ABSTRACT

Title of dissertation: DYNAMIC RESOURCE ALLOCATION IN WIRELESS HETEROGENEOUS NETWORKS

Vaibhav Singh, Doctor of Philosophy, 2015

Dissertation directed by: Professor Mark Shayman and
Professor Richard La
Department of Electrical and Computer Engineering

Deployment of low power basestations within cellular networks can potentially increase both capacity and coverage. However, such deployments require efficient resource allocation schemes for managing interference from the low power and macro basestations that are located within each other's transmission range. In this dissertation, we propose novel and efficient dynamic resource allocation algorithms in the frequency, time and space domains. We show that the proposed algorithms perform better than the current state-of-art resource management algorithms.

In the first part of the dissertation, we propose an interference management solution in the frequency domain. We introduce a distributed frequency allocation scheme that *shares* frequencies between macro and low power pico basestations, and guarantees a minimum average throughput to users. The scheme seeks to minimize the total number of frequencies needed to honor the minimum throughput requirements. We evaluate our scheme using detailed simulations and show that it performs on par with the centralized optimum allocation. Moreover, our proposed scheme outperforms a static frequency reuse scheme and the centralized optimal partitioning between the macro and picos.

In the second part of the dissertation, we propose a time domain solution to the interference problem. We consider the problem of maximizing the alpha-fairness utility over heterogeneous wireless networks (HetNets) by jointly optimizing *user association*, wherein each user is associated to any one transmission point (TP) in the network, and *activation fractions* of all TPs. Activation fraction of a TP is the fraction of the frame duration for which it is active, and together these fractions influence the interference seen in the network. To address this joint optimization problem which we show is NP-hard, we propose an alternating optimization based approach wherein the activation fractions (AFs) and the user association are optimized in an alternating manner. The sub-problem of determining the optimal activation fractions is solved using a provably convergent auxiliary function method. On the other hand, the sub-problem of determining the user association is solved via a simple combinatorial algorithm. Meaningful performance guarantees are derived in either case. Simulation results over a practical HetNet topology reveal the superior performance of the proposed algorithms and underscore the significant benefits of the joint optimization.

In the final part of the dissertation, we propose a space domain solution to the interference problem. We consider the problem of maximizing system utility by optimizing over the set of user and TP pairs in each subframe, where each user can be served by multiple TPs. To address this optimization problem which is NP-hard, we propose a solution scheme based on *difference of submodular function* *optimization* approach. We evaluate our scheme using detailed simulations and show that it performs on par with a much more computationally demanding difference of convex function optimization scheme. Moreover, the proposed scheme performs within a reasonable percentage of the optimal solution. We further demonstrate the advantage of the proposed scheme by studying its performance with variation in different network topology parameters.

DYNAMIC RESOURCE ALLOCATION IN WIRELESS HETEROGENEOUS NETWORKS

by

Vaibhav Singh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Mark Shayman, Chair/Advisor
Professor Richard La, Co-Chair/Co-Advisor
Professor Sennur Ulukus
Professor Ashok Agrawala
Dr. Mehdi Kalantari

Dedication
To my loving family.

Acknowledgments

I owe my gratitude to all those special people, who made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Mark Shayman, for giving me an invaluable opportunity to work on extremely interesting and practical problems during my graduate studies. I would like to thank him for giving me the liberty to decide the research problems based upon my interests. I would like to thank him for his guidance, discussions, and extreme patience during different phases of my dissertation. I would also like to thank him for listening to my ideas and helping me improve my skills in better presenting the ideas by giving very useful feedback.

I would also like to extend my sincere gratitude to my co-advisor, Professor Richard La. Without his extraordinary theoretical ideas and expertise, this thesis would have been a distant dream. I would like to thank him for being very generous and spending his invaluable time on correcting the manuscripts and helping me improve my writing skills. I would like to thank him for asking me the right questions in the discussions regarding my work, and helping me develop the right thinking approach towards research problems.

I would like to thank Dr. Narayan Prasad, who was my mentor for the work done in collaboration with NEC Labs. I thank him for introducing me to the combinatorial optimization techniques for resource allocation and also making me aware of the industry standards. I thank him for the invaluable discussions, which helped making the dissertation both practically and technically sound. I would also like to thank Prof. Bobby Bhattacharjee for his guidance during my graduate studies. I would like to thank my dissertation committee a lot for suggesting corrections and asking questions, to further improve my dissertation.

My colleagues have enriched my graduate life in many ways and deserve a special mention. I would like to thank Abhijit, Matt and Zihao for our interesting discussions regarding the LTS project. I would like to acknowledge the help and support from my friends Pritam and Biswa regarding my place of residence. I would also like to thank my friends Sumit and Pritam for interesting discussions regarding research and life in general.

I owe my deepest thanks to my family- my parents, my wife and my sister, who have always stood by me and guided me through my career, and have pulled me through toughest times. Words cannot express the gratitude I owe them. I would like to thank the love and happiness of my life, my beautiful wife Ashmita. I would like to thank her for her rock solid support through my graduate studies. I would like to thank her for tolerating me and letting me devote most of my time to studies while staying in Maryland and visiting her in Austin only on alternate weekends. I would like to thank her for proof reading all my technical documents in my graduate studies and helping me improve my technical writing. I would like to thank my ever loving mother, who is the foundation stone of my life and always inspires me to achieve excellence in my work. I would like to thank my father for being an inspiration and motivating me by himself being an epitome of hard work.

Also, I thank my sister Shilpa and my nephew Kinshu for spreading bundles of joy and happiness in my life.

Lastly, I would like to thank the almighty for everything.

Contents

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Motivation 2

1.2 Challenges of heterogeneous networks 4

1.3 Main contributions and organization 6

2 Related Work 9

2.1 Frequency allocation 9

2.2 Association and activation fractions 14

2.3 Coordinated scheduling in MIMO heterogeneous wireless networks . 15

3 Dynamic Frequency Resource Allocation 18

3.1 Introduction 18

3.2 System model 19

3.2.1 Mapping to LTE networks 20

3.3 Centralized allocation 22

3.3.1 Hardness of frequency allocation 25

3.3.2 Approximation of frequency allocation 26

3.4 Distributed allocation 27

3.4.1 PBS only network (1-tier network) 28

3.4.2 2-tier network with MBS 29

3.5 Distributed algorithm for 1-tier networks 30

3.5.1 First stage 30

3.5.2 Second stage 38

3.5.3 Example allocation 40

3.6 Distributed algorithm for 2-tier networks 42

3.6.1 First stage 43

3.6.2 Second stage 44

3.7 Dynamic allocation 46

3.8 Evaluation 47

3.8.1 1-tier algorithm 51

3.8.2 2-tier algorithm 53

4 On Optimal Association and Activation Fractions 62

4.1 Introduction 62

4.2 Problem statement 63

4.3 User association 67

4.3.1 Unified algorithm 72

4.4 AF optimization 80

4.5 Joint association and AF optimization 83

4.6 Evaluation 85

4.6.1 Association 86

4.6.2 Joint association and activation fraction optimization 89

4.7 Association and activation fractions (AFs) for users with limited queue 91

5 Coordinated Scheduling in MIMO Heterogeneous Wireless Networks using Submodular Optimization 94

5.1 Introduction 94

5.2 System model 95

5.3 Problem statement 97

5.4 Proposed approach 101

5.4.1 Difference of submodular (DS) function algorithm 102

5.5 Relaxed problem 107

5.6 Evaluation 109

6 Conclusion 117

Appendix 119

Bibliography 134

List of Tables

3.1 Notations and associated definitions . . . 23

3.2 **Dynamic Algorithm** . . . 47

3.3 Default parameters . . . 48

3.4 Required SINRs and spectral efficiency. . . . 49

4.1 **GLS Algorithm** . . . 73

4.2 **Restricted Greedy Algorithm** . . . 79

4.3 Utility versus α . . . 87

4.4 Local search improvement . . . 88

5.1 **DS Algorithm** . . . 106

5.2 **Greedy maximization** . . . 107

5.3 Default parameters . . . 109

5.4 PF performance of different algorithms. . . . 112

List of Figures

1.1 Example topology consisting of two BSs (M,P) and two users (Alice, Bob). Note that this figure is not to scale. 2

1.2 Association and activation fraction (AF) optimization 3

1.3 Two-timescale solution 6

3.1 Example topology containing 5 PBSs in the coverage area of a MBS BS-6. The conflict graph for the DIM is shown without the macro node (6), which has edges to all other nodes. 19

3.2 Schematic for distributed solutions; the 1-tier algorithm is used as a subroutine in the 2-tier allocation. 27

3.3 Breakdown for stage 1 (Identify Feasible Solution) of the pico-only allocation algorithm 31

3.4 Number of required FRBs in a 1-tier network with varying number of PBSs. 52

3.5 Number of required FRBs with varying number of PBSs (2-tier network). 53

3.6 Number of required FRBs with varying number of users outside PBS coverage area. 55

3.7 Number of required FRBs with varying maximum degree of PBSs in the conflict graph. 56

3.8 Number of required FRBs with varying period of FRB reallocations with greedy joint association. 59

3.9 Number of required FRBs with varying period of FRB reallocation with a fixed SNR-based association scheme. 60

4.1 Joint Association and AF optimization block diagram 84

4.2 Topology 86

4.3 Utility vs α 89

4.4 Activation fractions for heterogeneous networks. 89

4.5 Utility vs iterations 90

5.1 Block diagram 102

5.2 PF system utility with varying distance of closest pico TP from macro TP. 113

5.3 PF system utility with varying number of users per pico TP. 114

5.4 PF system utility with varying number of pico TPs in the cluster. . . 115

Chapter 1: Introduction

Cellular network deployments are saturated in areas with high density of users, such as in urban environments. In the downlink scenario with many users served by a single basestation (BS), it becomes difficult to satisfy the demands of all users, especially as user density and expected demands increase over time. Partitioning the spectrum across neighboring BSs does not address the congestion problem. While adding more spectrum expands the network capacity, it is often not a feasible solution.

Augmenting the macro BSs with low-power BSs provides a scalable solution for increasing the capacity of the network [1, 20, 31]. Service providers can deploy these low power BSs in high-density areas (e.g., malls, stadiums). However, such deployments require efficient resource allocation schemes for managing interference from the low power and macro BSs that are located within each other's transmission range. The following efficient resource management (RM) approaches can further increase the network capacity in a heterogeneous network by managing the interference amongst BSs. Firstly, in the frequency domain, BSs can take advantage of low pairwise interference (macro-pico or pico-pico) to share spectrum, thereby improving the aggregate throughput in the network. Secondly, in the time domain, we can switch off a BS for a fraction of a transmission frame to mitigate interference it causes for the neighboring BSs. We can also actively push more users to associate with small cells having less load, to reduce the load on the macro BS. Thirdly, in the space domain, we can also associate users to more than one BS so as to explore the possibility of network-level MIMO by means of tight cooperation amongst sets of network nodes. In this dissertation, we explore the three-pronged RM solution in frequency, time and space for heterogeneous wireless networks.

1.1 Motivation

Consider the following scenario depicted in Fig. 1.1 consisting of a macro BS (M) which transmits at 50 dbm, and a pico BS (P) which transmits at 10 dbm and is contained in M's coverage area. Alice is associated with M, while Bob is associated with P. Given that M's coverage area extends across the entire region, Bob receives some interference when both M and P transmit on the same frequency. In contrast, due to P's lower transmit power, Alice receives only little interference from P.

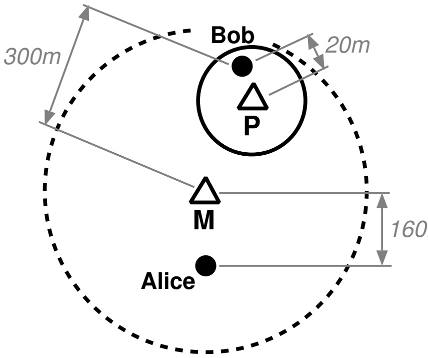
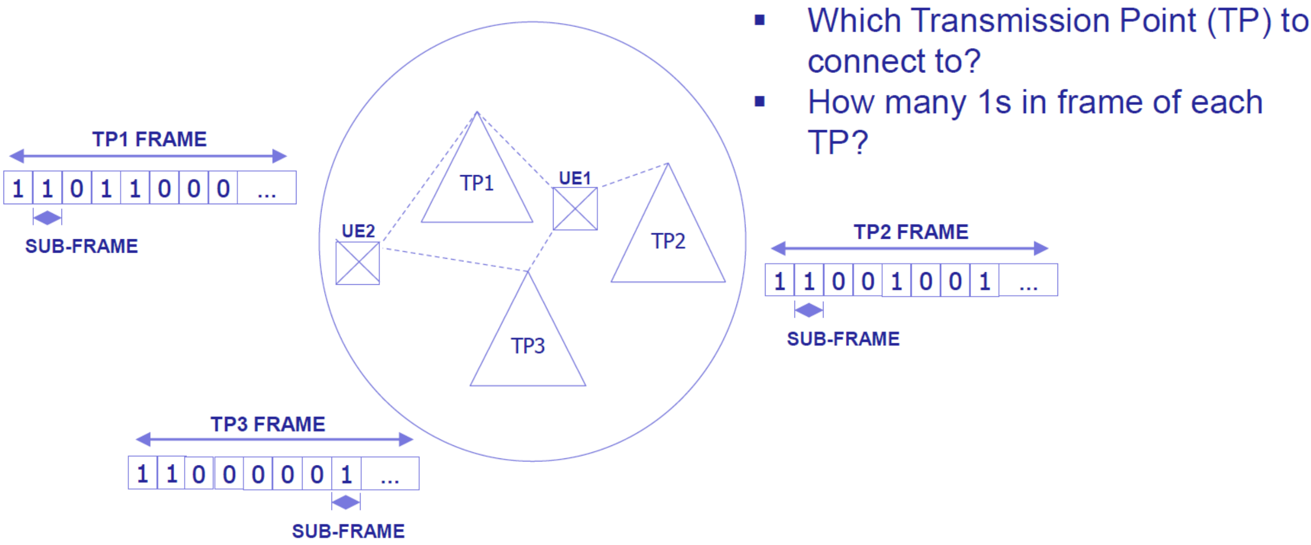


Figure 1.1: Example topology consisting of two BSs (M,P) and two users (Alice, Bob). Note that this figure is not to scale.

Partitioning the available spectrum assigns different frequency resources to each BS (and thus each user). However, since Alice sees little interference from P, M can share any frequency it allocates for Alice with P (and in turn with Bob). Suppose that P can serve Bob at a data rate of 540 kbps using a frequency shared with M. Then, compared to the throughput of, say, 810 kbps for one exclusive frequency for Alice, a shared frequency can achieve an aggregate throughput of 1350 kbps. This demonstrates how sharing can reduce the spectrum required by the network.



- Which Transmission Point (TP) to connect to?
- How many 1s in frame of each TP?

Figure 1.2: Association and activation fraction (AF) optimization

Consider another example topology in Fig. 1.2, where user equipment UE-1 can connect to any of the three TPs. UE-2 can connect to either TP-1 or TP-3. Note that if both the users connect to the same TP, then both of them will only get a fraction of resources available at that particular TP. Therefore, we need to load balance the users amongst TPs using association schemes to allocate resources efficiently. Suppose, UE-1 connects to TP-1 and faces interference from TP-2 and TP-3. One solution as proposed in the literature is subframe-based TP on-off, which is called the almost blank subframe approach. Almost blank subframe (ABS) is an approach to managing inter-cell interference in the *time domain*. The main idea of ABS is that all BSs follow universal frequency reuse and are switched off for a fraction of super-frame (hundred of milliseconds) so as to reduce interference experienced by neighboring BSs. Activation fraction (AF) of each transmission point is the fraction of the frame duration for which it is active, and together these fractions influence the interference seen in the network. In Fig. 1.2, a one in a time slot for a TP denotes that the TP is on in that time slot, and a zero denotes that the TP is switched off for that particular time slot. The fraction of ones in the collection of time slots for a TP is the activation fraction for the particular TP. We need to determine appropriate activation fractions for each TP in order to manage interference in the network efficiently.

Also, multiple transmission point (TP) coordination can be used to manage interference in heterogeneous networks effectively. Using network-level MIMO coordination amongst TPs, we can serve a user by transmitting independent data streams from multiple TPs.

1.2 Challenges of heterogeneous networks

One of the main challenges in designing RM schemes for heterogeneous networks is the random topology of the networks. Recent studies have shown that topologies without one common dominant interferer will be ubiquitous (referred to as small cell only deployments in 3GPP) and in such cases a new RM approach is needed. The

RM schemes for the heterogeneous networks need to take into account interference from the neighboring small cells in addition to the interference caused by macro TPs.

As mentioned in [30, 42] as well as other similar studies, another major challenge pertaining to frequency allocation and determining activation fractions in heterogeneous LTE networks is that the scheme should be executed at the timescale of hundreds of milliseconds in order to take advantage of time-varying loads at various TPs. The reason for this is that small cells (e.g., pico or femto cells) in general serve fewer users than macro cells and, as a result, the aggregated traffic at these cells is expected to fluctuate more than at macro cells, thereby leading to more frequent transitions between empty and non-empty queue states at the associated TPs. This in turn causes more rapid variations in interference to users in neighboring cells. Hence, in order to cope with the fast fluctuations in interference between neighboring cells and use resources more efficiently, resource coordination across heterogeneous cells has to be much more dynamic than in macro-only cases. As mentioned in [6], the re-deployments of resources at the timescale of hundreds of milliseconds ensure efficient resource allocation by closely tracking (i) the changes in user position, and resulting channel gain, and (ii) user arrivals and departures in the system.

The time domain solution pertaining to the design of ABS scheme in heterogeneous networks is quite challenging due to the well recognized interference coupling; while increasing the activation fraction of a TP will help it serve more users (or serve a given set of users better), it injects more interference and is detrimental to all users being served by other TPs.

1.3 Main contributions and organization

In the first two parts of the dissertation, we take a two timescale RM approach. In the time domain solution, as shown in Fig. 1.3, at a coarse timescale TPs coordinate to determine the association and AFs. At a faster timescale, each TP without any coordination from other TPs, uses efficient scheduling algorithms for the associated users. In the final part of the dissertation, we assume that a set of TPs (forming a cluster) coordinate at a fine timescale.

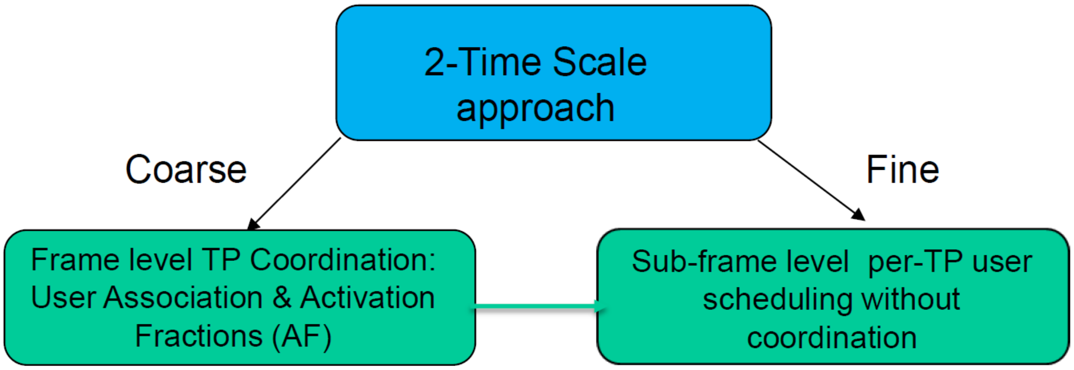

2-Time Scale approach
Coarse
Fine
Frame level TP Coordination: User Association & Activation Fractions (AF)
Sub-frame level per-TP user scheduling without coordination

Figure 1.3: Two-timescale solution

In the first part of the dissertation, we design interference management schemes in the frequency domain, and explore the generalization of the example in Fig. 1.1 involving multiple pico BSs (PBSs) within the coverage area of a macro BS (MBS). We address the problem of frequency allocation at a slow time scale (order of 100 ms) so as to *minimize* the total spectrum usage while meeting the minimum average throughput requirements for all users. By taking advantage of coordinated interference management, we can increase the aggregate throughput of current LTE deployments. Prior work in this area [11], [54], [59] uses coarse-grained feedback from users to assign frequency resources. We improve on this model by incorporating more fine-grained feedback from users, calculating the expected throughput rates at each user when nearby BSs broadcast interfering transmissions.

In the second part of the dissertation, we address interference management in the time domain, and adopt α-fairness utility as the system-wide utility which generalizes all popular utility functions [41], wherein we allow for assigning any arbitrary set of weights (reflecting priorities) to the users. We develop algorithms that yield good solutions for any given input fairness parameter α. These algorithms are obtained by adopting an alternating optimization based approach. The latter approach is well justified since the problem at hand is NP-hard and our goal is to obtain unified low-complexity algorithms that are suitable for all α. For the discrete user association subproblem, we first prove that this subproblem itself is NP-hard and proceed to completely characterize the underlying set function that needs to be optimized. We then suggest and comprehensively analyze a simple centralized combinatorial algorithm (referred to as the GLS algorithm) that involves a greedy stage followed by local search improvements. For the continuous AF optimization subproblem, we adopt the auxiliary function method and show that it is provably convergent and yields a local optima. Further, a key step in the case of AF optimization entails a novel geometric program (GP) formulation.

In the final part of the dissertation, we design an interference management scheme in the space domain by allowing a user to connect to multiple TPs. We prove that the scheduling problem in coordinated multipoint spatial multiplexing, including TP on-off, is NP-hard. We show that the scheduling problem can be expressed as a constrained maximization of difference of submodular functions, and its relaxed version can be expressed as a constrained maximization of difference of concave functions. We then propose an algorithm based on a difference of submodular function optimization. We demonstrate the gains achieved by our algorithm using extensive simulations. We also demonstrate that our algorithm performs at par (within 1.1%) with the more computationally complex difference of concave based scheme and performs within 14.35% of the optimal solution.

The organization of the dissertation is as follows : we first discuss prior resource allocation algorithms proposed in the literature in Chapter 2. We then propose and evaluate a novel dynamic frequency allocation algorithm in Chapter 3. We then study the problem of joint association and activation fractions and propose an efficient algorithm in Chapter 4. In Chapter 5 we propose and evaluate a coordinated multipoint scheduling algorithm based on difference of sub modular function optimization. Finally, in Chapter 6 we conclude the dissertation and discuss possible directions for future work.

Chapter 2: Related Work

In this chapter, we discuss the previous RM schemes proposed in the literature and contrast the schemes with our work. In the first section of this chapter, we present previous work related to interference management schemes in the frequency domain. In the next section, we discuss the previous works related to joint optimization of association and ABS. Finally, we discuss coordinated multipoint schemes to manage interference in hetnets, proposed in the literature.

2.1 Frequency allocation

Different techniques such as interference avoidance, interference management and interference cancellation have been proposed to counter the interference in the unplanned cellular network of low-power BSs deployed in coverage regions of existing MBSs. A broad survey of these techniques is given by Perez et al. [52]. Power control techniques are used in 3GPP for handling dominant interference scenarios as discussed in [4]. The femtocell BSs adjust transmission power of BSs for interference management in heterogeneous cellular network while using the same spectrum for BSs. As pointed out by Perez et al. [49], reducing the radiated power at femtocells also reduces the total throughput of femtocell users while improving the performance of victim UEs.

Madan et al. [42] formulate a centralized optimal power control and resource allocation problem for 2-tier network, which contains low power and macro BSs. As discussed by Perez et al. [52], the computational complexity of solving the minimization problem can be prohibitive if the number of low-power BSs is large. This problem is exacerbated by user arrivals and departures as well as inter-cell mobility, since each such event triggers a recomputation. In the rest of this section, we discuss prior work that provides heuristic solutions.

