:1307.8254*, 2013.
- [WWL09] Yongle Wu, Beibei Wang, KJ Ray Liu, and T Charles Clancy. “Repeated open spectrum sharing game with cheat-proof strategies.” *IEEE Transactions on Wireless Communications*, **8**(4):1922–1933, 2009.
- [WWZ12] Yue Wu, Hui Wang, Biaobiao Zhang, and K-L Du. “Using radial basis function networks for function approximation and classification.” *ISRN Applied Mathematics*, 2012.
- [XAS14] Yuanzhang Xiao, Kartik Ahuja, and Mihaela van der Schaar. “Spectrum sharing for delay-sensitive applications with continuing QoS guarantees.” In *IEEE Global Communications Conference*, pp. 1265–1270. IEEE, 2014.
- [XDV18] Yuanzhang Xiao, Florian Dorfler, and Mihaela Van Der Schaar. “Incentive design in peer review: rating and repeated endogenous matching.” *IEEE Transactions on Network Science and Engineering*, 2018.
- [XS12] Yuanzhang Xiao and Mihaela van der Schaar. “Dynamic spectrum sharing among repeatedly interacting selfish users with imperfect monitoring.” *IEEE Journal on Selected Areas in Communications*, **30**(10):1890–1899, 2012.

- [XS14] Yuanzhang Xiao and Mihaela van der Schaar. “Energy-Efficient Nonstationary Spectrum Sharing.” *IEEE Transactions on Communications*, **62**(3):810–821, 2014.
- [YCC11] Kai Yang, Doru Calin, Chan-Byoung Chae, and Simon Yiu. “Distributed beam scheduling in multi-cell networks via auction over competitive markets.” In *IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2011.
- [Yu06] Huizhen Yu et al. *Approximate solution methods for partially observable Markov and semi-Markov decision processes*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [ZLK08] Ann G Zauber, Iris Lansdorp-Vogelaar, Amy B Knudsen, Janneke Wilschut, Marjolein van Ballegooijen, and Karen M Kuntz. “Evaluating test strategies for colorectal cancer screening: a decision analysis for the US Preventive Services Task Force.” *Annals of internal medicine*, **149**(9):659–669, 2008.