Perhaps the simplest resource allocation technique in multi-BS scenarios is to partition frequencies between BSs. Prior work has explored probabilistic methods for partitioning the frequency space without assuming coordination between different transmitters. Sundaresan et al. [59] proposed a randomized hashing algorithm to avoid collision of frequency resources with interfering femto BSs. Chandrashekhar et al. [19] use F-ALOHA spectrum access to avoid persistent collisions with interfering femto cells in their allocated spectrum. In contrast, our work in Chapter 3, assumes coordination and communication between PBSs and an MBS to more effectively allocate spectrum.

In general, an MBS can be allocated frequencies exclusive from low power BSs [12, 19, 48, 59] to completely eliminate the cross-tier interference. However, as noted elsewhere [52] and our results, partitioning frequencies is inefficient.

Sundaresan et al. [59] do introduce a sharing model based on femto BS locations with respect to MBS. However, the sharing does not take user location (within the femto cell) into account, which can lead to low SINR for a poorly positioned (cell-edge) user. In our work, we share frequencies by explicitly taking the user locations (and achievable SINR) into account.

There exists another set of studies that examine the problem of power control and user association, using the network utility maximization framework. Chen and Baccelli [21] and Borst et al. [15] use Gibbs sampling to solve the problem of power control over one or more frequencies and user association. Stolyar and Viswanathan [58] propose a gradient-based scheme for power control. These demonstrate the potential to improve the overall network performance measured by the aggregate user utility via efficient schemes. However, there are several key differences between these studies and ours. First, we formulate the frequency allocation problem as one of minimizing the required number of frequency resources subject to minimum throughput requirements, as opposed to the aggregate utility of the users. Second, our goal is to design a practical scheme that can deal with practical constraints, in particular, limited backhaul bandwith and computational resources. While the approaches in [15, 21, 58] are interesting, they are iterative algorithms and require many iterations even for small problems as demonstrated in [15, 21].

Almost blank subframe (ABS) is another approach to managing inter-cell interference in the *time domain*. The main idea of ABS is that all BSs follow universal frequency reuse and are switched off for a fraction of super-frame (hundred of milliseconds) so as to reduce interference experienced by neighboring BSs. Our approach, on the other hand, switches off BSs in the frequency domain to produce different types of frequency resources shared by varying sets of BSs. Thus, our approach can be viewed as a frequency domain analogue of the time domain ABS approach. As mentioned in [5,6], for interference management in heterogeneous networks, we can use power control in the frequency domain and/or in the time domain (e.g., ABS). Although ABS schemes might be easier to implement, our scheme can operate at a comparable timescale (hundreds of milliseconds). Moreover, our scheme provides a more fine-grained control compared to ABS, for it selectively switches off individual frequencies to manage interference, rather than all the frequencies in a subframe.

The idea of sharing spectrum amongst the interfering BSs to increase spectral efficiency is not new. The cell geometry generally assumed in the literature for homogeneous cell scenario is symmetric hexagons, with each BS equidistant from its neighboring BSs. Fractional frequency reuse schemes have been proposed in the literature [32] to maximize the spectral efficiency of the cellular system. In these schemes, a cell is divided into two regions, an inner region comprising users close to the BS and an outer region which is comprised of cell edge users. Neighboring BSs can share frequencies used within inner regions. However, to minimize interference, the outer regions of neighboring BSs use different frequencies. Perez et al. [50] propose a centralized dynamic frequency reuse scheme for interference avoidance to support dynamic traffic in macro cells. However, the sharing does not take user location into account. Ali et al. [10] also propose a centralized dynamic frequency reuse scheme for macro cells while satisfying minimum rate requirements. The sharing in this work is restricted to standard fractional frequency reuse techniques. Such a frequency sharing framework can be inefficient if we assume an unplanned cellular network or one where BSs are added based on demand (hotspots) since the demarcation between inner and outer regions no longer remains uniform.

Prior work has also considered scenarios where a mobile user is allowed to share its frequency resources with the interfering BSs only if the user is guaranteed a minimum SINR [45,46]. Shi et. al [54] also focus on sharing frequency resources amongst interfering femto cells. The interference model in their work captures the interference amongst all the users in the system rather than low-power BSs; each user is modeled to require one resource block irrespective of its position in the associated cell (hence this model ignores individual user requirements). Under this model, the frequency allocation problem can be modeled as a (centralized) vertex coloring problem with every user in the system modeled as a vertex requiring a single color. We compare our techniques to coloring algorithms in Section 3.8.

Cao et al. [17] explore the same idea by using a heuristic coloring technique for non-interfering femto BSs in the first stage. The algorithm in the work includes a second stage in which each femto cell user sends a request to the adjacent BSs to share their resources. The request is granted by the BSs under the constraint that every user is guaranteed a minimum SINR. We note that Necker et al. [46] and Cao et al. [17] use shared frequencies based on a fixed SINR threshold, which leads to a somewhat more restrictive sharing model.

Finally, we note that interference cancellation techniques have been proposed [25, 29] but remain impractical due to the difficulty of eliminating errors in the cancellation process and the cost of custom hardware [52].

2.2 Association and activation fractions

In Chapter 4, we consider the problem of maximizing the alpha-fairness utility over heterogeneous wireless networks (HetNets) by jointly optimizing *user association*, wherein each user is associated to any one transmission point in the network, and *activation fractions* of all transmission points. User association (without AF optimization) is by itself a popular HetNet RM scheme, wherein the interference coupling problem is mitigated by assuming that all TPs in an area of interest have fixed activation fractions, which fixes the interference that would be seen by any user upon being associated to any TP. Under this assumption, association is then determined by approximately maximizing a system utility [16, 56, 63]. Another subsequent promising RM technique combines user association with partial muting of the high power macro TP. In this RM scheme (referred to as combined user association and enhanced intercell interference coordination) the activation fraction for only the Macro TP is optimized together with the user association [13, 14]. The intuition behind this combined approach is that the Macro TP is the dominant interferer for a majority of users in many topologies, so allowing a non-unity activation fraction for only the Macro TP can accrue most of the gain while retaining tractability.

Recent studies have shown that topologies without one common dominant interferer will be ubiquitous (referred to as small cell only deployments in 3GPP), and in such cases a new approach is needed. The problem we seek to solve is geared exactly towards such deployments and generalizes the existing combined user association and eICIC scheme. One attempt to solve our problem would be to extend the solutions proposed for the latter scheme, but then it becomes immediately clear that those solutions do not scale when activation fractions for all TPs have to be optimized. This is because those solutions explicitly maintain a rate for each TP-user link under each possible interference pattern, which grow exponentially in the number of TPs. We propose a simple formulation that imposes activation fractions via Bernoulli activation random variables and yields one average rate expression for each TP-user link that is a closed-form function of all activation fractions. The latter expression is also conservative since it is derived by invoking the fact that the instantaneous rate is a convex function of the activation variables corresponding to the interfering TPs. Interestingly, in the absence of fast fading our rate expression coincides with an approximate rate expression introduced in [55], which considered the problem of determining activation fractions to meet a given set of user traffic demands for a given user association. Our formulation sheds new insight on the observation made in [55] (further confirmed in our simulations here) that the rate expression is in fact very accurate for practical HetNets.

2.3 Coordinated scheduling in MIMO heterogeneous wireless networks

Multiple TP coordination can be used to effectively load balance the users to different TPs and serve them using efficient resource management (RM) algorithms. Further, multiple TP coordination is helpful in mitigating inter-TP interference. Hence, LTE-Advanced proposes the use of coordinated multipoint schemes (COMP) to explore the possibility of network-level MIMO by means of tight cooperation amongst sets of network nodes. Based on the correlation of data streams transmitted from different TPs, COMP transmission strategies can be divided into two categories. The first category is COMP joint transmission (JT) scheme, in which multiple TPs transmit the same data to a user. The signals from different TPs are constructively combined at the receiver to enhance the signal to interference plus noise ratio (SINR) of the user. This scheme is similar to spatial diversity in single TP MIMO cases. Secondly, in an alternative approach, multiple TPs can connect to the same user and each TP transmits independent data streams to increase the data rate of a user. This approach is based on the idea of spatial multiplexing in single TP MIMO. In Chapter 5 of this dissertation, we focus on the COMP spatial multiplexing transmission strategy.

Single TP user association is itself a popular RM scheme where association is determined by approximately maximizing a system utility [16,56,63]. Even the single TP association problem for weighted system utility ensuring proportional fairness (PF) is known to be NP-Hard. Zhao et. al. address the association problem for COMP to minimize data transfer on backhaul in [65]. However, in these works, the association problem is solved at a coarse time scale granularity of a frame constituted by hundreds of subframes. Chen et. al. propose a heuristic for scheduling at subframe granularity for diversity COMP schemes in [22]. However, they restrict the TP-user pairs based on long term signal strength, which depends on average channel gains of the user, rather than instantaneous channel gains. In contrast, our approach captures the advantage of coordination of TPs and channel gains at a fine time scale. Cheng et. al. propose heuristic algorithms in order to minimize the total power consumption in a MISO COMP scenario in [23]. In contrast, we address the problem of maximizing system utility in MIMO COMP scenario, wherein the system utility quantifies a tradeoff between system throughput and system fairness.

Considering the downlink, which is the focus in this dissertation, we see that successive convex approximation technique is a popular approach to ensure tractability of non-concave objective maximization [26, 43, 64]. In this optimization technique, the non-concave optimization is replaced by a series of concave maximization problems. The technique is proven to converge to a local optimum solution of the non-concave problem. This technique is used to solve the non-concave power control problem, which aims to maximize the weighted sum of rates in a single transmit and multiple interfering links setup [39,43]. This approach is also used to solve the single TP MIMO association problem in [53]. We use a discrete version of this technique to propose an efficient difference of submodular function algorithm for the COMP scheduling problem.

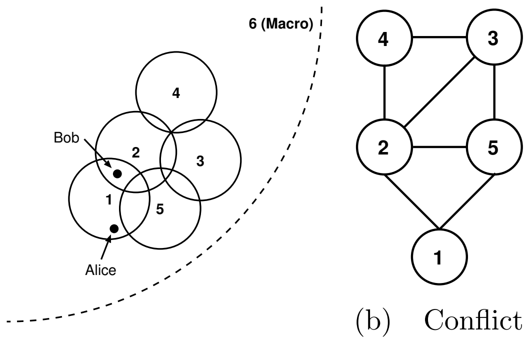
Switching on a TP for each subframe helps to serve the users in the TP coverage area. However, it causes additional interference to users connected to other TPs. Binary (on-off) power control in each subframe has been found to be a particularly effective interference avoidance technique [13,14]. Therefore, we seek to capture the benefits of subframe level binary power control over a HetNet in COMP transmission scenario as opposed to COMP works [22] that assume all TPs to be on.

Chapter 3: Dynamic Frequency Resource Allocation

3.1 Introduction

Deployment of low power pico basestations within cellular networks can potentially increase both capacity and coverage. However, such deployments require efficient frequency allocation schemes for managing interference from the pico and macro basestations that are located within each others' transmission range. Partitioning the available frequencies between the various basestations avoids the problem of interference, but can lead to inefficient spectrum usage.

In this chapter, we introduce a distributed frequency allocation scheme that *shares* frequencies between macro and pico basestations, and guarantees a minimum average throughput to cell users. The scheme seeks to minimize the number of frequencies used. We evaluate our scheme using detailed simulations and show that it performs on par with the centralized optimum allocation (solved by a centralized linear program). Moreover, our proposed scheme outperforms a static frequency reuse scheme and the centralized optimal partitioning between the macro and picos.



(a) Topology

(b) Conflict Graph

Figure 3.1: Example topology containing 5 PBSs in the coverage area of a MBS BS-6. The conflict graph for the DIM is shown without the macro node (6), which has edges to all other nodes.

3.2 System model

We consider a network consisting of one MBS and multiple PBSs within its coverage area. Fig. 3.1a shows an example network topology which fits this model. While we limit our study to the case of a single MBS, we can extend the frequency allocation algorithms to fit scenarios with more than one MBS. We first study a homogeneous (1-tier) network of PBSs, where all BSs use the same transmit power. We then consider a refined network including the MBS, which forms a heterogeneous (2-tier) network.

We focus on the frequency allocation problem at a superframe (order of 100 ms) time rather than the scheduling problem at a frame time scale, for downlink communication between the BS and the users. We assume a fixed association scheme where users choose the BS that provides them the highest SNR, with a bias towards choosing a PBS. Although it may be possible for multiple BSs to serve a user, we choose to consider only the single association case. In addition, we do not consider spatial diversity that can be provided by multiple transmit antennas, which may permit simultaneous scheduling of more than one user on the same frequency at a single BS.

We model the network as a conflict graph $G = (V, E)$, where each vertex $v \in V$ represents a BS in the network. Undirected edges $e \in E$ capture the presence of non-negligible interference at users within a pair of BSs. We say that two BSs BS-i and BS-j are *neighbors* when $e = (i, j) \in E$.

We use a *distance-based interference model* (DIM) to define the edges in the conflict graph, in which two PBSs are neighbors if their distance is less than a certain threshold value. In Fig. 3.1a, the solid circles surrounding each PBS ($\mathrm{PBS}_1 - \mathrm{PBS}_5$) have a radius of half of this threshold distance. Therefore, two PBSs are considered neighbors if their circles overlap. In our model, the MBS is assumed to be a neighbor to all PBSs.

The data rate for a user is largely determined by the signal strength at the user from the serving BS, as well as the interference present from any non-serving BSs sharing the same frequency. We capture the interference experienced by a user by considering the received power from the neighboring BSs of the user's serving BS.

3.2.1 Mapping to LTE networks

A single frequency resource in the LTE standard of OFDMA cellular system consists of a set of subcarriers, called a frequency resource block (FRB). One FRB spans a frequency band of 180 kHz in the LTE standard, which consists of 12 subcarriers separated by 15 kHz between two adjacent subcarriers.

As mentioned in [30, 42] as well as other similar studies, the frequency allocation in heterogeneous LTE networks should be executed at the timescale of hundreds of milliseconds in order to take advantage of time-varying loads at TPs. The reason for this is that small cells (e.g., pico or femto cells) in general serve fewer users than macro cells and, as a result, the aggregated traffic at these cells is expected to fluctuate more than at macro cells, thereby leading to more frequent transitions between empty and non-empty queue states at the TPs. This in turn causes more rapid variations in interference to users in neighboring cells. Hence, in order to cope with the fast fluctuations in interference between neighboring cells and use resources more efficiently, resource coordination across heterogeneous cells ought to be much more dynamic than in macro-only cases. As mentioned in [6], the re-deployments of resources at the timescale of hundreds of milliseconds ensure efficient resource allocation by closely tracking (i) the changes in users positions and resulting channel gains and (ii) user arrivals and departures in the system.

In a 2-tier network, we assume that all PBSs are connected to the MBS through a wired backhaul connection. This backhaul connection can be provided using a fiber cable. However, microwave backhaul connections are emerging as a more practical solution due to the time and cost of setting up a fiber connection [2,3]. The capacity of a microwave backhaul connection between a MBS and PBSs is constrained to hundreds of Mbps [3].

As a cluster of PBSs may have a large number of users during peak hours (e.g., during a football game), signaling traffic on the backhaul needs to be limited. This restricts coordination between BSs, which makes the frequency allocation problem more challenging.

3.3 Centralized allocation

The goal for our allocation scheme is to minimize the total number of FRBs needed to provide the minimum throughput required by the users. In order to minimize the number of needed FRBs, BSs can take advantage of sharing FRBs when some of the users can tolerate the interference present from non-serving BSs. Each BS requires knowledge of the average throughput their users can expect when a FRB is shared with any subset of the neighboring BSs.

We provide a list of notations and their meanings in Table 3.1.

Suppose that $C \in \mathcal{P}(\mathcal{B}) \setminus \emptyset = \mathcal{C}^{\star}$, i.e., a non-empty set of BSs, where $\mathcal{P}(\mathcal{B})$ denotes the power set of $\mathcal{B}$. For each BS $b \in C$ and user $u \in \mathcal{U}_b$, let $r_{u,C}$ denote the expected throughput of user u when it is served by a FRB shared by the BSs in C. Each user estimates these rates by measuring the strength of the pilot signals transmitted by each of the neighboring BSs, and then reports the *rate vector* to its associated BS.

The problem of minimizing the number of FRBs required to provide the minimum throughput to the users can be formulated as an integer LP. As we prove, this minimization problem is in general NP-hard for heterogeneous networks. Moreover, as mentioned in the next subsections, it is hard to provide a small constant

Notation	**Description**
b, o	A BS in the network
$\mathcal{B}$	Set of all BSs in the network
C	A nonempty set of BSs in the network
$B(u)$	BS to which user u is associated
$\mathcal{N}_b$	Set of all neighbors of BS b
x_C	Number of FRBs shared among the BSs in C
$y_{C,u}$	Number of FRBs assigned to user u, which are shared among the BSs in C
t	A FRB type
$O(t)$	BS which owns FRB type t
$P(t)$	Set of participating BSs in FRB type t
$\mathcal{T}_b$	Set of all FRB types owned by a BS b
$N(t)$	Number of FRBs of type t
$c(t)$	Cost associated with FRB type t
L_u	Sorted list of FRB types for user u according to decreasing efficiency
$\mathcal{C}(t,b)$	Compatible set for FRB type t with respect to BS b
$\mathcal{E}(t)$	Extended compatible set of type t
$\mathcal{U}$	Set of all users in the network
$\mathcal{U}_b$	Set of users associated with BS b
$r_{u,C}$	Expected throughput for the user u when it is served by a FRB type shared with the BSs in C
req_u	Average throughput requirement of the user u

Table 3.1: Notations and associated definitions

factor approximation guarantees. For this reason, we formulate a *relaxed* problem of minimizing the total number of FRBs as an LP problem. For every $C \in \mathcal{C}^\star$, let x_C denote the number of FRBs shared among the BSs in C and $y_{C,u}$ refine x_C for a specific user u. We define $\mathbf{x} = (x_C,\ C \in \mathcal{C}^\star)$ and $\mathbf{y} = (y_{C,u},\ C \in \mathcal{C}^\star \wedge u \in \mathcal{U})$. The following minimization problem provides us with a lower bound to the solution:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \sum_{C \in \mathcal{C}^\star} &\ x_C \\ \text{subject to} \sum_{C \in \mathcal{C}^\star} &\ r_{u,C} \cdot y_{C,u} \geq req_u,\ \forall\ u \in \mathcal{U} \\ \sum_{u \in \mathcal{U}_b} & y_{C,u} \leq x_C,\ \forall b \in C, \forall C \in \mathcal{C}^\star \end{aligned} \tag{3.1}$$

The first constraint states that the total rate of user u is at least its minimum required throughput req_u. The second constraint requires that the total number of FRBs that are shared among the BSs in C and are allocated to the users served by each BS b is at most x_C.

Unfortunately, solving the LP in (3.1) requires a centralized entity that has access to all information available at the BSs, i.e., rates $r_{u,C}$. However, even if a centralized agent could collect the rates from all BSs, solving the LP and communicating the solutions back to the BSs in a timely fashion would be difficult; the cardinality of $\mathcal{C}^\star$ grows exponentially with the number of BSs $|\mathcal{B}|$ in the network. For this reason, we focus on designing a *distributed* heuristic algorithm to approximate the centralized solution.

3.3.1 Hardness of frequency allocation

We can assume that disk graphs can model the interference conflict graph amongst the BSs with different transmission powers in a heterogeneous network. We will show that the frequency allocation problem in a heterogeneous network is NP-hard by proving that the vertex coloring problem on disk graphs, which is NP-hard, is a special case of the frequency allocation problem. Consider a frequency allocation problem where the average rates users receive on exclusive FRBs are equal to their minimum throughput requirements. Also, assume that the rates the users receive on shared FRBs are zero (for instance, because the minimum SINR value necessary for the lowest modulation and coding scheme is not achieved). Finally, each BS has a single associated user. Let G be the conflict graph among the BSs. From the above assumptions, if BS set C contains BS b that serves user u, the achieved rates of user u are given by

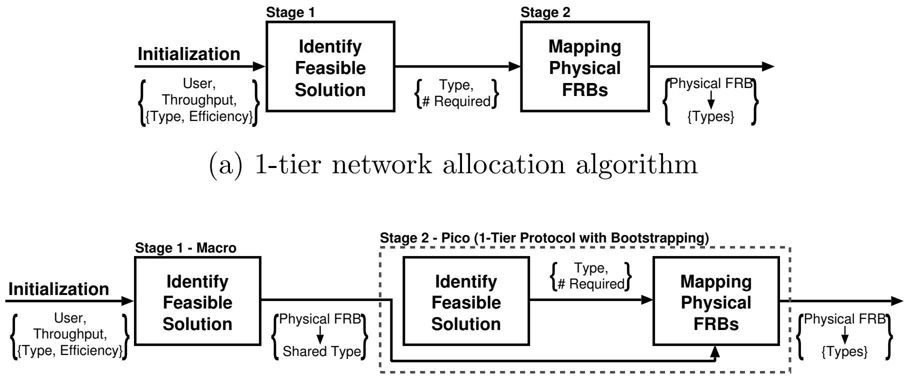
$$r_{u,C} \quad = \quad \begin{cases} 0 & \text{if } C \cap N_b \neq \phi, \\ req_u & \text{otherwise.} \end{cases}$$

Note that each BS requires one exclusive FRB to satisfy the associated user's requirement. Thus, the problem of minimizing the number of required FRBs is equivalent to the vertex coloring problem on disk graphs, where we need to color the graph G with the minimum number of colors in such a way that no two adjacent vertices are assigned the same color. This implies that a feasible frequency allocation would be a solution to the vertex coloring problem on graph G. It is clear that given a vertex coloring problem on any disk graph, we can construct a corresponding or equivalent frequency allocation problem. Hence, if there exists an optimal polynomial time frequency allocation algorithm for a heterogeneous network, it can also solve the vertex coloring problem on any disk graph and, hence, is optimal polynomial time for the NP-hard vertex coloring problem.

3.3.2 Approximation of frequency allocation

The best known online coloring algorithm for disk graphs provides an approximation ratio $O(min(\log n, \log \sigma))$, where σ is the ratio of maximum to minimum BS coverage radius and n is the number of nodes in the graph [18, 28] (references at the end of this document). For $\sigma = 2$, the approximation ratio is equal to 28 [28], which is obtained using first fit algorithm. Moreover, a variant of first fit algorithm can be used to obtain the best known offline coloring approximation ratio of five. Note that although constant factor approximation algorithms are known for the disk coloring problem, the approximation ratios of five and 28 [18, 28] are too high to give tight practical guarantees. We would like to point out that we use the first fit algorithm in the second stage of our scheme, different versions of which are known to provide the best known approximation ratio for offline and online coloring of disk graphs.

Although disk graph coloring is used to prove hardness for the frequency allocation problem, disk graph coloring is much simpler than the original frequency allocation problem that we address in this chapter. This is because even if we can optimally determine the number and types of FRBs required by each BS, the prob-


Stage 1
Initialization
Identify Feasible Solution
User, Throughput, {Type, Efficiency}
Type, # Required
Stage 2
Mapping Physical FRBs
Physical FRB → {Types}
Stage 1 - Macro
Stage 2 - Pico (1-Tier Protocol with Bootstrapping)
Initialization
User, Throughput, {Type, Efficiency}
Identify Feasible Solution
Physical FRB → Shared Type
Identify Feasible Solution
Type, # Required
Mapping Physical FRBs
Physical FRB → {Types}

(a) 1-tier network allocation algorithm

(b) 2-tier network allocation algorithm

Figure 3.2: Schematic for distributed solutions; the 1-tier algorithm is used as a subroutine in the 2-tier allocation.

lem of mapping the requested FRBs to physical FRBs can be modeled as a vertex coloring problem on much more general and complicated graphs compared to conflict graphs in wireless networks. As pointed out further in the chapter, the vertices of this graph are the *requested* FRBs of all BSs, and an edge exists between two vertices when they cannot be mapped to the same *physical* FRB due to interference. The number of colors required to color the graph is equal to the number of physical FRBs required to satisfy the requirements of different FRB types.

3.4 Distributed allocation

We begin with a broad procedural outline of our distributed FRB allocation scheme. In order to simplify the exposition, we first describe our technique on a network consisting of only PBSs, and later augment it to account for the MBS as well.

In our solution, multiple BSs may transmit over the same FRB, even if they are neighbors of each other. To capture and facilitate sharing of FRBs, it is useful to classify FRBs into different *types*: In our model, a FRB is "owned" by a single

BS that chooses to “share” the FRB with other BSs. We denote a FRB type by $(o : v)$, where o is the owner BS and v is a subset of the o's neighbors in the conflict graph. By denoting a FRB of type $(o : v)$, the BS o chooses to share the FRB only with BSs contained in the set v. The BSs in v may interfere with users served by o using the type $(o : v)$. If v is empty, o claims the FRB for its own exclusive use. For example, the FRB type $(1 : 2)$ denotes FRBs BS 1 is willing to share with BS 2.[1] Finally, note that $(1 : 2)$ and $(2 : 1)$ denote two different FRB types with two different owners.

Fig. 3.2 shows a block-level decomposition of our algorithms; Fig. 3.2a outlines the algorithm for 1-tier networks consisting of only PBSs, and Fig. 3.2b illustrates our proposed scheme for heterogeneous 2-tier networks comprising both a MBS and PBSs.

3.4.1 PBS only network (1-tier network)

Recall that we assume users associate with the BS that provides the highest SNR. With multiple overlapping transmission ranges and user positions, different users experience different levels of interference from the non-serving BSs. On one hand, in order to minimize the number of required FRBs, each BS should select FRB types for its users in a way that tries to maximize sharing with neighboring BSs. On the other hand, some users may be located close to neighboring BSs, and sharing with such neighboring BSs may not be beneficial. Hence, it is important to consider the expected throughput of the users when served using shared FRBs. We capture this

[1] For brevity, we write $(1 : 2)$ instead of $(1 : \{2\})$.

trade-off between sharing and expected throughput of a user using the *efficiency* of a FRB type for each user.

Users rank FRB types by their *efficiency*, defined as $r_{u,P(t)}/c(t)$ for a FRB type $t = (o : v)$ and user u. $P(t)$ is the set of participating BSs in FRB type t, i.e., the BSs in $\{o\} \cup v$. $r_{u,P(t)}$ is the expected throughput of user u when it is served by a FRB shared among the set of BSs $P(t)$.

We define the cost of a FRB type as a function which has an inverse relationship to the amount of sharing. Let $\mathcal{N}_o$ be the set of all neighbors of BS-o. The cost of a FRB type $t = (o : v)$ is given by $c(t) = 1 + |\mathcal{N}_o \setminus P(t)|$, i.e., the number of neighbors of BS-o that are not allowed to share the FRB type plus one.

The input to the algorithm is the set of users along with their minimum required throughput and ranked list of FRB types. BSs undertake an iterative distributed procedure, described in Section 3.5.1 to find a feasible solution that meets the minimum throughput requirement for each associated user. The output of this step specifies how many FRBs of each type are required to accommodate all users. We take these requirements and use a distributed first-fit graph coloring heuristic to map compatible FRB type(s) to physical FRBs, as described in Section 3.5.2.

3.4.2 2-tier network with MBS

Our complete solution, outlined in Fig. 3.2b, considers a MBS as well and uses the 1-tier algorithm described above as a subroutine. In the rest of this chapter, we refer to this procedure as the 2-tier algorithm as it considers a heterogeneous network containing one MBS and multiple PBSs.

The input to the 2-tier algorithm is the same as the 1-tier network case. The MBS first creates a feasible FRB allocation for its users only. In doing so, the MBS may choose to share FRBs with some of the PBSs. We describe the procedure used by the MBS in Section 3.6. The PBSs then take the resulting FRB allocation from the MBS and run the 1-tier algorithm, initializing the mapping stage with the FRBs shared with the MBS.

3.5 Distributed algorithm for 1-tier networks

We propose a two-stage algorithm to solve the FRB allocation problem in 1-tier networks. During the first stage, each BS coordinates with neighboring BSs to determine a set of FRB types which satisfy user throughput requirements. These FRB types are *not* the physical FRBs the BSs will use to serve the users. Instead, they are used to identify the set of FRBs that can be shared among different sets of BSs in order to reduce the number of physical FRBs needed to satisfy user throughput requirements. The second stage takes the FRB types determined in the first stage and maps them to physical FRBs while trying to minimize the total number of FRBs used.

3.5.1 First stage

In the first stage, each BS attempts to find a *feasible* set of FRBs through an *iterative* procedure, shown in Fig. 3.3, consisting of three steps in each round until it satisfies

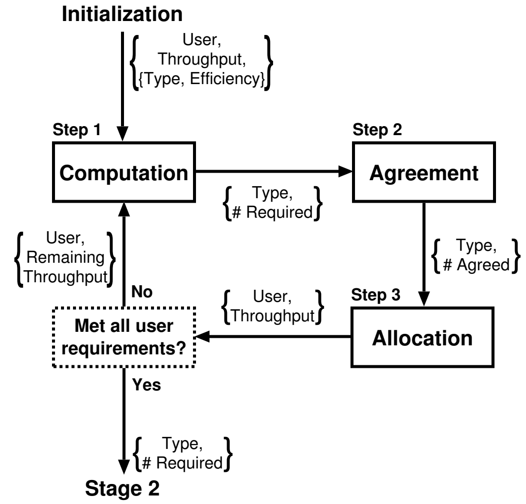


Figure 3.3: Breakdown for stage 1 (Identify Feasible Solution) of the pico-only allocation algorithm

all users' requirements. During the first step, each BS *computes* a set of FRBs of different types in order to satisfy the user requirements, which we call its *wish list*. BSs then exchange wish lists with their neighbors and identify the set of FRBs which the participating BSs agree on, called the *agreed list*. Finally, each BS takes the FRBs in the agreed list and *allocates* them to its associated users. Each BS repeats these steps until its users' minimum requirements are met, after which it passes the number of each FRB type allocated in the first stage to the second stage.

Initialization

During initialization, each BS b sorts the FRB types according to decreasing efficiency for every user $u \in \mathcal{U}_b$. We denote the sorted list for user u by L_u, where $L_u(k)$ is the k^{th} most efficient FRB type for user u. The length of L_u is $|\mathcal{P}(\mathcal{N}_b)|$, based on the number of available FRB types for b.

Initially set $req'_u = req_u$, where req_u is the required minimum throughput and

req'_u represents the remaining minimum throughput for future rounds to satisfy. Starting with priority $k = 1$, we perform the following three steps.

Step 1 - Computation

Each BS computes the number of FRBs of type $L_u(k)$ for all users $u \in \mathcal{U}_b$. Taking into consideration the remaining throughput requirements of its users, the BS adds $req'_u/r_{u,L_u(k)}$ FRBs of type $L_u(k)$ to its *wish list* for each user u. However, since sharing a FRB requires coordination among all participating BSs in a type, a BS may not be able to obtain all FRBs it requires unless the neighbors also wish to share the same number of *compatible* FRBs.

Step 2 - Agreement

Each BS exchanges its wish list with all neighbors, and then determines the set of FRBs agreed upon by all participating BSs, called the *agreed list.* Computation of this set requires determining which FRB types are compatible, and therefore can share a FRB.

We first introduce two concepts, *matching* and *compatibility*, which allow us to build pair-wise constraints between FRB types, termed *original constraints.* We motivate these concepts by considering the earlier example topology and corresponding conflict graph shown in Fig. 3.1.

Matching types- Assume there is a requirement of 1 FRB of type $(2 : 3, 5)$ at BS-2, 1 FRB of type $(3 : 2, 5)$ at BS-3, and 1 FRB of type $(5 : 2, 3)$ at BS-5. We can map all of these FRB types to the same physical FRB, as these types denote the sharing between the same set of participating BSs ($\{2, 3, 5\}$). We term this relationship between the FRB types as *matching*.

Two FRB types, $(o : v)$ and $(o' : v')$, are said to be *matching types* if (i) $o \neq o'$ and (ii) $\{o\} \cup v = \{o'\} \cup v'$. A *matching set* is a maximal set of FRB types such that (i) every pair of FRB types from the set are matching types and (ii) there is no other FRB type that is a matching type to the elements in the set and does not belong to the set.

Compatible types and sets- When BSs select matching FRB types, it means that they are willing to share with other participating BSs. However, a BS may be able to share a FRB with another BS without selecting matching types. For example, consider two FRB types $(1 : 2)$ and $(2 : 1, 4)$. While they are not matching types, as BS-1 and BS-4 do not interfere with each other, it does not matter to BS-1 whether BS-2 selects a FRB of type $(2 : 1)$ or $(2 : 1, 4)$.

We formalize this observation as follows: Suppose BS-o and BS-o' are neighbors which select FRB types $(o : v)$ and $(o' : v')$, respectively. For simplicity, we denote $\mathcal{N}_o \cap \mathcal{N}_{o'}$ by $\mathcal{N}_{o,o'}$. Two FRB types $(o : v)$ and $(o' : v')$ are said to be *compatible* if (i) $o \in v'$ and $o' \in v$, and (ii)

$$\mathcal{N}_{o,o'} \cap (v \setminus \{o'\}) = \mathcal{N}_{o,o'} \cap (v' \setminus \{o\}) \tag{3.2}$$

which implies that there is no common neighbor between BS-o and BS-o' with which only one of them elects to share an FRB.

Given a FRB type $t = (o : v)$, the *compatible set* $\mathcal{C}(t, o')$ with respect to a neighboring BS-o' contains FRB types that are compatible with $(o : v)$ and owned by BS-o' where $o' \neq o$. For example, the compatible set for FRB type $(1 : 2)$ with respect to BS-2 is $\{(2 : 1),\ (2 : 1, 3),\ (2 : 1, 4),\ (2 : 1, 3, 4)\}$.

Note that matching types are compatible with each other.

Original constraints- We look to define constraints based on the compatibility of various FRB types in order to compute the agreed list. The most basic constraints exist between pairs of matching FRB types, ensuring that the total numbers of FRB types from both compatible sets (with respect to the other owner BS) are equal. We call these *original constraints.*

Consider the following three matching FRB types: $(2 : 3, 5)$, $(3 : 2, 5)$, and $(5 : 2, 3)$. The pair-wise original constraints for the types in the matching set are listed below. The first constraint corresponds to the compatible set of $(2 : 3, 5)$ with respect to BS-3 and $(3 : 2, 5)$ with respect to BS-2. In total there are $\binom{3}{2}$ original constraints, which generalizes to $\binom{n}{2}$ for a matching set containing n different FRB types.

$$N(2 : 3, 5) + N(2 : 1, 3, 5) = N(3 : 2, 5)$$

$$N(2 : 3, 5) + N(2 : 3, 4, 5) = N(5 : 2, 3)$$

$$N(3 : 2, 5) + N(3 : 2, 4, 5) = N(5 : 2, 3) + N(5 : 1, 2, 3)$$

We define the general form of original constraints by considering two matching FRB types, t_1 and t_2.

$$\sum_{t'_1 \in \mathcal{C}(t_1, O(t_2))} N(t'_1) = \sum_{t'_2 \in \mathcal{C}(t_2, O(t_1))} N(t'_2) \tag{3.3}$$

Algorithmic Constraints- Since each pair of matching types gives rise to an original constraint, the presence of many interfering BSs introduces a large number of original constraints. Moreover, even for a fixed FRB type owned by BS-b, the compatible set can vary from one neighboring BS to another, increasing the computational burden for identifying all necessary compatible sets. For this reason, instead of working with these original constraints, we consider a new form of constraint which captures the complete relationship between FRB types within a matching set and is easier to work with.

We first define an *extended compatible set* for an FRB type which, unlike the compatible set, takes into account compatibility with respect to *all* owners of FRB types in the matching set.

Given a FRB type $t = (o : v)$, we let $H(t)$ represent the set of BSs which are a neighbor of BS-o but not a neighbor of at least one of the owners in the matching set of type t. We define the extended compatible set as:

$$\mathcal{E}(t) = \{(o : v, H') | H' \in \mathcal{P}(H(t))\} \tag{3.4}$$

where $\mathcal{P}(H(t))$ represents the power set of $H(t)$.

Just as original constraints are derived from the compatible sets, we derive the algorithmic constraints from the extended compatible sets of FRB types. Given a matching set containing FRB types $\{t_1, t_2, ..., t_n\}$ we define the algorithmic constraint by

$$\sum_{t'_1 \in \mathcal{E}(t_1)} N(t'_1) = \sum_{t'_2 \in \mathcal{E}(t_2)} N(t'_2) = ... = \sum_{t'_n \in \mathcal{E}(t_n)} N(t'_n) \tag{3.5}$$

Proposition 3.1. *Algorithmic constraints are equivalent to the set of original constraints.*

Proof : Appendix A.

Third Step - Allocation

Once a given BS-b knows the number of FRBs available for each type, it allocates them to each user. There are different ways in which the allocation can be carried out. Here, we describe one allocation scheme similar to a waterfilling algorithm. An alternative second approach is explained in Appendix B.

Let $avail(t)$ denote the number of available FRBs of type t, which is initially set to the agreed number of FRBs of type t from the second step (already calculated in Line 7). Users are first sorted by decreasing value of efficiency of their current priority FRB type $L_u(k)$. Then, starting with the first user in sorted order, we perform the following steps:

1. Temporarily assign $\alpha = \min(req'_u / r_{u,P(L_u(k))},\ avail(L_u(k)))$ FRBs of type $L_u(k)$ to user u;

2. Update available FRBs of type $L_u(k)$ to $avail(L_u(k)) = avail(L_u(k)) - \alpha$.

If the above temporary assignments to the users satisfy the algorithmic constraints, we update the remaining throughput requirement of each user u to $req'_u - \alpha \cdot r_{u,P(L_u(k))}$. We also update the FRB types allocated to the users according to their temporarily assigned FRB types.

If the temporary assignments do not satisfy the algorithmic constraints, then we repeat the k^{th} round after updating the FRB type requirements in the wishlist based on the temporary FRB assignments. Therefore, if there is at least one BS for which the temporary FRB allocations do not fulfill the algorithmic constraints, we need to repeat the round until all BSs can honor the algorithmic constraints. However, the alternative approach described in Appendix B guarantees that the algorithmic constraints will always be met by considering only allocations that satisfy them.

The iterative procedure described above terminates after a finite number of rounds, after satisfying the minimum throughput requirements for all users. Once the procedure reaches the exclusive FRB type in a user's priority list, we can fully satisfy the remaining throughput requirement of the user with FRBs of this type. Finally, we round up the required number of FRBs of each type.

We use the output from the iterative procedure, i.e., the number of FRBs of each type, as an input to the following distributed LP problem solved at each BS. We denote by $\mathcal{T}_b$ the set of all different FRB types owned by BS b and, for each type $t \in \mathcal{T}_b$, nx_t is the number of FRBs of type t assigned in the first stage.

The number of FRBs of type t assigned to user $u \in \mathcal{U}_b$ is denoted by $ny_{t,u}$ and let $\mathbf{ny} = (ny_{t,u}, u \in \mathcal{U}_b \text{ and } t \in \mathcal{T}_b)$.

$$
\begin{aligned}
\min_{\mathbf{ny}} & \sum_{t \in \mathcal{T}_b} c(t) \cdot \left(\sum_{u \in \mathcal{U}_b} ny_{t,u} \right) \\
\text{subject to} & \sum_{t \in \mathcal{T}_b} r_{u,P(t)} \cdot ny_{t,u} \geq req_u, \ \forall\ u \in \mathcal{U}_b \\
& \sum_{u \in \mathcal{U}_b} ny_{t,u} = nx_t, \ \forall\ t \in \mathcal{T}_b \setminus \{(b : \emptyset)\}
\end{aligned}
\tag{3.6}
$$

The LP presented above reduces the number of FRBs needed by each BS without violating any agreements with the neighbors. We keep the number of FRBs of every *shared* type fixed (second constraint), and minimize the number of exclusive FRB types by assigning available FRBs to the users more efficiently than in the iterative procedure.

3.5.2 Second stage

The first stage computes the required FRBs of each type, but does not allocate them to *physical* FRBs. For example, consider a conflict graph that is not connected such that we can partition it into two disjoint subgraphs. It is clear that we could allocate a single physical FRB to both sets with no resulting interference. The second stage serves to establish this physical FRB allocation.

We formulate the problem of mapping requested FRBs in the first stage to physical FRBs as a vertex coloring problem. The vertices of the graph are the requested FRBs of all BSs, where edges exist between two vertices when they are unable to be mapped to the same physical FRB due to interference. The number of colors required to color the graph is equal to the number of physical FRBs required to satisfy the output from the first stage.

We design a distributed greedy first-fit approximation algorithm for the coloring problem: We take the requested FRB types of each BS and map it to the first physical FRB that can accommodate the FRB type. Before a FRB of type t can be mapped to a physical FRB, the following conditions must be met:

(a) The owner of the FRB type t cannot own any other FRB type already mapped to the physical FRB.

(b) The neighbors of BS $O(t)$ which are not included in the set $P(t)$, i.e., $\mathcal{N}_{O(t)} \backslash P(t)$, constitute a complement set of BSs for type t. Any BS in the complement set of type t should not be a participating BS of any FRB type already mapped to this physical FRB.

(c) Any BS in the union of the complement sets of FRB types already mapped to the physical FRB should not be in $P(t)$.

Using the example in Fig. 3.1, consider the addition of type $(3 : \emptyset)$ to a physical FRB which is already assigned to type $(2 : 4, 5)$. In this case, the above conditions (b) and (c) for compatibility are not met. Therefore, we cannot map type $(3 : \emptyset)$ to this physical FRB. On the other hand, type $(5 : 2)$ satisfies all of the compatibility requirements.

If we cannot map a FRB type to physical FRBs that are already assigned to other FRB types, we assign it to a new physical FRB. The algorithm repeats until we map all FRB type requirements to physical FRBs.

Our algorithm is designed to minimize the total number of physical FRBs required to accommodate the throughput requirements of all users in the system. However, it can also handle a scenario where we are given a set of available physical FRBs to allocate as follows: We first compute an initial allocation to meet users' requirements, making use of our algorithm. Then, if there are any remaining FRBs, they are assigned to the users in proportion to the initial allocation. Thus, each user in the system would get a throughput that is proportional to its minimum requirement.

3.5.3 Example allocation

We provide an example to demonstrate the benefit of our 1-tier allocation approach over other schemes such as reuse-1 and exclusive frequency allocation. Consider a portion of the topology shown in Fig. 3.1a consisting of PBS_1, PBS_2, and PBS_5. We assume user Alice is associated with PBS_1, Bob associated with PBS_2, and Carol associated with PBS_5. The rate for each FRB type for all users are listed below (in kbps), sorted in decreasing order of efficiency.

Alice	Bob	Carol
$(1:2) \rightarrow 270$	$(2:1) \rightarrow 360$	$(5:1,2) \rightarrow 270$
$(1:\emptyset) \rightarrow 360$	$(2:\emptyset) \rightarrow 360$	$(5:2) \rightarrow 360$
$(1:2,5) \rightarrow 90$	$(2:1,5) \rightarrow 120$	$(5:1) \rightarrow 270$
$(1:5) \rightarrow 120$	$(2:5) \rightarrow 180$	$(5:\emptyset) \rightarrow 360$

Assume each user requires an average throughput of 1000 kbps.

1. *Priority* $k = 1$ *:* BS-1 requires 4 FRBs of type $(1 : 2)$ for Alice, BS-2 requires 3 FRBs of type $(2 : 1)$ for Bob, and BS-5 requires 4 FRBs of type $(5 : 1, 2)$ for Carol. The agreed list for this round consists of 3 FRBs of types $(1 : 2)$ and $(2 : 1)$, according to the constraint that $N((1 : 2)) = N((2 : 1))$. We update Alice's residual throughput requirement to $1000 - 3 * 270 = 190$ kbps, and Bob's residual throughput requirement to 0 kpbs. BS-2 stops participating in the algorithm as it has satisfied all user requirements.

2. *Priority* $k = 2$ *:* BS-1 requires 1 FRB of type $(1 : \emptyset)$ for Alice, and BS-5 requires 3 FRBs of type $(5 : 2)$ for Bob. Since BS-1 is only interested in allocating exclusive subcarriers, it satisfies Alice's residual throughput requirement and sets it to 0 kbps. The agreed list for this round is empty. BS-1 stops participating in the algorithm as it has satisfied all user requirements.

3. *Priority* $k = 3$ *:* BS-5 requires 4 FRBs of type $(5 : 1)$ for Carol. The agreed list for this round is empty again.

4. *Priority* $k = 4$ *:* BS-5 requires 3 FRBs of type $(5 : \emptyset)$ for Carol. Since BS-5 is only interested in allocating exclusive subcarriers, it satisfies Carol's throughput requirement and sets it to 0 kpbs. The agreed list for this round is empty. BS-5 stops participating in the algorithm as it has satisfied all user requirements.

The final list of requirements from the first stage is given in the table below.

Since (1 : 2) and (2 : 1) are compatible, this allocation requires a total of 7 physical FRBs.

FRB Type	# Required
(1 : 2), (2 : 1)	3
$(1 : \emptyset)$	1
$(5 : \emptyset)$	3

In contrast to our allocation scheme, the reuse-1 algorithm requires $\lceil 1000/90 \rceil = 12$ FRBs and the exclusive frequency allocation algorithm requires $3 * \lceil 1000/360 \rceil = 9$ FRBs.

3.6 Distributed algorithm for 2-tier networks

While we can apply the proposed 1-tier algorithm for use in 2-tier networks, the performance degrades when applied to conflict graphs in which nodes have high degrees. Since we assume all users associated with a PBS see interference from the MBS, the MBS node in the conflict graph has degree $|\mathcal{B}| - 1$. Therefore, we propose a new algorithm for 2-tier networks which handles users associated with the MBS first, removing the need for PBSs to consider the MBS.

In addition, we group the PBSs into *clusters* to simplify allocation and reduce the computational complexity of our algorithm. Clusters capture the topological structure in practical deployments, and consist of a group of PBSs within an area such as a stadium or shopping mall. Consider a graph where the vertices correspond to the PBSs. Edges exist between two PBSs if their distance is less than a given threshold d, where d is chosen such that inter-cluster interference is negligible. Each connected component in this graph corresponds to a cluster of PBSs.

We propose a two stage algorithm for allocation in 2-tier networks. The MBS first allocates physical FRBs to satisfy its associated users, and sends out allocation information which pertains to the clusters. The PBSs in each cluster run the 1-tier algorithm to satisfy user requirements, bootstrapping with the physical FRBs shared with the MBS.

3.6.1 First stage

We classify the users associated with the MBS into three classes based on the amount of *sharing* they can do with the PBS clusters. We assume that there are H PBS clusters:

Class A – Class A users are sufficiently far from all H PBS clusters. More specifically, when a class A user shares its FRB with *all* the PBSs, its data rate does not degrade by more than a certain threshold β compared to its data rate on an MBS exclusive FRB. The threshold β is set to 20 percent in our numerical studies.

Class B – Class B users are sufficiently far from $H-1$ clusters but close to one PBS cluster. The closest cluster to the user is called the *interfering cluster*. When a class B user is served on an FRB shared with all the PBSs in the $H-1$ non-interfering clusters and some PBSs in the interfering cluster, its data rate does not decrease by more than β (compared to its data rate on an MBS exclusive FRB). Thus, class B users can be served using FRBs shared with the PBSs in non-interfering clusters and, possibly, some of PBSs in interfering clusters which are also sufficiently far, without sacrificing their throughput significantly.

Class C – All other users are class C users. These users are in general served only by FRBs exclusive to the MBS.

Each of the assigned FRB types is mapped to a distinct physical FRB, as FRB types owned by the MBS are not compatible with each other. The MBS sends out a subset of this list of physical FRBs to each cluster in the network. A cluster's list contains the physical FRBs, and their associated MBS-shared FRB type based on the PBSs that can share the FRB within the cluster.

3.6.2 Second stage

Each PBS runs the 1-tier algorithm described in Section 3.5, coordinating with the other PBSs in their cluster. Stage 1 of the algorithm remains the same, where each BS identifies the feasible solution that satisfies its users. In stage 2, we now bootstrap the set of physical FRBs shared with the MBS prior to running the distributed first-fit approximation algorithm. This algorithm chooses to first use existing compatible MBS shared FRBs before allocating other FRBs, ensuring utilization of the physical FRBs shared with the MBS. If all FRBs that the MBS is willing to share with the ℓth cluster ($\ell = 1, 2, \cdots, H$) are exhausted, then the cluster computes additional FRBs that are not shared with the MBS. This can be done by using the 1-tier network algorithm after reducing the required throughput of the users by the amount provided via the FRBs shared with the MBS.

However, we need to redefine the notion of compatibility for FRB types shared by the MBS. In the first stage, MBS allocates FRBs to users based on their class. It is possible that a MBS is willing to share a FRB with all PBSs in a cluster, yet more efficient if only a subset of the PBSs actually share it due to interference. In contrast to the previous notion of compatibility introduced in (3.2), it is acceptable for the PBS FRB type to elect not to share with all neighboring BSs contained in the MBS FRB type. FRB types t_m and t_p, owned by the MBS and a PBS, respectively, are *compatible* if $P(t_p) \subset P(t_m)$.

A key idea that we exploit in our algorithm is the following: Since we assume that different clusters of PBSs cause minimal interference to PBSs in other clusters, any FRBs used by some cluster of PBSs (which are not shared with the MBS) can be shared with all other clusters of PBSs with minimal interference. Suppose that each cluster ℓ of PBSs ($\ell = 1, 2, \cdots, H$) computes the number of additional FRBs of each type it needs to satisfy the minimum throughput requirements of the users served by the cluster, which we denote by ξ_ℓ. Based on the above observation, the total number of FRBs we need is approximately equal to the sum of the FRBs needed by the MBS (including those shared with PBSs) and the maximum among the additional required FRBs of the clusters, i.e., $\max_{\ell=1,2,\ldots,H} \xi_\ell$.

While we only consider a 2-tier scenario with a single MBS, our algorithms can be used to handle scenarios where more than one MBS interferes with PBSs. A simple solution is to allocate the MBSs the FRBs not shared with other MBSs and then use the 2-tier algorithm to determine sharing of MBS FRBs with PBSs. A more interesting solution is to use the 1-tier algorithm to assign FRBs to the MBSs. In this case, we would have multiple macro FRB types based on the conflict graph among the MBSs. Using the 1-tier algorithm, we calculate the respective FRB types required by MBS users and the corresponding MBS-shared physical FRBs. Further, for each different set of MBSs which share the MBS FRBs, we use 2-Tier algorithm to determine sharing of these FRBs with PBSs.

3.7 Dynamic allocation

We also consider a dynamic system where the user population varies with time as opposed to the static case considered in previous sections. Every instance of departing user associated with a BS creates a slack of FRBs of the type previously allocated to the departing user at the BS. FRB slack at a BS is defined as the set of FRBs available to the BS but not currently allocated to any user associated with the BS.[2] In the 2-tier algorithm, the MBS-shared physical FRBs may be sufficient to satisfy the throughput requirements of the users associated with the PBSs. Further, the surplus MBS-shared FRBs create an 'implicit slack' at the PBSs. The PBSs can share surplus physical FRBs based on the associated MBS-shared FRB types. A newly arriving user can associate with a BS that can satisfy the requirement of the user with the available slack of FRBs. Thus, given sufficient slack, the FRB allocation to a new user does not require any additional FRBs.

We propose a greedy joint association and FRB allocation scheme for the dynamic system using the slack of FRBs available at BSs. The pseudocode for the

[2] In practice, these extra FRBs will be reallocated to other users to increase their throughput.

Table 3.2: **Dynamic Algorithm**

```
1: If Periodic optimization time instance Then
2:     Use the 2-tier allocation algorithm
3:     Slack = Set of unused MBS-shared FRBs
4: Endif
5: If User $u_d$ departs Then
6:     Slack = Slack $\cup$ FRB types used by the departing $u_d$
7: Endif
8: If User $u_a$ arrives Then
9:     If Slack available at BS $b$ satifies $u_a$ Then
10:        $u_a$ associates with $b$
11:        Allocate the required FRBs to $u_a$ from the Slack
12:        Update the Slack
13:    Else
14:        $u_a$ associates with BS providing the highest SNR
15:        Allocate exclusive FRBs to $u_a$
16:        Update the Slack
17:    Endif
18: Endif
```

scheme is provided in Table 3.2. *(Line 1-3)* We run the 2-tier allocation algorithm periodically (with period of T_a) to (re)-allocate the FRBs. *(Line 9-12)* An arriving user associates with the BS that has sufficient slack of FRBs available to satisfy the throughput requirement of the user. When more than one BS can satisfy the user's requirement, the user associates with the BS that offers the highest SNR to the user. *(Line 13-16)* When no BS has enough slack, the user associates with the BS offering the highest SNR and we assign additional FRBs exclusive to the serving BS to satisfy the user requirement.

3.8 Evaluation

We present an evaluation of our 1-tier and 2-tier allocation algorithms. Table 3.3 contains the default parameters used in our numerical studies and simulations unless

Parameter	Value
Macro transmission power per FRB	10^3 mW
Pico transmission power per FRB	0.1 mW
Path Loss Exponent	3.5
Coverage radius of the macro BS	700 m
Coverage radius of the PBS BS	50 m
Power spectral density of noise	-172 dBm/Hz
log-normal shadowing standard deviation	6 dB
BW of a frequency resource block	180 kHz
Number of users in a hotspot area	20
User min. avg. throughput requirement	2000 kbps
Min. distance between macro and PBS BS	50 m
Min. distance between any 2 PBS BSs	30 m
Threshold distance d for defining a cluster	150 m
Threshold distance for interference	150 m

Table 3.3: Default parameters

otherwise stated.

At the beginning of each realization or sample path, we generate mutually independent log-normal random variables to model the shadow fading between all (user, BS) pairs. Each user computes its SINR, according to the path loss to each BS based on the parameters given in Table 3.3 and the realized random variables for shadow fading. The users report their SINR to their serving BSs, which use the reported SINR values to choose appropriate modulation and coding schemes according to the required SINRs and corresponding data rates based on the spectral efficiency of the modulation and coding scheme shown in Table 3.4 as used by Perez et. al [51]. The throughput of a user on a particular FRB type is the product of the spectral efficiency and the bandwidth of an FRB .

We determine the threshold distance for interference between any two PBSs (as shown in Table 3.3) based on a maximum allowable interference power received

SINR threshold (dB)	Spectral Efficiency((bit/s)/Hz)
-6.5	0.15
-4.0	0.23
-2.6	0.38
-1.0	0.6
1.0	0.88
3.0	1.18
6.6	1.48
10.0	1.91
11.4	2.41
11.8	2.73
13.0	3.32
13.8	3.90
15.6	4.52
16.8	5.12
17.6	5.55

Table 3.4: Required SINRs and spectral efficiency.

by a user present at any location in the coverage area of a PBS; two PBSs do not interfere if every user associated with either of the PBSs receives a SINR of at least 6.6 dB in order to guarantee a spectral efficiency of 1.48 (bit/s)/Hz. We use the same threshold distance value for defining a cluster.

PBS user locations are uniformly distributed within the coverage area of PBSs. In the 2-tier network topology, the remaining users' locations are uniformly distributed within the coverage area of the MBS. As proposed by Peng et. al [48], each user associates with the BS that provides the highest SNR with 6 dB bias in favor of PBSs.

We compare our 1- and 2-tier schemes to the following allocation algorithms:

- Centralized LP allocation – This obtains an optimal solution that minimizes the number of required FRBs by solving the centralized LP for the 2-tier network as described in Section 3.3.

- Partitioned pico – This scheme partitions FRBs among PBSs and allocates distinct FRBs to the PBSs, using the greedy first fit algorithm.

- Centralized macro-pico allocation – Under this scheme, an MBS does not share FRBs with PBSs. However, PBSs are allowed to share FRBs among themselves. The FRBs are allocated to the PBSs using a (centralized) LP solver.

- Frequency reuse algorithm [32] for conventional homogeneous networks – A user is allocated an FRB shared with all the neighboring BSs if the user's SINR remains above a threshold value even when all the neighboring BSs transmit on the shared FRB. We choose an SINR threshold that provides the best performance for the frequency reuse scheme from the set of threshold values given by {1 dB, 2 dB, . . . , 10 dB}. Roughly speaking, the BSs serve the users that are near their serving BSs with shared FRBs.

- User coloring algorithm [54] – In the frequency reuse algorithm, a user either shares an FRB with all its neighboring BSs or uses an exclusive FRB. However, in the user coloring algorithm, a user might share an FRB with a subset of neighboring BSs, based on interference models in [45, 46].

- FERMI [11] – This 1-tier centralized algorithm distinguishes the users that share frequency resources with all neighboring BSs from those that do not share resources with any neighboring BS. Thus, the type of FRBs are limited to either all shared or exclusive types. This user categorization is based on a threshold on the ratio of throughput of the user on the two different types of frequency resources. The desired number of all shared and exclusive FRBs in each BS is determined based on the category and requirement of each user attached to the BS. The actual number of all shared FRBs at a BS is the minimum among the desired all shared FRBs of the BS and its neighbors.

- LRA [59] – It allows sharing of frequency resources amongst a single MBS and multiple PBSs, for allocation in a 2-tier scenario. A PBS is allowed to use an FRB already used by an MBS user present in a square grid as long as the user is at a sufficient distance (140m) from the PBS. Moreover, each PBS is allocated exclusive FRBs not shared with neighboring PBSs.

3.8.1 1-tier algorithm

We begin with an evaluation of the 1-tier algorithm by examining the number of required FRBs as we increase the number of PBSs. The PBSs are uniformly distributed within a circular area with a radius of 250 meters. Each PBS has twenty (other) users inside its coverage radius, and thus the total number of users in the system increases with the number of PBSs. Each point in Fig. 5.1 is an average of 80 realizations.

Unlike the PBS partitioning scheme that allocates exclusive FRBs not shared with neighboring PBSs, our 1-tier algorithm allows sharing of FRBs among interfering BSs. As a result, the 1-tier algorithm requires at least 26.5% fewer FRBs,

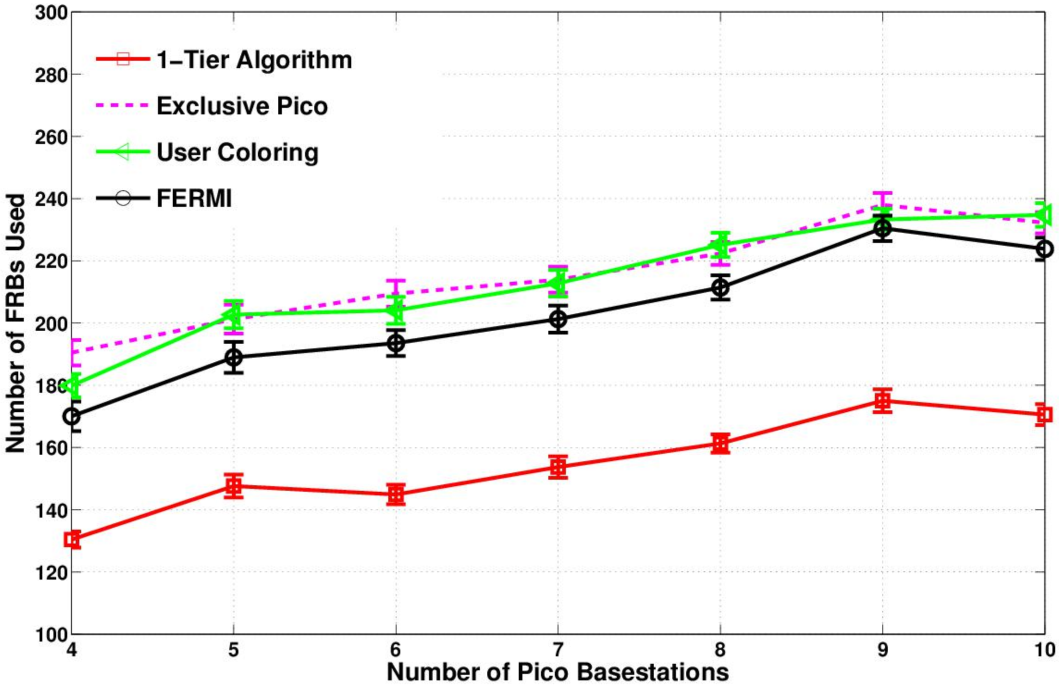


Figure 3.4: Number of required FRBs in a 1-tier network with varying number of PBSs.

demonstrating the advantage of sharing FRBs even amongst interfering PBSs. The user coloring scheme uses on average 37.6% more FRBs than the 1-tier algorithm when there are 7 PBSs. This is because the user coloring scheme uses a fixed SINR threshold to decide the type of sharing for a physical FRB, as opposed to explicitly accounting for the efficiency of FRBs of different types. FERMI requires at least 28% more FRBs compared to the 1-tier algorithm. This shows the advantage of using more general FRB types in the 1-tier algorithm as opposed to the use of just exclusive and all shared FRB types in FERMI.

3.8.2 2-tier algorithm

In the rest of this section, we focus on the 2-tier algorithm. We study its performance as (i) the number of PBSs, (ii) the number of MBS users (iii) the level of interference among the PBSs, and (iv) the number of users are varied with time, and compare the number of required FRBs to that of the centralized LP and other existing schemes.

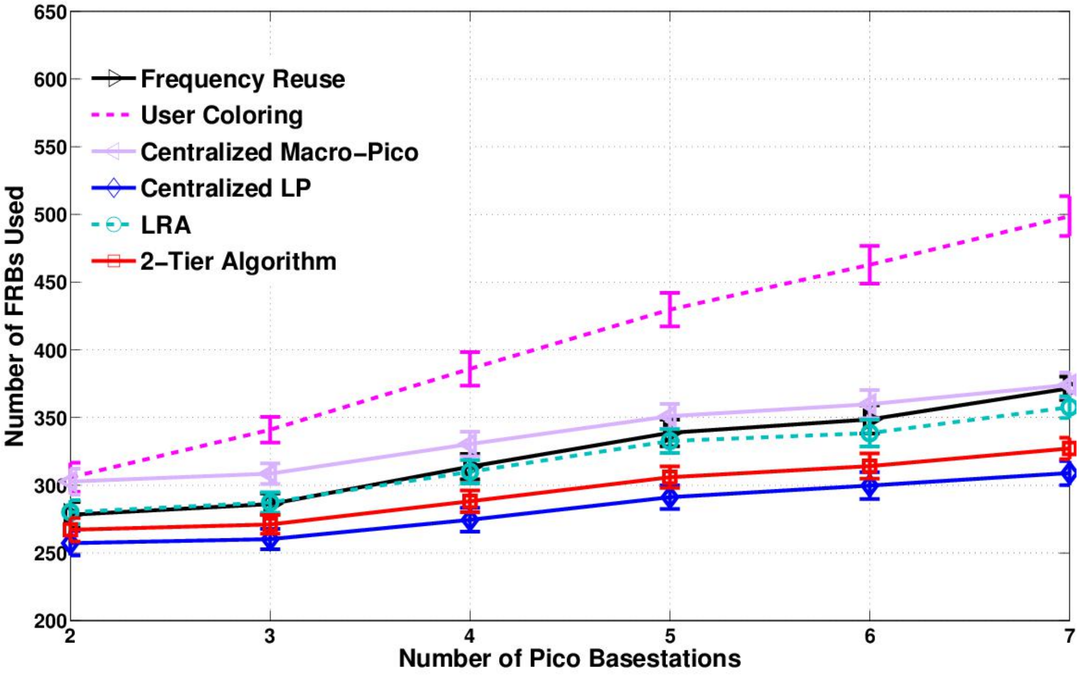


Figure 3.5: Number of required FRBs with varying number of PBSs (2-tier network).

Effects of increasing network size : We begin with a scenario comprising 50 users inside the MBS coverage radius but ouside the PBS coverage areas, and change the number of PBSs. Each PBS has twenty (other) users inside its coverage radius. Thus, the total number of users increases with the number of PBSs. Users are uniformly distributed within the coverage area of their serving BSs, and we compute the average of 80 realizations. To ensure that the MBS users do not belong to the coverage area of PBSs, we generate MBS user positions using the uniform distribution over the MBS coverage area and any user that lies in the coverage area of a PBS is discarded and a new user position is generated until it is not covered by any of the PBSs.

With an increasing number of PBSs, the number of FRBs needed is expected to rise. However, the ability to share amongst PBSs also increases with the number of PBSs. As shown in Fig. 5.3, the 2-tier algorithm performs well (within 6% of the centralized LP for all scenarios). In contrast, the frequency reuse scheme [32] performs up to 13.8% worse than the 2-tier algorithm. The centralized macro-pico allocation scheme also performs worse than the 2-tier algorithm, illustrating the benefits of sharing FRBs between MBS and PBSs. The LRA algorithm performs up to 9.4% worse than the 2-tier algorithm which demonstrates the advantage of sharing amongst PBSs in a 2-tier scenario.

Effects of MBS load : We plot the number of required FRBs as the number of users increases. There are 7 PBSs, each of which contains 20 users in its coverage area. With an increasing number of users placed outside the PBS coverage areas, the fraction of users served by the MBS increases. In each realization, the MBS users are randomly placed according to the uniform distribution on the coverage area of the MBS with a radius of 650 m, and we compute the average of 80 realizations (Fig. 5.2).

Fig. 5.2 shows that as the fraction of users served by the MBS increases, the

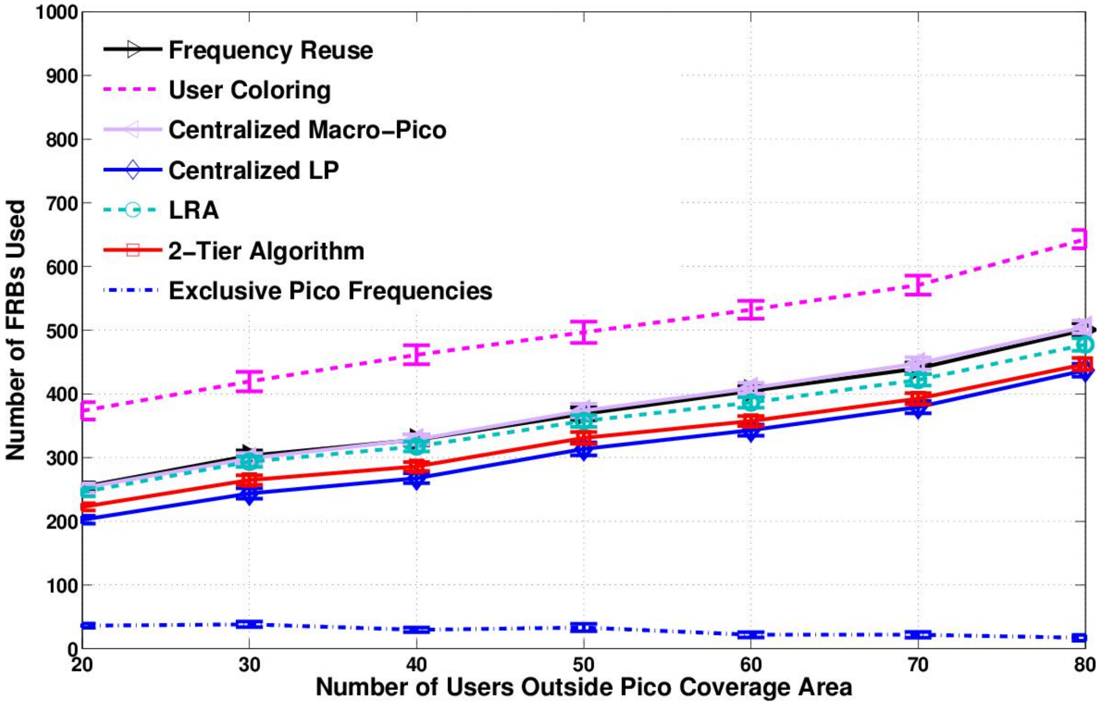


Figure 3.6: Number of required FRBs with varying number of users outside PBS coverage area.

MBS is able to share more FRBs with PBSs. The 2-tier algorithm first allocates the FRBs shared with the MBS to PBSs in the second stage before turning to FRBs exclusive to PBSs. Since the number of users in the coverage area of a PBS is fixed, as shown in Fig. 5.2 (labeled as “Exclusive Pico Frequencies”), the number of FRBs exclusive to PBSs decreases from 16.2% of the total number of FRBs to 3.8% as the number of MBS users rises from 35.8% to 65.8%.

In addition, as the number of MBS users rises, the performance of 2-tier algorithm approaches that of the centralized LP. The 2-tier algorithm is only 2.2% worse than the LP when the fraction of MBS users is 65.8%, while it is 10% higher with 35.8% MBS users. This is because as the MBS serves more users, it can share more FRBs with the PBSs. Hence, it needs fewer FRBs exclusive to PBSs. In contrast, the centralized macro-pico allocation scheme that does not allow MBS and PBSs to share FRBs cannot utilize these sharing opportunities. Consequently, the gap between the 2-tier algorithm and the centralized macro-pico allocation scheme widens with increasing MBS users. Finally, the user coloring [54] and frequency reuse [32] schemes perform consistently worse, as they cannot take advantage of the sharing opportunities. Thus, with sufficiently many MBS users, the proposed distributed 2-tier algorithm approaches the optimal centralized LP.

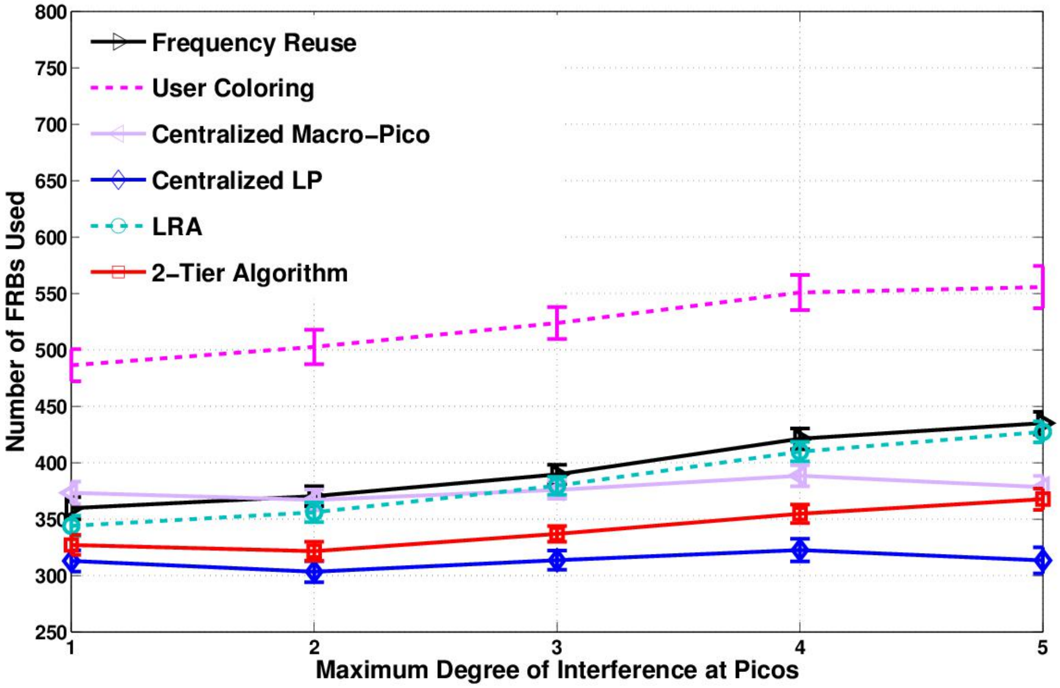


Figure 3.7: Number of required FRBs with varying maximum degree of PBSs in the conflict graph.

Effects of PBS clustering : The 2-tier algorithm performs best when multiple PBSs can take advantage of the FRBs shared with the MBS. In general, as the number of PBS clusters increases, so does the opportunity for FRB sharing. However, how well the PBSs can share the FRBs among themselves depends on the level of interference they cause to each other. Here, we investigate the effects of the level of clustering among the PBSs on the performance of the schemes. We place 7 PBSs in varying configurations such that the level of interference among the PBSs changes.

Fig. 3.7 plots the number of required FRBs as a function of the maximum degree among the PBSs in the conflict graph. The average numbers of clusters for maximum degree of 1, 2, 3, 4 and 5 are 7, 5.7, 4.6, 3.7 and 2.85, respectively. As expected, the number of FRBs required by the 2-tier algorithm is positively correlated with the maximum degree (and hence inversely with the number of clusters). Also, the centralized macro-pico allocation scheme does not perform well when the maximum degree is small since the MBS is not permitted to share FRSs with PBSs.

As the PBSs cluster together, the opportunity to share FRBs with many PBSs diminishes and, consequently, the number of required FRBs goes up. Moreover, as the clustering increases and it becomes more difficult for the PBSs to share FRBs with the MBS, the performance of the 2-tier algorithm approaches that of the centralized macro-pico allocation scheme. In fact, in the worst case where no sharing is possible, the 2-tier algorithm performs worse than the centralized macro-pico allocation scheme. This is due to the fact that the latter employs an optimal allocation among the PBSs by solving the centralized LP, whereas the 2-tier algorithm adopts a distributed heuristic algorithm.

In the above simulations we consider a fixed association bias of 6 dB. But, at a lower association bias, more users will attach to the MBS rather than PBSs. As a result, more macro shared FRBs will be available to the PBSs, which will in turn reduce the number of exclusive pico FRBs. Thus, as we decrease the association bias, the 2-tier algorithm converges to the optimal frequency allocation solution. On the other hand, at a higher association bias, the users will prefer being served by PBSs, which will result in an increase of exclusive pico FRBs. As we increase the bias, the performance of the 2-tier algorithm will converge to the performance of the 1-tier algorithm.

Scenarios with user arrivals and departures : We study how varying the period between optimizations affects the overall performance with dynamic user arrivals and departures, as described in Section 3.7. We evaluate the greedy joint allocation scheme with time-varying user population in a network consisting of two PBSs and an MBS. Users arrive in the system according to a Poisson process with rate 200 users per unit time. The holding times of the users are Pareto distributed with minimum value 0.15 and shape parameter 1.5.

The MBS is located at (0 m, 0 m), and two PBSs are placed at (300 m, 0 m) and (200 m, 0 m). An arriving user is placed in the coverage of each PBS with probability 0.25 according to the uniform distribution. With remaining probability 0.5, the user is placed in the coverage area of the MBS according to the uniform distribution. Each user selects its serving BS using the association scheme explained in Algorithm 2 in Section 3.6.

We plot the minimum number of FRBs we need to meet the throughput requirements of users in 90% of the instances of user arrivals and departures, as a function of the update period of 2-tier algorithm. The reported results are the aver-

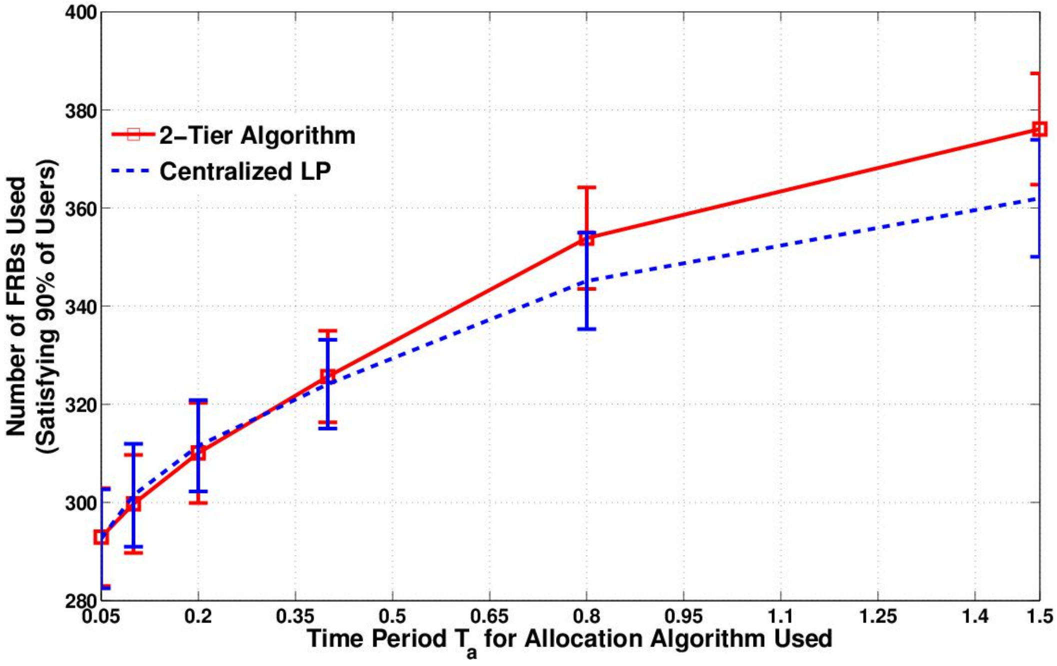


Figure 3.8: Number of required FRBs with varying period of FRB reallocations with greedy joint association.

age of 10 sample paths, each of which has 5000 user arrivals. As shown in Fig. 3.8, as the 2-tier algorithm is executed more frequently, i.e., the update period T_a decreases, fewer FRBs are needed. Hence, it demonstrates a clear tradeoff between the performance of the dynamic allocation and the frequency at which we execute the 2-tier algorithm.

We also compare the performance of our 2-tier algorithm to that of the centralized LP. The plot indicates that the 2-tier algorithm performs comparably to the centralized LP up to T_a=0.4. This observation is due to higher "implicit slack", i.e., FRBs shared with the MBS which are not fully utilized by the PBSs, offered by the 2-tier algorithm compared to the centralized LP. The centralized LP solution tends to share more MBS FRBs with PBSs compared to the 2-tier algorithm and, as a result, has lower implicit slack. An arriving user might attach to a PBS offering sufficient implicit slack without requiring any additional FRBs. However, in case of insufficient implicit slack, the user might require additional FRBs to satisfy its minimum average throughput requirement. As we increase T_a, after a certain point ($T_a = 0.4$ in Fig. 3.8), the optimal allocation of centralized LP outperforms the 2-tier algorithm, as the implicit slack is exhausted by the arriving users in the time interval.

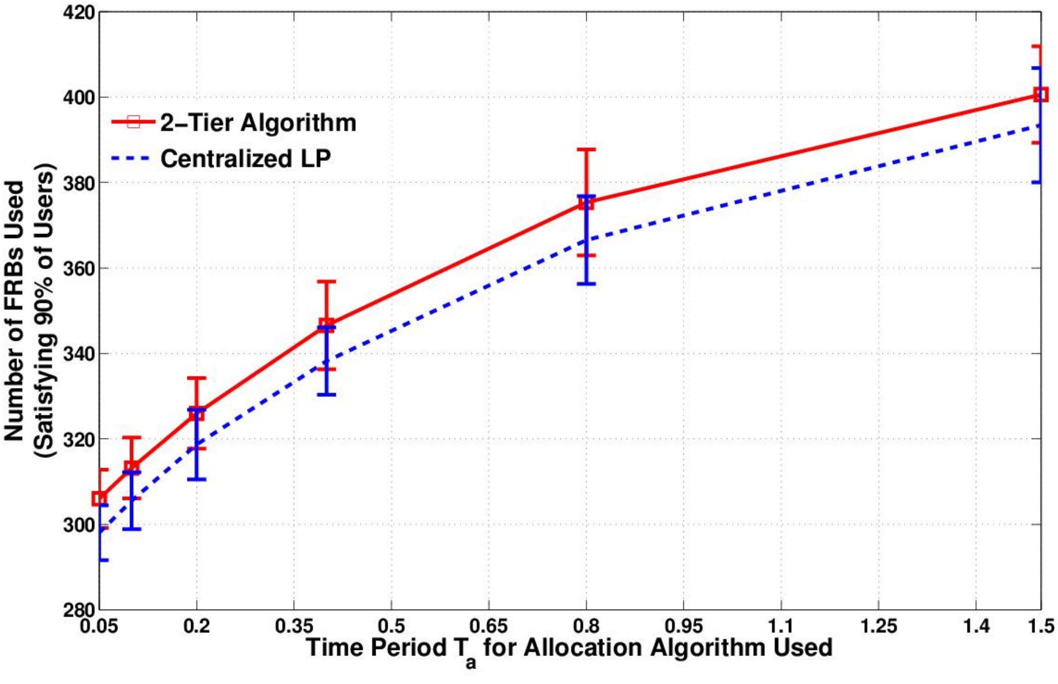


Figure 3.9: Number of required FRBs with varying period of FRB reallocation with a fixed SNR-based association scheme.

Finally, we replace the greedy association scheme with fixed association in which a user selects the BS with the highest SNR with 6 dB bias towards PBSs. Fig. 3.9 plots the number of FRBs required to meet the throughput requirements in 90% of the instances (user arrivals and departures). The plot suggests that the 2-tier algorithm continues to perform on par with the centralized LP. However, because the association scheme no longer takes advantage of implicit slack, in most cases both algorithms require slightly more FRBs than in Fig. 3.8.

Chapter 4: On Optimal Association and Activation Fractions

4.1 Introduction

It is well established by now that all emerging cellular wireless networks will be dense HetNets formed by a multitude of disparate transmission points (TP) deployed in a highly irregular fashion [8]. In this chapter, we propose a RM scheme which decides at the onset of each frame *the user association and the activation fractions (AF).* In particular, our scheme decides *which set of users should each TP serve over that frame such that each user is served by exactly one TP (user association) and how often should each TP transmit over that frame (activation fraction of that TP).*

We adopt α-fairness utility as the system-wide utility which generalizes all popular utility functions [41]. However, even under our more tractable model, the joint optimization of user association and activation fractions is NP-hard. Consequently, we propose an alternating optimization based approach wherein the respective sub-problem is solved at each step. For the discrete user association sub-problem, we completely characterize the underlying set function that needs to be optimized and suggest a simple combinatorial algorithm (referred to as the GLS algorithm) that involves a greedy stage followed by local search improvements. For performance guarantees, we first develop a novel bound on the performance of the greedy stage when used to maximize (minimize) any submodular (supermodular) set function and then derive firm guarantees by specializing this bound to the particular set function of interest. Two key new results that follow from our analysis are: (i) firm (instance independent) guarantees for all $\alpha \leq \frac{\ln(3)}{\ln(2)}$, and (ii) the fact that the same guarantees can be achieved by a restricted (online) greedy algorithm that has a significantly lower complexity. Our results reveal that a simple greedy algorithm can maximize the weighted proportional fairness system utility (i.e., when $\alpha = 1$) to within a constant additive gap of $2\ln(2)$. Next, for the continuous AF optimization sub-problem, we adopt the auxiliary function method and show that it is provably convergent and yields a local optima. Finally, the performance of all our proposed algorithms is compared with baseline schemes via extensive simulations over a HetNet topology generated as per 3GPP LTE guidelines (configuration 4b in [7]). Our results highlight the significant gains that can be achieved in realistic HetNet deployments via the joint optimization.

4.2 Problem statement

Let $\mathcal{U} = \{1, \cdots, K\}$ denote the set of users and let $\mathcal{B}$ denote the set of transmission points (TPs) with cardinality $|\mathcal{B}| = B$. Further, suppose that the time axis is divided into multiple frames, where each frame (of unit duration) consists of several consecutive slots. The set of fast and slow fading coefficients seen by all the K users from all the B TPs on a slot together describe the system state on that slot. The fast and slow fading coefficients seen by each user are assumed to be mutually independent. For simplicity, the fast fading coefficients for each user are assumed to change across slots in an independent identically distributed (i.i.d.) manner, while the slow fading coefficients are assumed to change across frames in an i.i.d. manner. *The choice of the activation fraction for each TP along with the user association for all TPs is made once for each frame to optimize the system utility.* This choice can be based on the slow fading realization in that frame but does not consider any previous such choices. Each TP then independently implements its per-slot scheduling policy over the users associated with it in that frame, where the latter scheduling policy respects the assigned activation fraction and exploits the instantaneous fast fading coefficients seen by the associated users. Consequently, we can suppress the dependence on the frame and slot indices in the following.

Let $\mathcal{U}^{(b)}$, $\forall b \in \mathcal{B}$ denote the set of users associated to TP b and let $\boldsymbol{\rho} = [\rho_b]_{b\in\mathcal{B}}$ denote the activation vector, where $\rho_b \in [0,1]$ denotes the activation fraction assigned to TP b. We systematically derive an average rate that each user can obtain over the frame under the given user association and activation vector. The derived average rates are conservative in terms of optimizing the system utility. We begin by assuming that each TP b allocates a fraction $\gamma_{k,b} \in [0,1]$ of the frame to serve each associated user $k \in \mathcal{U}^{(b)}$, where these fractions are determined at the onset of the frame, i.e., each TP is assumed to adopt a fractional round robin policy. Note that for any choice of the allocated fractions, this fractional round robin policy will be inferior (in terms of optimizing the utility) to an efficient per-slot scheduling policy (cf. [57]) that can exploit instantaneous fading. Next, we assume that the activation fraction is implemented via a Bernoulli activation variable $\mathcal{X}_b$ with $E[\mathcal{X}_b] = \rho_b$ that is i.i.d. across slots in the frame and is independent of all other random variables. Then, the average rate r_k of user $k \in \mathcal{U}^{(b)}$ is given by,

$$\gamma_{k,b}\rho_b\mathbb{E}\left[\log\left(1+\frac{\beta_{k,b}}{1+\sum_{b'\neq b}\beta_{k,b'}\mathcal{X}_{b'}}\right)\right] \tag{4.1}$$

where the the channel gain variables $\{\beta_{k,b}, \beta_{k,b'}\}$ include both fast and slow fading as well as noise normalized transmit powers and the expectation is over the activation variables as well as the fast fading. Upon invoking the fact that the instantaneous rate in (4.1) is convex in the activation variables, which we recall are independent of the fast fading coefficients, we can further lower bound (4.1) to obtain a conservative average rate,

$$r_k = \gamma_{k,b}\underbrace{\rho_b\mathbb{E}\left[\log\left(1+\frac{\beta_{k,b}}{1+\sum_{b'\neq b}\beta_{k,b'}\rho_{b'}}\right)\right]}_{\triangleq R_{k,b}(\boldsymbol{\rho})}, \tag{4.2}$$

where now the expectation is over only the fast fading. Note that r_k in (4.2) depends on the slow fading realization (comprising of the path losses and shadowing factors) over the frame of interest. Letting $\mathbf{r} = [r_1, \cdots, r_K]$ denote the vector of such conservative rates obtained for all the K users over the frame, the achieved system utility is given by

$$\sum_{k\in\mathcal{U}} w_k u(r_k, \alpha), \tag{4.3}$$

where $\alpha \geq 0$ is a tunable fairness fraction and

$$u(r_k, \alpha) = \begin{cases} \frac{r_k^{(1-\alpha)}}{1-\alpha} & \alpha \in (0,1) \\ \log(r_k) & \alpha = 1 \\ -\frac{r_k^{(1-\alpha)}}{\alpha-1} & \alpha > 1 \end{cases} \tag{4.4}$$

and $w_k > 0$ denotes the weight associated with user $k \in \mathcal{U}$. These weights can be used to assign different priorities to different users and we assume that they are normalized, i.e., $\sum_{k \in \mathcal{U}} w_k = 1$. We can now write our optimization problem of interest, which is a mixed optimization problem, as

$$\boxed{\begin{aligned} &\max_{\substack{\boldsymbol{\rho} \in [0,1]^B; x_{k,b} \in \{0,1\}; \\ \gamma_{k,b} \in [0,1] \; \forall \; k,b}} \left\{ \sum_{k \in \mathcal{U}} \sum_{b \in \mathcal{B}} x_{k,b} \left(w_k u(\gamma_{k,b} R_{k,b}(\boldsymbol{\rho}), \alpha) \right) \right\} \\ &\quad \text{s.t.} \sum_{b \in \mathcal{B}} x_{k,b} = 1, \; \forall \; k \in \mathcal{U}; \; \sum_{k \in \mathcal{U}} \gamma_{k,b} \leq 1 \; \forall \; b \in \mathcal{B}. \end{aligned}} \tag{4.5}$$

Note that in (4.5) the binary variable $x_{k,b}$ is one if user k is associated to TP b and zero otherwise, so that the first set of constraints ensures that each user is associated with only one TP. Consequently, $\mathcal{U}^{(b)} \triangleq \{k : x_{k,b} = 1\}$ yields the user set associated with TP b. The variables $\{\gamma_{k,b}\}$ are referred to here as allocation fractions and their sum is upper bounded by unity for each TP, as depicted in the second set of constraints. Note that in (4.5), we enforce $\{\mathcal{U}^{(b)}\}_{b \in \mathcal{B}}$ to be a partition of $\mathcal{U}$. This is meaningful and indeed important since we are targeting short-term optimality by maximizing a system utility independently over each frame. The joint optimization problem in (4.5) is unfortunately intractable, indeed even the user association sub-problem (for any fixed $\boldsymbol{\rho}$) is shown to be NP-hard. Consequently, we develop efficient methods (with meaningful guarantees) for the user-association and the AF optimization sub-problems, and then leverage these methods in an alternating optimization framework to solve the joint problem in (4.5).

4.3 User association

For any fixed $\boldsymbol{\rho}$, we proceed to systematically consider the user-association sub-problem of (4.5) given by

$$\begin{aligned} &\max_{\substack{x_{k,b}\in\{0,1\};\\ \gamma_{k,b}\in[0,1]\ \forall\ k,b}} \left\{ \sum_{k\in\mathcal{U}} \sum_{b\in\mathcal{B}} x_{k,b}\left(w_k u(\gamma_{k,b} R_{k,b}(\boldsymbol{\rho}))\right) \right\} \\ &\text{s.t.} \sum_{b\in\mathcal{B}} x_{k,b} = 1,\ \forall\ k \in \mathcal{U};\ \sum_{k\in\mathcal{U}} \gamma_{k,b} \leq 1,\ \forall\ b \in \mathcal{B}, \end{aligned} \tag{4.6}$$

over three regimes defined by the values that the parameter α can take. We first define a ground set, $\underline{\Omega} = \{(k,b) : k \in \mathcal{U}, b \in \mathcal{B}\}$, that consists of all possible tuples and where each tuple (k,b) denotes an association of user k to transmission point b. We also define the set $\underline{\Omega}^{(b)} = \{(k,b) : k \in \mathcal{U}\}$ for each transmission point $b \in \mathcal{B}$ which consists of all tuples whose transmission point is b, along with the set $\underline{\Omega}_{(k)} = \{(k,b) : b \in \mathcal{B}\}$ for each user k which consists of all tuples whose user is k. Finally, we define a family of sets $\underline{\boldsymbol{\mathcal{I}}}$, as the one which includes each subset of $\underline{\Omega}$ such that the tuples in that subset have mutually distinct users. Formally,

$$\underline{\mathcal{G}} \subseteq \underline{\Omega} : |\underline{\mathcal{G}} \cap \underline{\Omega}_{(k)}| \leq 1,\ \forall\ k \Leftrightarrow \underline{\mathcal{G}} \in \underline{\boldsymbol{\mathcal{I}}}. \tag{4.7}$$

Using the definitions, we see that $\underline{\boldsymbol{\mathcal{I}}}$ is a partition matroid. We start with the regime $\alpha > 1$ and note that upon choosing $\alpha = 2$, (4.6) represents optimization of average system delay, whereas as $\alpha \to \infty$, (4.6) represents association to achieve max-min fairness [41]. Note that for any given user association, i.e, for any given feasible choice of variables $\{x_{k,b}\}$, (4.6) is a continuous optimization problem. Moreover, it is separable across the set of TPs and for each TP $b \in \mathcal{B}$, we have a convex optimization problem over the set of variables $\{\gamma_{k,b}\}$ for $k \in \mathcal{U} : x_{k,b} = 1$. Using K.K.T. conditions it can be verified that for each TP $b \in \mathcal{B}$

$$\begin{aligned} &\max_{\substack{\gamma_{k,b} \in [0,1]\ \forall\ k \\ \sum_{k \in \mathcal{U}} \gamma_{k,b} \leq 1}} \left\{ \sum_{k \in \mathcal{U}} x_{k,b} \left(w_k u(\gamma_{k,b} R_{k,b}(\boldsymbol{\rho})) \right) \right\} = \\ &- \left(\sum_{k \in \mathcal{U}} x_{k,b} \left(w_k \frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{\alpha - 1} \right)^{1/\alpha} \right)^{\alpha} \end{aligned} \tag{4.8}$$

Consequently, upon defining

$$\Theta_k^{(b)}(\alpha) = \left(w_k \frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{\alpha - 1} \right)^{1/\alpha}, \ \forall\ \alpha > 1,$$

(4.6) reduces to the following discrete optimization problem.

$$\min_{\substack{x_{k,b} \in \{0,1\}\ \forall\ k,b \\ \sum_{b \in \mathcal{B}} x_{k,b} = 1\ \forall\ k}} \left\{ \sum_{b \in \mathcal{B}} \left(\sum_{k \in \mathcal{U}} x_{k,b} \Theta_k^{(b)}(\alpha) \right)^{\alpha} \right\}. \tag{4.9}$$

We now consider the case $\alpha \in (0, 1)$. In this case, (4.6) reduces to

$$\max_{\substack{x_{k,b} \in \{0,1\}\ \forall\ k,b \\ \sum_{b \in \mathcal{B}} x_{k,b} = 1\ \forall\ k}} \left\{ \sum_{b \in \mathcal{B}} \left(\sum_{k \in \mathcal{U}} x_{k,b} \Theta_k^{(b)}(\alpha) \right)^{\alpha} \right\}, \tag{4.10}$$

where $\Theta_k^{(b)}(\alpha) = \left(w_k \frac{(R_{k,b}(\boldsymbol{\rho}))^{1-\alpha}}{1-\alpha}\right)^{1/\alpha}$, $\forall\, \alpha \in (0, 1)$.

Recalling the sets $\underline{\Omega}, \underline{\Omega}_{(k)}, \underline{\Omega}^{(b)}$ defined before, we further define the set function

$$g(\underline{\mathcal{G}}, \alpha) = \sum_{b\in\mathcal{B}} \Big(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\underline{\Omega}^{(b)}} \Theta_{k'}^{(b')}(\alpha)\Big)^{\alpha}, \tag{4.11}$$

$\forall\, \underline{\mathcal{G}} \subseteq \underline{\Omega}, \underline{\mathcal{G}} \neq \emptyset$ with $g(\emptyset, \alpha) = 0$, where $\emptyset$ denotes the empty set. The minimization problem in (4.9) can now be reformulated as

$$\min_{\underline{\mathcal{G}}:\underline{\mathcal{G}}\in\underline{\boldsymbol{\mathcal{I}}}\ \&\ |\underline{\mathcal{G}}|=K}\{g(\underline{\mathcal{G}}, \alpha)\}, \tag{4.12}$$

whereas the maximization problem in (4.10) can be re-formulated as

$$\max_{\underline{\mathcal{G}}:\underline{\mathcal{G}}\in\underline{\boldsymbol{\mathcal{I}}}\ \&\ |\underline{\mathcal{G}}|=K}\{g(\underline{\mathcal{G}}, \alpha)\}. \tag{4.13}$$

Similarly, for $\alpha = 1$, (4.6) can be reformulated as in (4.13) but where

$$\begin{aligned} \mathrm{g}(\underline{\mathcal{G}}, 1) = \sum_{(k,b)\in\underline{\mathcal{G}}} & w_k \ln(w_k R_{k,b}(\boldsymbol{\rho})) - \\ & \sum_{b\in\mathcal{B}} \Big(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\underline{\Omega}^{(b)}} w_{k'}\Big) \ln\Big(\sum_{(k',b')\in\underline{\mathcal{G}}\cap\underline{\Omega}^{(b)}} w_{k'}\Big). \end{aligned} \tag{4.14}$$

We offer the following result that completely characterizes these set functions for all α.

Proposition 4.1. *For any $\alpha > 0$ the problem 4.6 is NP-Hard. Further, for any $\alpha > 1$, the set function $g(., \alpha)$ is a normalized, nonnegative and nondecreasing su-*

permodular set function over $\underline{\Omega}$. For any $\alpha \in (0,1)$, the set function $g(.,\alpha)$ is a normalized, nonnegative and nondecreasing submodular set function over $\underline{\Omega}$.

The set function $g(.,1)$ is a normalized submodular set function over $\underline{\Omega}$. A sufficient condition for $g(.,1)$ to be nonnegative is given by

$$\sum_{k\in\mathcal{U}} \frac{1}{R_{k,b}(\boldsymbol{\rho})} \leq 1, \ \ \forall\, b \in \mathcal{B}. \tag{4.15}$$

A stronger condition that suffices for $g(.,1)$ to be nonnegative and nondecreasing is given by

$$R_{k,b}(\boldsymbol{\rho}) \geq \frac{1}{w_k(1-w_k)^{\frac{1-w_k}{w_k}}}, \ \ \forall\, k \in \mathcal{U}, \ b \in \mathcal{B}. \tag{4.16}$$

Proof. Hardness of User Association: The hardness of the user association subproblem for a fixed $\boldsymbol{\rho}$ can be shown via a reduction from the *partition problem.* To show this, we consider the case $\alpha > 1$ and suppose that there is an optimal polynomial time user association algorithm. Further, we restrict ourselves to input instances in which the rates that all users can obtain from two distinct TPs $b1, b2 \in \mathcal{B}$ are identical to one, whereas the rate that each user can obtain from any other TP is zero. Thus, we assume that $R_{k,b}(\boldsymbol{\rho}) = 1, \ \forall\ k \in \mathcal{U}\ \&\ b \in \{b1, b2\}$ while $R_{k,b}(\boldsymbol{\rho}) = 0, \ \forall\, k \in \mathcal{U}\ \&\ b \in \mathcal{B} \setminus \{b1, b2\}$. We allow the user weights to be any input set of positive scalars that sum to 1. Then, the problem in (4.9) simplifies to

$$\min_{\substack{x_{k,b}\in\{0,1\}\ \forall\ k\in\mathcal{U}, b\in\{b1,b2\} \\ \sum_{b\in\{b1,b2\}} x_{k,b}=1\ \forall\ k}} \left\{ \sum_{b\in\{b1,b2\}} \left(\sum_{k\in\mathcal{U}} x_{k,b} w_k^{1/\alpha} \right)^{\alpha} \right\}. \tag{4.17}$$

Then, defining $\hat{z} = \arg\min_{z\in[0,1]}\{z^\alpha + (1-z)^\alpha\}$, it is readily verified that $\hat{z}$ is unique and equal to $1/2$, with $\hat{z}^\alpha + (1-\hat{z})^\alpha = 2^{1-\alpha}$. Letting $W = \sum_{k\in\mathcal{U}} w_k^{1/\alpha}$, this implies that the objective value in (4.17) returned by the optimal polynomial time user association algorithm will be equal to $W^\alpha 2^{1-\alpha}$ if and only if there exists a partition of the set of user weights (each raised to power $1/\alpha$) into two parts that have an identical sum. This in turn implies that the algorithm at hand is an optimal polynomial time algorithm for the NP-complete subset sum problem. Indeed, suppose $\{y_1, \cdots, y_K\}: \;\; y_k > 0, \; \forall k$ is any input set to the latter problem where we need to determine if there exists a partition of that set into two parts of identical sum. Setting $w_k = \frac{y_k^\alpha}{\sum_{i=1}^K y_i^\alpha}, \; \forall \; k = 1, \cdots, K$, we obtain a valid input set of weights for (4.17). Then, from the output of the supposed optimal algorithm at hand, we can immediately determine if there is such a partition for the set $\{\frac{y_k}{(\sum_{i=1}^K y_i^\alpha)^{1/\alpha}}\}_{k=1}^K$ and thus the set $\{y_k\}_{k=1}^K$, which yields the desired contradiction. The same reduction can be established for $\alpha = 1$ as well as $\alpha \in (0, 1)$.

To prove the remaining parts of this proposition, we note that $x^\alpha, \; \forall x \geq 0$ is concave in x when $\alpha \in (0, 1)$ and convex in x when $\alpha > 1$. Then, we note the fact that composition of a nonnegative modular set function with a real valued concave (convex) function yields a submodular (supermodular) set function. Further, the sum of submodular (supermodular) functions is submodular (supermodular) and submodularity as well as supermodularity is preserved under set restriction. Using these facts, we obtain the desired result. Similarly, for $\alpha = 1$ we note that $-x\ln(x), \; \forall x \geq 0$ is concave in x with $\lim_{x\to 0^+} -x\ln(x) = 0$. This fact along with the aforementioned arguments establishes the proof in this case. $\square$

4.3.1 Unified algorithm

In Table 4.1 we propose the GLS Algorithm, which is a simple combinatorial algorithm to solve the problem in (4.6). It considers the respective reformulated versions in (4.12) or (4.13) and comprises of two stages. The first one is the greedy stage (steps 1 to 6). Here in each greedy iteration the feasible tuple (k', b') (with respect to the ones already selected so far) offering the best gain is selected, until no such tuple can be found. In particular, this tuple (k', b') is determined by solving

$$\arg\max_{(k,b)\in\underline{\Omega}:\hat{\underline{\mathcal{G}}}\cup(k,b)\in\underline{\boldsymbol{\mathcal{I}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b),\alpha)-g(\hat{\underline{\mathcal{G}}},\alpha)\},\alpha\leq 1,$$

$$\arg\min_{(k,b)\in\underline{\Omega}:\hat{\underline{\mathcal{G}}}\cup(k,b)\in\underline{\boldsymbol{\mathcal{I}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b),\alpha)-g(\hat{\underline{\mathcal{G}}},\alpha)\},\alpha>1$$

The second stage of Algorithm 4.1 is the *local search improvement* stage and comprises of steps 7 to 13. Here, a feasible pair of tuples is determined in each local search iteration as $(k', b_1), (k', b_2) =$

$$\begin{cases}\arg\max_{\substack{k\in\mathcal{U}\ \&\ b,b'\in\mathcal{B}\\(k,b)\in\breve{\underline{\mathcal{G}}},(k,b')\notin\breve{\underline{\mathcal{G}}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b')\setminus(k,b),\alpha)\},\alpha\leq 1,\\ \arg\min_{\substack{k\in\mathcal{U}\ \&\ b,b'\in\mathcal{B}\\(k,b)\in\breve{\underline{\mathcal{G}}},(k,b')\notin\breve{\underline{\mathcal{G}}}}}\{g(\hat{\underline{\mathcal{G}}}\cup(k,b')\setminus(k,b),\alpha)\},\alpha>1\end{cases}$$

and the corresponding relative improvement is deemed to be better than Δ by checking if

$$g((\breve{\underline{\mathcal{G}}}\cup(k',b_2)\setminus(k',b_1)),\alpha)-g(\breve{\underline{\mathcal{G}}},\alpha)>\Delta\text{sgn}(g(\breve{\underline{\mathcal{G}}},\alpha))g(\breve{\underline{\mathcal{G}}},\alpha),\ \alpha\leq 1,$$

$$g((\breve{\underline{\mathcal{G}}}\cup(k',b_2)\setminus(k',b_1)),\alpha)-g(\breve{\underline{\mathcal{G}}},\alpha)<-\Delta g(\breve{\underline{\mathcal{G}}},\alpha),\ \alpha>1,$$

Table 4.1: **GLS Algorithm**

```
Initialize with $\alpha$, $\Delta \geq 0$, MaxIter $\geq 1$, $\underline{\hat{\mathcal{G}}} = \emptyset$ and $\mathcal{U}' = \mathcal{U}$.
Repeat
    Determine $(k', b')$ as the tuple in $\underline{\Omega}$ which offers the best gain among all tuples $(k, b) \in \underline{\Omega}$ such that $\underline{\hat{\mathcal{G}}} \cup (k, b) \in \underline{\boldsymbol{\mathcal{I}}}$.
    Update $\underline{\hat{\mathcal{G}}} = \underline{\hat{\mathcal{G}}} \cup (k', b')$ and $\mathcal{U}' = \mathcal{U}' \setminus \{k'\}$
Until $\mathcal{U}' = \emptyset$.
Set $\underline{\breve{\mathcal{G}}} = \underline{\hat{\mathcal{G}}}$, Iter $= 0$.
Repeat
    Increment Iter $=$ Iter $+ 1$.
    Find a pair of tuples: $(k', b_1) \in \underline{\breve{\mathcal{G}}}$ and $(k', b_2) \in \underline{\Omega} \setminus \underline{\breve{\mathcal{G}}}$ such that the relative improvement upon swapping $(k', b_1) \in \underline{\breve{\mathcal{G}}}$ with $(k', b_2)$ is better than $\Delta$.
    If such a pair exists then
        Update $\underline{\breve{\mathcal{G}}} = \underline{\breve{\mathcal{G}}} \cup (k', b_2) \setminus (k', b_1)$.
    End If
Until no such pair exists or Iter $=$ MaxIter.
Output $\underline{\breve{\mathcal{G}}}$.
```

where $\text{sgn}(x) = 1,\ \forall x \geq 0$ and -1 otherwise.

We now proceed to analyze the performance of our proposed GLS algorithm. Our goal is to bound the gap (by obtaining easily computable bounds) between the optimal system utility value and the one returned by the GLS Algorithm. Towards this end, let $\underline{\mathcal{G}}^{\text{opt}}$ denote the optimal solution to the problem in (4.12) for $\alpha > 1$ or (4.13) for $\alpha \in (0, 1]$, and let $\underline{\breve{\mathcal{G}}}, \underline{\hat{\mathcal{G}}}$ denote the counterparts obtained by our algorithm as the final output and after the greedy stage, respectively. We will first analyze the performance of the greedy first stage. To do so, we derive new bounds that relate the optimal solution to that returned by the greedy stage. These bounds are in fact applicable to arbitrary (not necessarily nonnegative or nondecreasing) submodular or supermodular set functions.

Proposition 4.2. *For any given α, the greedy stage yields an output $\hat{\underline{\mathcal{G}}}$ such that*

$$g(\hat{\underline{\mathcal{G}}}, \alpha) \geq g(\underline{\mathcal{G}}^{\mathrm{opt}} \cup \hat{\underline{\mathcal{G}}}, \alpha) - g(\hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\mathrm{opt}}, \alpha), \;\; \forall\, \alpha \leq 1$$
$$g(\hat{\underline{\mathcal{G}}}, \alpha) \leq g(\underline{\mathcal{G}}^{\mathrm{opt}} \cup \hat{\underline{\mathcal{G}}}, \alpha) - g(\hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\mathrm{opt}}, \alpha), \;\; \forall\, \alpha > 1.$$

Proof. For notational convenience let us denote a tuple as $\underline{e} = (k, b)$. We expand $\hat{\underline{\mathcal{G}}}$ as $\hat{\underline{\mathcal{G}}} = \{\hat{\underline{e}}_1, \hat{\underline{e}}_2, \cdots, \hat{\underline{e}}_K\}$ where $\hat{\underline{e}}_i$ denotes the tuple added at the i^{th} greedy step and let $\delta_i,\ i = 1, \cdots, K$ denote the associated incremental gain. Further, we define the sets $\hat{\underline{\mathcal{G}}}_i = \{\hat{\underline{e}}_1, \hat{\underline{e}}_2, \cdots, \hat{\underline{e}}_i\},\ \forall\ \ i = 1, \cdots, K$ with $\hat{\underline{\mathcal{G}}}_0 = \phi$. Then, note that both $\underline{\mathcal{G}}^{\mathrm{opt}}, \hat{\underline{\mathcal{G}}} \in \underline{\boldsymbol{\mathcal{I}}}$ and are maximal members in $\underline{\boldsymbol{\mathcal{I}}}$, i.e., $|\underline{\mathcal{G}}^{\mathrm{opt}}| = |\underline{\mathcal{G}}| = K$. Invoking a result on maximal members in a matroid (cf. [40]), we can deduce that without loss of generality, we can expand $\underline{\mathcal{G}}^{\mathrm{opt}} = \{\underline{e}_1^{\mathrm{opt}}, \underline{e}_2^{\mathrm{opt}}, \cdots, \underline{e}_K^{\mathrm{opt}}\}$ such that for each $i \in \{1, \cdots, K\}$,

$$\text{Either } \underline{e}_i^{\mathrm{opt}} = \hat{\underline{e}}_i, \;\text{ or}$$
$$\underline{e}_i^{\mathrm{opt}} \notin \hat{\underline{\mathcal{G}}} \;\&\; (\hat{\underline{\mathcal{G}}} \setminus \hat{\underline{e}}_i) \cup \underline{e}_i^{\mathrm{opt}} \in \underline{\boldsymbol{\mathcal{I}}}. \tag{4.18}$$

Then, letting $\tilde{\underline{\mathcal{G}}} \triangleq \hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\mathrm{opt}}$ we have the chain of inequalities (4.19) given below which yields the desired result. In (4.19) the first inequality follows from submodularity of $g(., \alpha)$ and the fact that for each $i : \hat{\underline{e}}_i \in \hat{\underline{\mathcal{G}}} \cap \underline{\mathcal{G}}^{\mathrm{opt}}$, $\hat{\underline{\mathcal{G}}}_{i-1} \subseteq \hat{\underline{\mathcal{G}}}_{i-1} \cup \tilde{\underline{\mathcal{G}}}$ and $\hat{\underline{e}}_i \notin \hat{\underline{\mathcal{G}}}_{i-1} \cup \tilde{\underline{\mathcal{G}}}$. The second inequality follows from (4.18) along with the fact that for each $i : \underline{e}_i^{\mathrm{opt}} \notin \hat{\underline{\mathcal{G}}}$, the greedy algorithm would have considered $\underline{e}_i^{\mathrm{opt}}$ but chose $\hat{\underline{e}}_i$ instead since the latter offered a larger incremental gain. The third inequality also

$$\begin{aligned} g(\underline{\hat{\mathcal{G}}}, \alpha) =& \sum_{i=1}^{K} \delta_i = \sum_{i:\underline{\hat{e}}_i \in \underline{\hat{\mathcal{G}}} \cap \underline{\mathcal{G}}^{\text{opt}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\hat{e}}_i, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1}, \alpha)) + \sum_{i:\underline{\hat{e}}_i \in \underline{\tilde{\mathcal{G}}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\hat{e}}_i, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1}, \alpha)) \\ &\geq \sum_{i:\underline{\hat{e}}_i \in \underline{\hat{\mathcal{G}}} \cap \underline{\mathcal{G}}^{\text{opt}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\tilde{\mathcal{G}}} \cup \underline{\hat{e}}_i, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\tilde{\mathcal{G}}}, \alpha)) + \sum_{i:\underline{\hat{e}}_i \in \underline{\tilde{\mathcal{G}}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\hat{e}}_i, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1}, \alpha)) \\ &= g(\underline{\hat{\mathcal{G}}}, \alpha) - g(\underline{\tilde{\mathcal{G}}}, \alpha) + \sum_{i:\underline{\hat{e}}_i \in \underline{\tilde{\mathcal{G}}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{\hat{e}}_i, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1}, \alpha)) \\ &\geq g(\underline{\hat{\mathcal{G}}}, \alpha) - g(\underline{\tilde{\mathcal{G}}}, \alpha) + \sum_{i:\underline{e}_i^{\text{opt}} \notin \underline{\hat{\mathcal{G}}}} (g(\underline{\hat{\mathcal{G}}}_{i-1} \cup \underline{e}_i^{\text{opt}}, \alpha) - g(\underline{\hat{\mathcal{G}}}_{i-1}, \alpha)) \\ &\geq g(\underline{\hat{\mathcal{G}}}, \alpha) - g(\underline{\tilde{\mathcal{G}}}, \alpha) + \sum_{i:\underline{e}_i^{\text{opt}} \notin \underline{\hat{\mathcal{G}}}} (g(\underline{\hat{\mathcal{G}}} \cup \underline{e}_i^{\text{opt}}, \alpha) - g(\underline{\hat{\mathcal{G}}}, \alpha)) \\ &\geq g(\underline{\hat{\mathcal{G}}}, \alpha) - g(\underline{\tilde{\mathcal{G}}}, \alpha) + g(\underline{\mathcal{G}}^{\text{opt}} \cup \underline{\hat{\mathcal{G}}}, \alpha) - g(\underline{\hat{\mathcal{G}}}, \alpha), \end{aligned} \tag{4.19}$$

follows from submodularity of $g(., \alpha)$ and the fact that for each $i : \underline{e}_i^{\text{opt}} \notin \underline{\hat{\mathcal{G}}}$ we have $\underline{\hat{\mathcal{G}}}_{i-1} \subseteq \underline{\hat{\mathcal{G}}}$, and the final inequality also follows from submodularity of $g(., \alpha)$. Note that none of the steps require $g(., \alpha)$ to be a non-negative set function or that the incremental gains should be nonnegative. The second relation in the proposition can be proved in an analogous fashion. □

We now demonstrate the utility of the bounds in Proposition 4.2 by specializing them to the set functions of interest to us in (4.11) and (4.14).

Proposition 4.3. *For any given α, the greedy stage yields an output $\underline{\hat{\mathcal{G}}}$ such that*

$$\begin{aligned} &g(\underline{\hat{\mathcal{G}}}, \alpha) \geq g(\underline{\mathcal{G}}^{\text{opt}}, \alpha)/2, \ \ \forall\, \alpha \in (0,1), \\ &g(\underline{\hat{\mathcal{G}}}, \alpha) \geq g(\underline{\mathcal{G}}^{\text{opt}}, \alpha) - 2\ln(2), \ \ \forall\, \alpha = 1, \\ &(3 - 2^{\alpha}) g(\underline{\hat{\mathcal{G}}}, \alpha) \leq g(\underline{\mathcal{G}}^{\text{opt}}, \alpha), \ \ \forall\, \alpha > 1. \end{aligned} \tag{4.20}$$

Proof. For $\alpha \in (0,1)$, since $g(., \alpha)$ is submodular and nondecreasing, we can readily obtain (4.20) from Proposition 4.2 by observing that $g(\underline{\mathcal{G}}^{\text{opt}} \cup \hat{\underline{\mathcal{G}}}, \alpha) \geq g(\underline{\mathcal{G}}^{\text{opt}}, \alpha)$ and $g(\hat{\underline{\mathcal{G}}}, \alpha) \geq g(\hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\text{opt}}, \alpha)$. Note that (4.20) is the classical result derived earlier [47]. For $\alpha = 1$, the result in (4.20) is novel and thus more interesting. To prove (4.20), we first rewrite the bound in Proposition 4.2 as

$$\begin{aligned} g(\hat{\underline{\mathcal{G}}}, 1) \geq g(\underline{\mathcal{G}}^{\text{opt}}, 1) + g(\underline{\mathcal{G}}^{\text{opt}} \cup \hat{\underline{\mathcal{G}}}, 1) - g(\underline{\mathcal{G}}^{\text{opt}}, 1) \\ - g(\hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\text{opt}}, 1). \end{aligned} \tag{4.21}$$

Then, recall from (4.14) that $g(., 1)$ is the sum of a modular function and a submodular function where the latter depends only on the user weights, and the sum of these weights across all users is unity. Consequently, we can infer that

$$\begin{aligned} & g(\underline{\mathcal{G}}^{\text{opt}} \cup \hat{\underline{\mathcal{G}}}, 1) - g(\underline{\mathcal{G}}^{\text{opt}}, 1) - g(\hat{\underline{\mathcal{G}}} \setminus \underline{\mathcal{G}}^{\text{opt}}, 1) \\ = & - \sum_b (x_b + y_b) \ln(x_b + y_b) + \sum_b (z_b + y_b) \ln(z_b + y_b) \\ & + \sum_b (x_b - z_b) \ln(x_b - z_b) \end{aligned} \tag{4.22}$$

where x_b is the sum of weights of users associated to TP b by the greedy solution (and hence is known), $y_b + z_b$ is the sum of weights of users associated to TP b by the optimal solution and z_b is the sum of weights of users associated to TP b by both the greedy and the optimal solutions. Note further that $\sum_b x_b = \sum_b (y_b + z_b) = 1$.

Combining (4.22) with (4.21) we can obtain the following specialized bound,

$$\begin{aligned}g(\underline{\hat{\mathcal{G}}},1) \geq & g(\underline{\mathcal{G}}^{\text{opt}},1) \\ & + \min_{\substack{y_b,z_b\geq 0; z_b\leq x_b\ \forall b\\ \sum_b(y_b+z_b)=1}}\{-\sum_b (x_b+y_b)\ln(x_b+y_b) \\ & + \sum_b (z_b+y_b)\ln(z_b+y_b) \\ & + \sum_b (x_b-z_b)\ln(x_b-z_b)\}.\end{aligned} \tag{4.23}$$

Then, by using the necessary K.K.T. conditions for the optimization problem in the RHS of (4.23), it can be shown that the minima is attained at $y_b = x_b$ & $z_b = 0$, $\forall\, b$ so that

$$\begin{aligned}&\min_{\substack{y_b,z_b\geq 0; z_b\leq x_b\ \forall b\\ \sum_b(y_b+z_b)=1}}\{-\sum_b (x_b+y_b)\ln(x_b+y_b) \\ &+ \sum_b (z_b+y_b)\ln(z_b+y_b) + \sum_b (x_b-z_b)\ln(x_b-z_b)\} = -2\ln(2).\end{aligned} \tag{4.24}$$

This proves the result in (4.20). Next, we consider $\alpha > 1$ and specialize the bound in Proposition 2 as

$$\begin{aligned}g(\underline{\hat{\mathcal{G}}},\alpha) \leq g(\underline{\mathcal{G}}^{\text{opt}},\alpha) + g(\underline{\mathcal{G}}^{\text{opt}}\cup\underline{\hat{\mathcal{G}}},\alpha) - g(\underline{\mathcal{G}}^{\text{opt}},\alpha) - g(\underline{\hat{\mathcal{G}}}\setminus\underline{\mathcal{G}}^{\text{opt}},\alpha) \\ = g(\underline{\mathcal{G}}^{\text{opt}},\alpha) + \sum_b ((v_b+t_b)^\alpha - (v_b+u_b)^\alpha \\ - (t_b-u_b)^\alpha),\end{aligned} \tag{4.25}$$

where now t_b is the sum of gains of all users associated to TP b by the greedy solution (i.e., sum of $\Theta_k^{(b)}(\alpha)$ in (4.9) for all tuples in $\hat{\underline{\mathcal{G}}} \cap \Omega^{(b)}$ and hence is known) so that $g(\hat{\underline{\mathcal{G}}}, \alpha) = \sum_b t_b^\alpha$. $v_b + u_b$ is the sum of gains of all users associated to TP b by the optimal solution and u_b is the sum of gains of all users associated to TP b by both the greedy and the optimal solutions. Clearly, we can further bound

$$\begin{aligned} g(\hat{\underline{\mathcal{G}}}, \alpha) \leq g(\underline{\mathcal{G}}^{\text{opt}}, \alpha) + & \max_{\substack{v_b, u_b \geq 0; u_b \leq t_b \ \forall b \\ \sum_b (v_b + u_b)^\alpha \leq g(\hat{\underline{\mathcal{G}}}, \alpha)}} \{\sum_b ((v_b + t_b)^\alpha \\ & - (v_b + u_b)^\alpha - (t_b - u_b)^\alpha)\} \end{aligned} \tag{4.26}$$

Again invoking the necessary K.K.T. conditions for the optimization problem in the RHS of (4.26), it can be shown that the maxima is attained at $v_b = t_b$ & $u_b = 0$, $\forall\, b$ so that

$$\begin{aligned} & \max_{\substack{v_b, u_b \geq 0; u_b \leq t_b \ \forall b \\ \sum_b (v_b + u_b)^\alpha \leq g(\hat{\underline{\mathcal{G}}}, \alpha)}} \{\sum_b ((v_b + t_b)^\alpha - (v_b + u_b)^\alpha - (t_b - u_b)^\alpha)\} \\ & = (2^\alpha - 2) g(\hat{\underline{\mathcal{G}}}, \alpha) \end{aligned} \tag{4.27}$$

This then proves the result in (4.20) □

Remark 4.1. *Note that the last bound in Proposition 4.3 is meaningful in the regime $\alpha \in \left(1, \frac{\ln(3)}{\ln(2)}\right)$ since then $3 - 2^\alpha > 0$. As a result, we can deduce that for all $\alpha \in \left(0, \frac{\ln(3)}{\ln(2)}\right)$, the greedy stage of the GLS algorithm itself provides firm (instance independent) guarantees. However, as α is progressively increased, the performance of the greedy stage is increasingly degraded compared to the optimal and the local search stage of the GLS algorithm becomes increasingly important.*

For nonnegative nondecreasing submodular set functions, which we recall does

Table 4.2: **Restricted Greedy Algorithm**

```
Initialize with any ordering $\pi(.)$ defined on $\mathcal{U}$ and $\underline{\hat{\mathcal{G}}}^{\text{rg}} = \emptyset$.
For $k = 1$ to $K$,
    Determine $(\pi(k), b')$ as the tuple in $\underline{\Omega}$ which offers the best incremental gain among all tuples $(\pi(k), b) \in \underline{\Omega}$.
    Update $\underline{\hat{\mathcal{G}}}^{\text{rg}} = \underline{\hat{\mathcal{G}}}^{\text{rg}} \cup (\pi(k), b')$.
End For.
Output $\underline{\hat{\mathcal{G}}}^{\text{rg}}$.
```

not hold for our set functions when $\alpha \geq 1$, a somewhat lesser known result is that a restricted version of the greedy algorithm can also yield identical constant factor approximation [33]. We next establish a similar result with respect to the bounds in Propositions 4.2 and 4.3. In particular, we detail the Restricted Greedy Algorithm in Table 4.2 and prove its guarantee in the following proposition.

Proposition 4.4. *For any given ordering $\pi(.)$, the Restricted Greedy Algorithm yields a solution that also satisfies the bounds in Proposition 4.2 for all α. Thus, the solution of the Restricted Greedy Algorithm also satisfies the bounds in Proposition 4.3 for all α and hence yields the same firm guarantees for all $\alpha \in \left(0, \frac{\ln(3)}{\ln(2)}\right)$.*

Proof. We expand the solution yielded by the Restricted Greedy Algorithm as $\underline{\hat{\mathcal{G}}}^{\text{rg}} = \{\underline{\hat{e}}^{\text{rg}}_{\pi(1)}, \underline{\hat{e}}^{\text{rg}}_{\pi(2)}, \cdots, \underline{\hat{e}}^{\text{rg}}_{\pi(K)}\}$ where $\underline{\hat{e}}^{\text{rg}}_{\pi(i)}$ denotes the tuple added at the i^{th} step as per the ordering $\pi(.)$. Then, all the arguments in the proof of Proposition 4.2 go through even upon replacing $\underline{\hat{\mathcal{G}}}$ with $\underline{\hat{\mathcal{G}}}^{\text{rg}}$ and $\underline{\hat{e}}_i$ with $\underline{\hat{e}}^{\text{rg}}_{\pi(i)}$, $\forall\ i$. The key point to note here is that we do not require the incremental gains obtained across the steps to be ordered. In other words, we do not use the fact that the incremental gains obtained during the greedy stage of the GLS algorithm are ordered as $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_K$ whereas no such ordering is ensured for those obtained during the restricted greedy algorithm. □

Notice here that the restricted greedy algorithm can be viewed as an *online* algorithm by setting the order $\pi(.)$ to be the order in which users arrive. Then, for any given set of users, our result implies that the online greedy and the offline greedy (i.e., the greedy stage of the GLS algorithm) yield the same guarantees.

4.4 AF optimization

The association scheme described in the previous section determines $\mathcal{U}^{(b)}$, the set of users associated to TP b for all $b \in \mathcal{B}$. In this section, for a given user association, we seek to determine ρ_b for each b so as to optimize the system utility over different α regimes. Our approach is based on the auxiliary function method. Such a method has been used for precoder optimization over the single-cell downlink in [24] and over the multi-cell downlink in [60].

$\alpha > 1:$ The AF optimization problem in this regime is given by

$$\min_{\boldsymbol{\rho}\in[0,1]^B} \left\{\sum_{b\in B} \left(\sum_{k\in\mathcal{U}^{(b)}} \tilde{w}_k/(R_{k,b}(\boldsymbol{\rho}))^{1-1/\alpha}\right)^{\alpha}\right\} \tag{4.28}$$

where $\tilde{w}_k = (\frac{w_k}{\alpha-1})^{1/\alpha}$ and $R_{k,b}(\boldsymbol{\rho})$ is given by (4.2). Note that optimization over the activation vector $\boldsymbol{\rho}$ seems intractable since $R_{k,b}(\boldsymbol{\rho})$ involves an expectation of a nonlinear function of $\boldsymbol{\rho}$. Nevertheless, we demonstrate that a local optima can be obtained via a provably convergent *auxiliary function method.* We let $\boldsymbol{\beta}_k = \{\beta_{k,b}\}\ \forall b \in B$ denote the vector containing all fading coefficients pertaining to user

k on any slot. Then, we introduce auxiliary variables $g_{k,b}(\boldsymbol{\beta}_k)$ for each vector $\boldsymbol{\beta}_k$ for each user $k \in \mathcal{U}^{(b)}$ for each TP b. Using $g_{k,b}(\boldsymbol{\beta}_k)$ as a filter at user k to detect the signal transmitted from TP b over that slot, the mean squared error (MSE), $e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})$, is given by

$$e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) = \left| g_{k,b}(\boldsymbol{\beta}_k)\sqrt{\beta_{k,b}} - 1 \right|^2 + |g_{k,b}(\boldsymbol{\beta}_k)|^2 + |g_{k,b}(\boldsymbol{\beta}_k)|^2 \sum_{b' \neq b} \beta_{k,b'} \rho_{b'} \quad (4.29)$$

Using the mutual information and MSE identity and introducing more auxiliary variables (cf. [24]), we have

$$R_{k,b}(\boldsymbol{\rho}) = \rho_b \mathbb{E}[\max_{g_{k,b}(\boldsymbol{\beta}_k), s_{k,b}(\boldsymbol{\beta}_k) \geq 0} \{1 - s_{k,b}(\boldsymbol{\beta}_k) e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))\}](4.30)$$

The solution of each inner maximization problem in (4.30) is obtained by setting $g_{k,b}(\boldsymbol{\beta}_k)$ to be the MMSE filter $\hat{g}_{k,b}(\boldsymbol{\beta}_k)$ with $s_{k,b}(\boldsymbol{\beta}_k) = \hat{s}_{k,b}(\boldsymbol{\beta}_k) = 1/\hat{e}_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})$, where $\hat{e}_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) = e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) \mid_{g_{k,b}(\boldsymbol{\beta}_k) = \hat{g}_{k,b}(\boldsymbol{\beta}_k)}$. Using (4.30), the problem in (4.28) (for the given association) can be reformulated as the following optimization problem over variables $\boldsymbol{\rho}$, $\boldsymbol{s} = \{s_{k,b}(\boldsymbol{\beta}_k)\}, \boldsymbol{g} = \{g_{k,b}(\boldsymbol{\beta}_k)\}$ $\forall \boldsymbol{\beta}_k, k \in \mathcal{U}^{(b)}, b \in B$.

$$\min_{\boldsymbol{\rho} \in [\mathbf{0}, \mathbf{1}], \mathbf{g} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{1}} \left\{ \sum_{b \in B} \left(\sum_{k \in \mathcal{U}^{(b)}} \frac{\tilde{w}_k}{(\rho_b \mathbb{E}[1 - s_{k,b}(\boldsymbol{\beta}_k) e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))])^{1-1/\alpha}} \right)^{\alpha} \right\}$$

Note that for a fixed $\boldsymbol{\rho}$, (4.31) can be optimized over $\mathbf{s}, \mathbf{g}$ via the closed form expressions given above. On the other hand, for fixed $\mathbf{s}, \mathbf{g}$ to optimize (4.31) over $\boldsymbol{\rho}$, we introduce additional variables $\boldsymbol{z} = \{z_b\}$ $\forall b \in B$ and $\boldsymbol{t} = \{t_{k,b}\}$, $\forall$ $k \in \mathcal{U}^{(b)}, b \in B$

and express the reduced problem in (4.31) as

$$\begin{aligned} &\min_{\boldsymbol{\rho}\in[\mathbf{0},\mathbf{1}],\mathbf{z}\geq\mathbf{0},\mathbf{t}\geq\mathbf{0}} \left\{\sum_{b\in B} z_b^{\alpha}\right\} \\ &\text{subject to} \\ &z_b \geq \sum_{k\in\mathcal{U}^{(b)}} \tilde{w}_k t_{k,b}^{1/\alpha-1} \quad \forall k, b \\ &t_{k,b} \leq \rho_b \mathbb{E}[1 - s_{k,b}(\boldsymbol{\beta}_k) e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho}) + \log(s_{k,b}(\boldsymbol{\beta}_k))] \quad \forall k, b \end{aligned} \tag{4.31}$$

Notice that (4.31) can in turn be re-written as

$$\begin{aligned} &\min_{\boldsymbol{\rho}\in[\mathbf{0},\mathbf{1}],\mathbf{z}\geq\mathbf{0},\mathbf{t}\geq\mathbf{0}} \{\sum_{b\in B} z_b^{\alpha}\} \\ &\text{subject to} \\ &\sum_{k} z_b^{-1} \tilde{w}_k t_{k,b}^{1/\alpha-1} \leq 1 \quad \forall k, b \\ &\frac{t_{k,b}\rho_b^{-1} + \mathbb{E}[s_{k,b}(\boldsymbol{\beta}_k) e_{k,b}(\boldsymbol{\beta}_k, \boldsymbol{\rho})]}{1 + \mathbb{E}[\log(s_{k,b}(\boldsymbol{\beta}_k))]} \leq 1 \quad \forall k, b \end{aligned} \tag{4.32}$$

The problem in (4.32) is a geometric program (GP) since all constraints are inequalities involving posynomials. Thus, we can repeat the following two steps until convergence.

(a) Fix $\boldsymbol{\rho}$ and minimize (4.31) over $\boldsymbol{s}, \boldsymbol{g}$ using closed form solution of (4.30).

(b) Fix $\boldsymbol{s}, \boldsymbol{g}$ and minimize (4.31) over $\boldsymbol{\rho}$ by solving equivalent GP in (4.32).

Note that in the described auxiliary function method we have a monotonic improvement in the objective value of (4.31) so that convergence is guaranteed. Using standard tools we can also deduce the additional result stated in the following proposition.

Proposition 4.5. *The auxiliary function method provably converges. Each cluster point of the obtained sequence of activation vectors satisfies the KKT conditions of (4.28).*

Regarding the practical implementation of the proposed auxiliary function method, we first note that the GP in (4.32) can be efficiently solved. Further, we can approximate the expectations in (4.31) and (4.32) by using a set of generated fading coefficients $\{\boldsymbol{\beta}_k\}$ $\forall k$. As a further simplification we can use the following approximate expression for the average rate

$$\tilde{R}_{k,b}(\boldsymbol{\rho}) = \rho_b \log\left(1 + \frac{\tilde{\beta}_{k,b}}{1 + \sum_{b' \neq b} \tilde{\beta}_{k,b'} \rho_{b'}}\right) \tag{4.33}$$

where each $\tilde{\beta}_{k',b'}$ is obtained by averaging $\beta_{k',b'}$ over the fast fading. Analogous solutions for the AF optimization problem in the regime $\alpha = 1$ and the regime $\alpha < 1$ are derived.

4.5 Joint association and AF optimization

We propose two joint association and AF optimization algorithms for solving the problem in (4.5). These algorithms follow an alternating optimization approach where user association (stage 1) and AF (stage 2) are optimized in an alternating fashion. Fig. 4.1 shows a block-level decomposition. The first algorithm is the Joint GLS-AF algorithm, in which we first run the GLS algorithm (Algorithm in

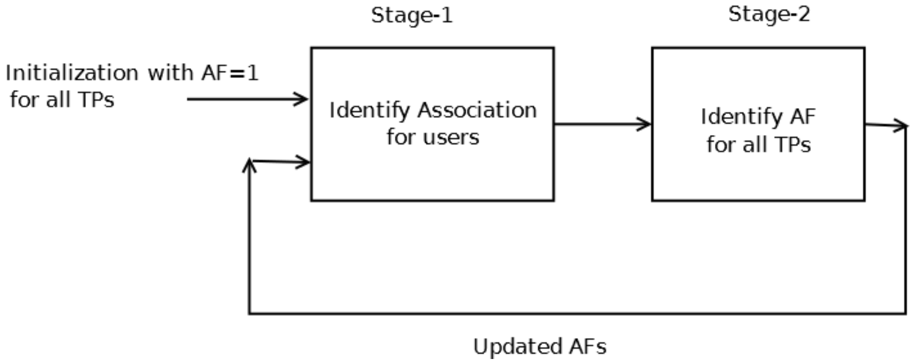

Stage-1
Stage-2
Initialization with AF=1 for all TPs
Identify Association for users
Identify AF for all TPs
Updated AFs

Figure 4.1: Joint Association and AF optimization block diagram

Table 4.1) and use the obtained association in our AF optimization algorithm in Section 4.4. We repeat the following two steps until the benefit in terms of the α-fairness system utility falls below a threshold.

(a) Stage1–Fix $\boldsymbol{\rho}$ and use GLS algorithm to calculate the user association.

(b) Stage2–Fix the association and optimize over $\boldsymbol{\rho}$ using the auxiliary function method given in Section 4.4.

In the Joint GLS-AF algorithm, TPs that do not serve any user in any one iteration will be discarded in all subsequent iterations. To overcome this limitation we propose the Joint Relaxed Association AF (Joint RA-AF) algorithm. To obtain the association, this algorithm in stage 1 solves the convex optimization problem obtained by relaxing variables $x_{k,b}$, $\forall\, k, b$ in (4.9) or (4.10) to be continuous variables in $[0, 1]$. In this solution, a user k can have $x_{k,b}$ nonzero for more than one TP b. In stage 2, the algorithm fixes $x_{k,b}$ for all k, b and optimizes the AF. To do so, it uses the auxiliary function method of Section 4.4 on the objective in the problem (4.9) rather than (4.28) as $x_{k,b}$ can now have fractional values. This two stage procedure is repeated until the benefit in system utility falls below a threshold. Finally, the

Joint RA-AF algorithm rounds $x_{k,b}$ to obtain a feasible association.

4.6 Evaluation

We present a detailed evaluation of our proposed Greedy Local Search (GLS), Restricted Greedy (RG) and Joint Association & AF algorithms for an LTE HetNet deployment. In our evaluation topology (Fig. 4.2), an enhanced NodeB (eNB) covers the coordination area. The eNB site consists of three cells (sectors), where each sector contains a set of 11 transmission points formed by one macro and ten lower power (pico) nodes (red squares in Fig. 4.2). We drop 99 users (green dots in Fig. 4.2) on the eNB site. Thus, there are a total of $B = 33$ TPs and $K = 99$ users. All TPs and users have a single antenna each. The transmit power for macro BS and pico nodes is 46 dBm and 35 dBm respectively. We assume a noise power of -104 dBm.

We use the simplified expression in (4.33) for optimization and evaluation in Section 4.6.1 and Section 4.6.2. All the other parameters used in our numerical studies, such as the distributions for dropping users and pico nodes, the path loss and shadowing values, are according to 3GPP evaluation guidelines unless otherwise stated. We also assume identical weights for all users and use $\Delta = 10^{-9}$ to capture even small improvements in the LS stage.

We compare the GLS and RG algorithms proposed in Section 4.3 to the following association schemes:

- Relaxed Upper bound (RU)–Solves the convex optimization problem obtained

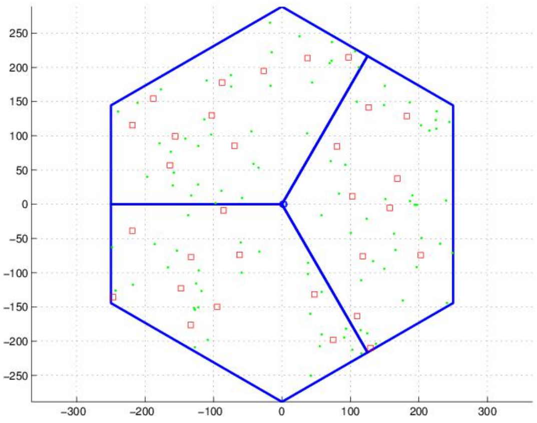


Figure 4.2: Topology

by relaxing $x_{k,b}$ in (4.9) or (4.10). Though the obtained solution need not be feasible for (4.6), the scheme provides us with an upper bound to the association problem (4.6).

- Relaxed Rounded Association (RRA)–Solves the convex optimization problem obtained by relaxing $x_{k,b}$ in (4.9) or (4.10). Each user k connects to the TP b corresponding to highest $x_{k,b}$ in the obtained convex optimization solution. This scheme requires solving a convex problem and it can be computationally quite complex compared to the GLS/RG algorithms in a dense deployment.
- Max SNR Association (MSA)– Each user independently connects to the TP from which it sees the highest average channel gain.

4.6.1 Association

In this section we evaluate the association algorithms by examining their returned utility function values for varying α. We also evaluate the additional gain yielded by the local search (LS) stage over the greedy one in the GLS algorithm.

α	Greedy	GLS	RU	RRA	MSA	RG	LSI
0.25	67.75	67.82	67.82	67.82	65.08	67.48	1
0.5	112.67	112.67	112.71	112.52	107.03	110.39	0
0.75	288.57	288.57	288.82	288.46	277.65	283.98	0
1.0	-133.93	-133.87	-133.3 1	-133.93	-154.67	-139.76	1

Table 4.3: Utility versus α

$\alpha \leq 1$: We begin with an evaluation of the GLS and RG algorithms in the regime $\alpha \leq 1$, where we consider the maximization of the objective in (4.10). We set $\rho = 1$ for each of the 33 TPs and list the utility values of different association algorithms in Table 4.3. As suggested by the guarantee in Proposition 4.3, we observe that the greedy stage of the GLS Algorithm itself performs very close to the upper bound RU, and hence close to the optimal and provides good gains over the MSA scheme. In the $\alpha \leq 1$ regime, the main advantage of the proposed RG and GLS algorithms over RRA is they have a much lower computational complexity, while performing at par with the latter. An additional advantage of the RG algorithm is that we can use it to assign TPs to the arriving users in an online fashion (without altering previous associations). We also observe from Table 4.3 that the local search iterations (LSIs) of the GLS Algorithm are at most 1 and that there is a slight utility gain obtained by the LS stage. This gain is limited because here the greedy stage itself is almost optimal.

$\alpha > 1$: Next we study the performance of GLS and RG algorithms in $\alpha > 1$ region, where we consider the minimization of the objective in (4.9). As seen in Fig. 5.1, the proposed GLS and RG algorithms perform very similarly and they noticeably outperform RRA in $\alpha > 3$ regime while beating MSA over the entire range of $\alpha > 1$.

α	Greedy	GLS	LSI
1.25	563.9	563.9	0
1.5	411.4	411.3	1
1.75	408.7	406.8	2
2.0	462.6	458.9	2
2.25	565.6	559.0	2
2.5	728.5	717.2	2
2.75	975.2	956.1	2
3.0	1345.8	1314.2	2
3.25	1904.6	1853.0	2
3.5	2754.6	2671.2	2
3.75	4045.1	3911.4	2
4.0	5953.6	5740.7	2

Table 4.4: Local search improvement

For example, GLS performs 13.5 % better than RRA and 80% better than MSA at $\alpha = 4$. MSA performs poorly throughout the $\alpha > 1$ regime since that algorithm has a user specific rather than system specific view. The superiority of GLS and RG over RRA & MSA increases with increase in α. For example, at a high $\alpha = 10$, which approaches max-min fairness, the GLS outperforms RRA and MSA by 93.2 % and 100% respectively.

In Table 4.4 we study the advantage of doing local search in the $\alpha > 1$ region. It is known that the greedy algorithm does not yield a constant factor approximation for the minimization of a supermodular set function. Therefore, the greedy stage need not be close to the optimal and there is room for improvement by the LS stage. As seen in Table 4.4, though the number of LS iterations are at most two, the order of gain over the greedy is up to 3.6%. At a higher $\alpha = 10$, the gain of GLS over greedy jumps to 43%, with the number of LSI equal to 5. Therefore, as α is progressively increased, the local search stage of the GLS algorithm becomes increasingly important.

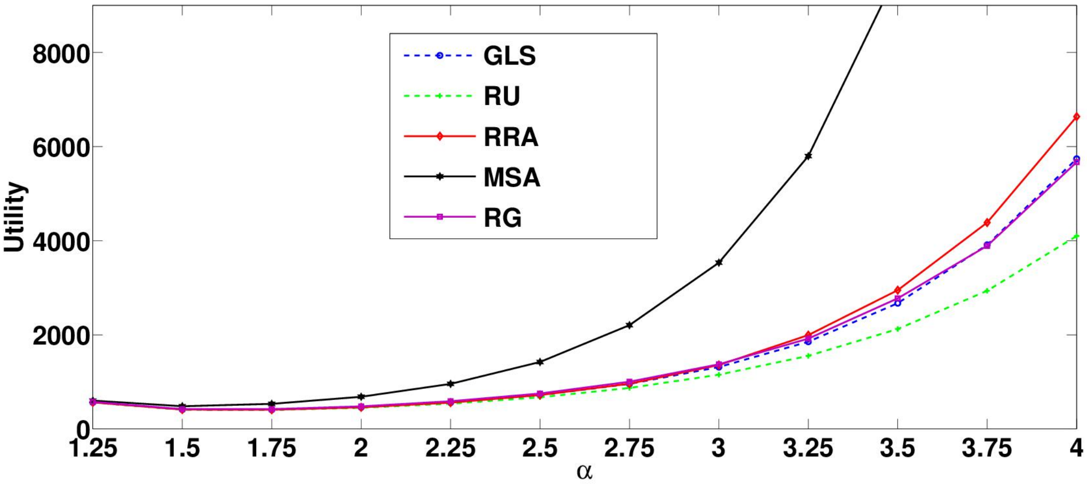
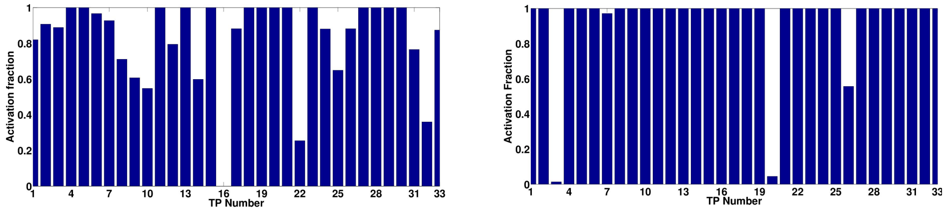


Figure 4.3: Utility vs α

4.6.2 Joint association and activation fraction optimization



α=3 α=0.5

Figure 4.4: Activation fractions for heterogeneous networks.

Fig. 4.4 illustrates the different AF yielded by the Joint GLS-AF algorithm for $\alpha = 0.5$ and $\alpha = 3.0$, respectively. We observe that in both the cases at least one macro AF is reduced to minimize the macro interference to small cells. For $\alpha = 0.5$, maximizing the utility forces the AF of a macro to almost zero, whereas the AF in all small cells except three is equal to one. In our evaluation, we observe that a high percentage of activation fractions in the $\alpha < 1$ range are binary $\{0, 1\}$ valued. For example, note that the AF when $\alpha = 0.5$ are mostly binary except for that of one pico TP. It can be proved that for $\alpha = 0$, i.e., the weighted sum rate utility, the optimal activation fractions will be binary valued. These observations do not apply for AF optimization at higher α values, as evident in Fig. 4.4.

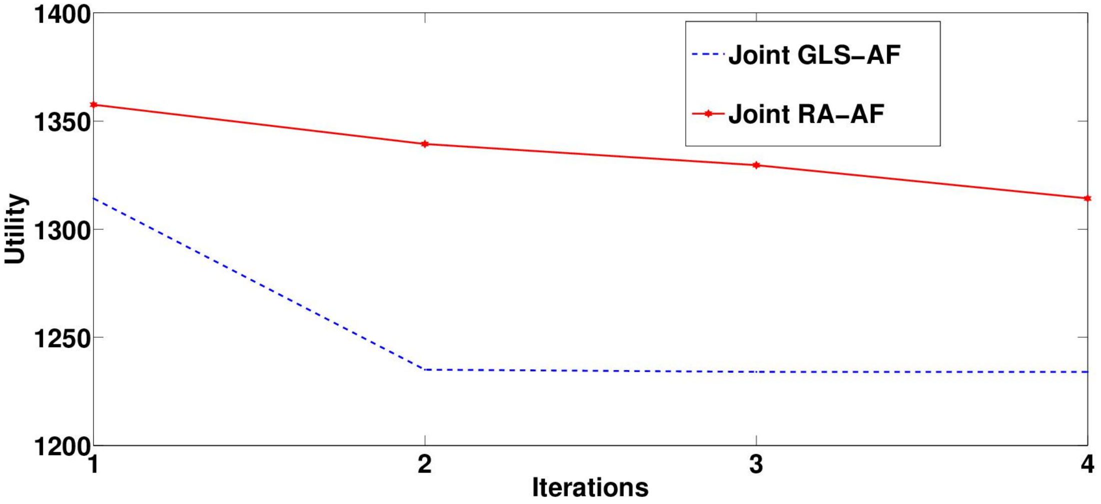


Figure 4.5: Utility vs iterations

In Fig. 4.5 we study the performance of the two joint algorithms for $\alpha = 3.0$ up to 4 iterations. Each point in the plot at an iteration is the utility value corresponding to the updated association, where that association is calculated using the updated value of activation fractions. The value at the first iteration is the utility corresponding to the association done using AF equal to one for all TPs. In the Joint RA-AF, at every iteration we calculate the utility by rounding the fractional association as done in the RRA algorithm. However, as mentioned in Section 4.5, fractional values of association variables $\{x_{k,b}\}$ are passed on to the second stage of AF identification. MSA with $\rho = 1$ for each TP with a utility value of 3531.8, performs much worse than the Joint GLS-AF & Joint RA-AF schemes. We obtain a gain of 6.1% for Joint GLS-AF over the case when we do only association via GLS with a fixed $\boldsymbol{\rho} = \mathbf{1}$, which demonstrates the benefit of doing the joint association and AF optimization. The Joint RA-AF scheme performs worse (upto 8.45%) than the Joint GLS-AF algorithm at every iteration, illustrating that the benefits of GLS over RRA observed before at $\boldsymbol{\rho} = \mathbf{1}$ are preserved even in the joint optimization problem.

For $\alpha = 0.5$, Joint GLS-AF performs 23.36% better than MSA with $\boldsymbol{\rho} = \mathbf{1}$, as compared to the gain of 4.6% obtained by GLS over MSA observed in Table 4.3, again demosntrating the gain of optimizing AF and the association jointly. We observe that Joint GLS-AF and Joint RRA-AF algorithms perform very close to each other in $\alpha < 1$ regime. This is because of the at par performance of GLS and RRA schemes in this α regime.

4.7 Association and activation fractions (AFs) for users with limited queue

In a practical LTE system, a user might need limited service restricted to a fraction of frame. Moreover, due to the burstiness of the traffic, users should have an upperbound on their service in each frame. Otherwise, the extra resources allocated to a user by a TP in a frame might be wasted. Therefore, we study the association and activation fraction optimization for limited queue size of the users in addition to the backlogged traffic model studied in previous sections of this chapter. We can now write our joint optimization problem of interest, which is a mixed optimization problem, as

$$\boxed{\begin{aligned} &\max_{\substack{\boldsymbol{\rho}\in[0,1]^B; x_{k,b}\in\{0,1\};\\ \gamma_{k,b}\in[0,1]\ \forall\ k,b}} \left\{\sum_{k\in\mathcal{U}}\sum_{b\in\mathcal{B}} x_{k,b}\left(w_k u(\gamma_{k,b}R_{k,b}(\boldsymbol{\rho}))\right)\right\} \\ &\text{s.t.} \sum_{b\in\mathcal{B}} x_{k,b} = 1,\ \forall\ k\in\mathcal{U};\ \sum_{k\in\mathcal{U}}\gamma_{k,b}\leq 1\ \forall\ b\in\mathcal{B};\ \gamma_{k,b}R_{k,b}\leq Q_k. \end{aligned}} \tag{4.34}$$

The third constraint puts an upper bound on the allocation fraction such that the service provided by the TP does not exceed the service required by user. This problem is intractable and it can be proved that even the association sub-problem is NP-hard, because the association problem for even the backlogged traffic model (unconstrained traffic) is NP hard.

We use an alternating optimization framework to solve the problem jointly. We adopt an iterative two stage procedure, where the first stage determines the association assuming activation fractions are fixed. The second stage determines the activation fractions assuming association is the one which is obtained in the first stage. We iterate over the two stages until we stop geting benefit in terms of system utility. The two stages are solved as follows:

Stage 1 Association- We fix the activation fractions in (4.34) and obtain a mixed optimization problem. Further, we optimize over the allocation fractions $\gamma_{k,b}$ using KKT conditions. Note that the solution to optimization over allocation fraction partitions the users connected to a TP b into two subsets $U'{}_b$ and U''_b. The quantity

λ_b partitions the two sets such that

$$\begin{aligned} &k \in U'_b \quad \lambda_b > \frac{R_{k,b} w_k}{{Q_k}^{\alpha}} \\ &k \in U''_b \quad otherwise, \end{aligned} \tag{4.35}$$

where λ_b is given by

$$\lambda_b = \{ \frac{\sum_{k \in U'_b} R_{k,b}^{(1-\alpha)/\alpha} {w_k}^{1/\alpha}}{1 - \sum_{k \in U''_b} Q_k / R_{k,b}} \}^{\alpha}. \tag{4.36}$$

Using the optimal allocation fractions, we can reduce the mixed optimization problem to a discrete problem. We conjecture that the objective of the discrete problem has a known structure such as submodularity or supermodularity. Further, we use a form of greedy and local search algorithms as proposed in the discrete optimization literature to determine the association.

Stage 2 AF optimization- We fix the association obtained in the first stage and obtain a continuous optimization problem. We will use auxiliary function method for the problem such that the procedure converges to a solution that satisfies KKT conditions.

Chapter 5: Coordinated Scheduling in MIMO Heterogeneous Wireless Networks using Submodular Optimization

5.1 Introduction

In the hetnet architecture, the basic unit of coordination is referred to as a cluster. Clusters partition the set of TPs in the system and each cluster is assigned a separate set of users that it should serve. The TPs in the cluster are connected through high-speed backhaul and can coordinate in fine time scale granularity of the order of a subframe. However, the coordination amongst different clusters is at a slower timescale of the order of hundreds of milliseconds. In this chapter, we focus on the resource allocation within each cluster in a fine time scale of subframes (milliseconds). The set of users and the set of TPs in the cluster are fixed at this fine time scale. In this work, we assume that each user can be served by a set of TPs in the coordinated multipoint (COMP) cluster. *We seek to determine which users should be scheduled in each subframe, and which TPs should be assigned to the scheduled users to maximize the system utility in the spatial multiplexing MIMO COMP schemes.* This COMP scheduling problem is quite challenging due to the fact that as we increase the number of TPs serving a user, we increase the throughput of that user, but limit service to other users.

5.2 System model

We consider the downlink in a heterogeneous network with universal frequency reuse and focus on a cluster of TPs, which can simultaneously transmit in a subframe. Let $\mathcal{B}$ denote the set of TPs in the cluster, and $\mathcal{U}$ denote the set of users in the coordination unit, which need to be served by $\mathcal{B}$ in a physical resource block (PRB). We assume that there is a single PRB in each subframe. Each user may connect to a set of TPs $\mathcal{S} \subset \mathcal{B}$. However, each TP can serve only a single user in a PRB. The problem is to design a centralized scheduling scheme which at each subframe selects a set of users to be served in the subframe, such that each user is served by a disjoint set of TPs. The decision variables are $\mathbf{x} = \{x_{i,b}\} \forall i, b$, where $x_{i,b} = 1$ if user i is associated with TP b, and $x_{i,b} = 0$ otherwise.

The objective is to maximize the sum of user utility $\sum_{i \in \mathcal{U}} u_i(R_i, \alpha)$, where R_i is an estimate of average throughput, $\alpha \geq 0$ is a tunable fairness parameter and u_i is the utility function of user i given by:

$$u_i(R_i, \alpha) = \begin{cases} \frac{R_i^{(1-\alpha)}}{1-\alpha} & \alpha \in (0, 1) \\ \log(R_i) & \alpha = 1 \\ -\frac{R_i^{(1-\alpha)}}{\alpha-1} & \alpha > 1. \end{cases} \tag{5.1}$$

This family of utility functions defines various tradeoffs between total throughput and fairness. We favor system throughput by choosing small values of α, and favor system fairness by choosing larger values of α. The case of $\alpha = 1$ provides proportional fairness (PF).

The channel matrix from a TP to a user contains elements $h_{m,n}$, which is the channel gain from the n^{th} transit antenna of the TP to the m^{th} receive antenna of the user. Let $H_{i,b}$ be the channel matrix from TP b to user i. Based on the channel from serving TPs, the user feeds back precoder matrix index (PMI) to the central scheduler, which determines the precoder weights. The precoder weights of each user are assumed to be known at the scheduler. $W_{i,b}$ denotes the precoding matrix for TP b transmission to user i. The signal received by user i is given by

$$Y_i = \sum_{b \in \mathcal{B}} H_{i,b} W_{i,b} s_{i,b} x_{i,b} + \sum_{b \in \mathcal{B}} \sum_{k \neq i} H_{i,b} W_{k,b} s_{k,b} x_{k,b}. \tag{5.2}$$

$s_{i,b}$ is the stream of data transmitted from TP b to user i, and is independent of all streams $s_{k,j}$, where $k \neq i$ or $j \neq b$.

The maximum achievable rate [62] by user i is given by

$$\log_2(det(I_{n_r} + K_i^d(\mathbf{x}) \cdot (K_i^n)^{-1}(\mathbf{x})), \tag{5.3}$$

where $K_i^d(\mathbf{x})$ is the desired signal matrix for user i given by $\sum_{b \in \mathcal{B}} (x_{i,b} H_{i,b} W_{i,b} W_{i,b}^H H_{i,b}^H P_b)$, $K_i^n(\mathbf{x})$ is the noise and interference matrix given by $\sum_{b \in \mathcal{B}} \sum_{k \neq i} (\sigma^2 I_{n_r} + x_{k,b} H_{i,b} W_{k,b} W_{k,b}^H H_{i,b}^H P_b)$. Here n_r denotes the number of receive antennas at each user, and I_{n_r} is the identity matrix of dimension equal to the number of receive antennas at each user. P_b is the transmission power of TP b.

The gradient-based scheduling algorithm was proved to be asymptotically optimal for concave and continuously differentiable utility functions of long term user average rates in [57], assuming users always have data to be served. This scheme can be easily extended to our COMP setting. $r_i(\mathbf{x})$ is the effective rate that the user i gets when it is served by the TP set determined by $\mathbf{x}$. $R_i(t)$ is the estimated long term average throughput of user i at time instance of subframe t. $R_i(t)$ is updated for the next subframe according to

$$R_i(t+1) = (1-p)R_i(t) + pr_i(\mathbf{x}), \tag{5.4}$$

where p is the average throughput update parameter.

5.3 Problem statement

In order to maximize the average system utility, in each subframe we wish to maximize weighted sum of rates obtained by each user in subframe, given by $\sum_{i \in \mathcal{U}} w_i r_i(\mathbf{x})$ [57]. Here, $r_i(\mathbf{x})$ is the maximum achievable rate by user i in the PRB and is given by (5.3) and w_i is inversely proportional to the long term throughput achieved by user i.

$$w_i(R_i, \alpha) = \tfrac{1}{R_i^\alpha} \qquad \alpha \geq 0 \tag{5.5}$$

The discrete optimization problem can be stated as

COMP Scheduling Problem

$$\boxed{\begin{array}{l} \text{maximize}_{x_{i,b}\in\{0,1\}} \\ \displaystyle\sum_{i\in\mathcal{U}} w_i \log\left(det(I_{n_r} + K_i^d(\mathbf{x}) \cdot (K_i^n(\mathbf{x}))^{-1})\right) \\ \text{subject to} \\ \displaystyle\sum_{i\in\mathcal{U}} x_{i,b} \le 1 \quad \forall b \end{array}} \tag{5.6}$$

The constraint in the optimization problem ensures that no two users in a PRB are served by the same TP. The quantity $\sum_{i\in\mathcal{U}} x_{i,b}$ determines if a TP b transmits in the subframe. If $\sum_{i\in\mathcal{U}} x_{i,b} = 0$, then the TP b is silent in the subframe.

Proposition 5.1 The problem (5.6) is NP-Hard.

Proof. We prove the hardness of MIMO COMP scheduling for fixed precoders by showing that even the MISO version of the problem is NP-hard. We assume that TPs serve using multiple antennae, and users have only a single antenna. We will show that MISO COMP scheduling for fixed precoders is NP-hard by showing that the subset sum problem, which is NP-hard, is a special case of MISO COMP scheduling problem. Consider the scheduling problem for input instances in which there are N TPs in set $\mathcal{B}$ and only two users in the cluster. Also, the quantity

$$\begin{aligned} ||H_{i,b}W_{k,b}||^2 P_b &= Kc_b, \quad i = k \\ &= 0, \quad otherwise \end{aligned}$$

where $c_b \in Z^+, \quad \forall b \in B$. The COMP scheduling problem is to come up with subsets of TPs that serve each user. Note that the set of TPs serving a user will be exclusive of the set of TPs which serve the other user, as each TP can serve not more than a single user. For equal weight users, the problem can be stated as

$$\begin{aligned} maximize_{x_{i,b} \in \{0,1\}} & \log\left(1 + \frac{\sum_{b \in \mathcal{B}} c_b x_{1,b}}{\sigma^2/k}\right) \\ & + \log\left(1 + \frac{\sum_{b \in \mathcal{B}} c_b x_{2,b}}{\sigma^2/k}\right) \end{aligned} \tag{5.7}$$

where both the vectors $\mathbf{x1} = \{x_{1,b}\}, \ \forall b$ and $\mathbf{x2} = \{x_{2,b}\}, \ \forall b$ are mutually exclusive, i.e., **x1** and **x2** cannot have value equal to one at the same position. Also, the solution must satisfy the constraint that $\sum_{b \in \mathcal{B}} c_b x_{1,b} + \sum_{b \in \mathcal{B}} c_b x_{2,b} \leq \sum_{b \in \mathcal{B}} c_b$. Note that the objective in the problem (5.7) is concave, and the maximum possible value of the objective is $2 \log\left(1 + \frac{\sum_{b \in \mathcal{B}} c_b/2}{\sigma^2/k}\right)$. This is when $\sum_{b \in \mathcal{B}} c_b x_{1,b} = \sum_{b \in \mathcal{B}} c_b x_{2,b} = \sum_{b \in \mathcal{B}} c_b/2$.

Since **x1** and **x2** are mutually exclusive, the solution of the COMP scheduling problem guarantees the maximum value of objective $2 \log\left(1 + \frac{\sum_{b \in \mathcal{B}} c_b/2}{\sigma^2/k}\right)$ when the solution of problem (5.7) obtains two subsets of the set $\{c_b\} \ \forall b$ which have equal sum and form a partition of the set $\{c_b\} \ \forall b$. Indeed, suppose $\{y_1, .., y_K\} : y_k > 0, \forall k$ is any input set to the subset sum problem where we need to determine if there are two subsets which form a partition of the set and have equal sum. We can appropriately map the input set to the corresponding set $\{c_b\} \ \forall b$. Then from the solution of scheduling problem, we can determine if there are two subsets that are disjoint and have equal sum. It is clear that given a subset sum problem, we can construct a corresponding equivalent COMP scheduling problem. Hence, if there exists an optimal polynomial time COMP scheduling algorithm, it can also solve the subset sum problem on any set and, hence, is optimal polynomial time for the NP hard subset sum problem. □

The objective of the problem (5.6) can be expressed as the following difference of two weighted logarithmic sums:

$$\begin{aligned}\sum_{i \in \mathcal{U}} w_i \log(det(K_i^n(\mathbf{x}) + K_i^d(\mathbf{x}))) \\ - \sum_{i \in \mathcal{U}} w_i \log(det(K_i^n(\mathbf{x}))).\end{aligned} \tag{5.8}$$

If we allow $x_{i,b}$ to take continuous values in the interval range [0,1] rather than just binary values, each weighted logarithmic sum in (5.8) can be proved to be concave in the following way: We note that $\log(det(X))$, $\forall X \in S_n^{++}$ is concave in X, where S_n^{++} is the set of $n \times n$ positive definite matrices. Also, we note the fact that composition of a nonnegative affine function with a real-valued concave function yields a concave function. Further, the weighted sum of concave functions is concave if the weights are nonnegative.

The version of problem (5.6) in which we allow the $x_{i,b}$ to take continuous values in the interval range [0,1] is referred to as, the relaxed version of problem (5.6) throughout this chapter. Note that even the relaxed version of problem (5.6) is hard to solve as the objective is a sum of concave and convex functions which makes it a non-concave maximization problem [39].

5.4 Proposed approach

We begin with an outline of our proposed approach using successive convex approximation. Next, using the discrete version of successive convex approximation, we describe different steps in order to solve the original (discrete) problem. As noted in [37], in an alternative approach [39, 53], continuous version of successive convex approximation scheme can be used to solve the relaxed problem. In the next section, we present the continuous version of the scheme for the relaxed (continuous) version of the problem (5.6). Fig. 5.1 gives a unified block-level decomposition of discrete and continuous versions of successive convex approximation. In the first step, we represent the objective as a difference of two functions $f(x) - g(x)$, where both the functions $f(x)$ and $g(x)$ are submodular in the case of (5.6). Next we follow an iterative procedure, which iterates upon two steps, namely step 2 and step 3. In step 2 of the k^{th} iteration in this procedure, we approximate $g(x)$ as a modular function $g_k(x)$ (or an affine function in the case of the relaxed problem), where $g_k(x)$ is an upper bound to $g(x)$. In step 3 of k^{th} iteration in this procedure, we maximize the objective $f(x) - g_k(x)$, which is effectively maximizing the lower bound of the original objective. We iterate upon steps 2 and 3 until we converge to a solution. In the following subsection, we describe the difference of submodular function algorithms, which is a discrete version of the successive convex approximation procedure in Fig. 5.1.

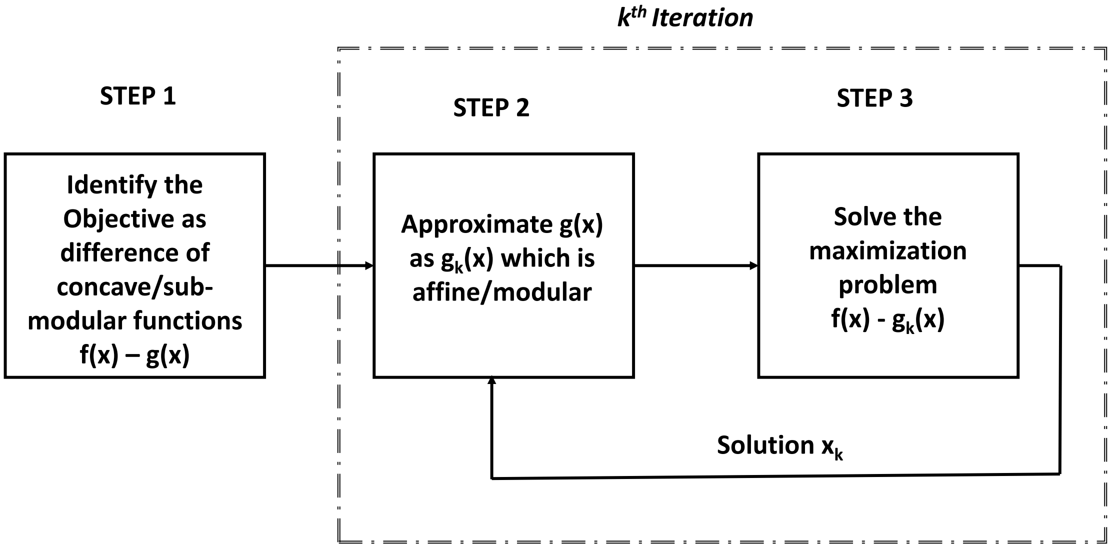

kth Iteration
STEP 1
STEP 2
STEP 3
Identify the Objective as difference of concave/sub-modular functions f(x) – g(x)
Approximate g(x) as gk(x) which is affine/modular
Solve the maximization problem f(x) - gk(x)
Solution xk

Figure 5.1: Block diagram

5.4.1 Difference of submodular (DS) function algorithm

We now proceed to reduce the COMP scheduling problem (5.6) to a constrained maximization of difference of two submodular functions. Interestingly, we can represent any set function as a difference of two submodular functions. However, in general this requires exponential complexity [35]. In order to express the objective set function in (5.6) as a DS function as done in *Step 1* of Fig. 5.1, we first define a ground set $\Omega = \{(i, b) : i \in \mathcal{U}, b \in \mathcal{B}\}$. The ground set consists of all possible tuples, and each tuple (i, b) denotes an association of user i to TP b. Further, we define a family of sets $\mathcal{I}$ as the one which includes each subset of Ω such that the tuples in that subset have mutually distinct TP. (Ω, I) is said to be a *partition matroid* when there exists a partition $\Omega = \cup_{i=1}^{J} \Omega_i$, where $\Omega_i \cap \Omega_j = \emptyset$, $\forall\ i \neq j$, along with integers

$n_i \geq 1 \ \forall \ i$ such that

$$C \subseteq \Omega : |C \cap \Omega_i| \leq n_i \ \forall \ i \Leftrightarrow C \in I. \tag{5.9}$$

Each subset of tuples having a common TP-i constitute the set Ω_i, and each such set Ω_i containing distinct TP form the partition of the ground set Ω. Using the definition we see $\mathcal{I}$ is a partition matroid. Let G be the solution set, which should belong to the partition matroid $\mathcal{I}$. Let

$$\begin{aligned}\phi(G) = &\sum_i w_i \log \Bigg(det(\sum_{b:(i,b)\in G} H_{i,b} W_{i,b} W_{i,b}^H H_{i,b}^H P_b \\ &+ \sum_{(k,b)\in G: k\neq i} H_{i,b} W_{k,b} W_{k,b}^H H_{i,b}^H P_b + \sigma^2 I_{n_r}) \Bigg)\end{aligned} \tag{5.10}$$

and

$$\begin{aligned}\phi'(G) = &\sum_i w_i \log \Bigg(det(\sum_{(k,b)\in G: k\neq i} \\ &H_{i,b} W_{k,b} W_{k,b}^H H_{i,b}^H P_b + \sigma^2 I_{n_r}) \Bigg)\end{aligned} \tag{5.11}$$

Proposition 5.2: For any α, the set functions $\phi(G)$ and $\phi'(G)$ are nondecreasing submodular set functions over Ω.

Proof. We note that $\log(det(X))$, $\forall X \in S_n^{++}$ is concave in X. Then, we note the fact that composition of a nonnegative modular set function with a real-valued concave function yields a submodular set function. Further, the weighted sum of submodular functions is submodular if the weights are nonnegative. Also submodularity is preserved under set restriction. Using these facts, we obtain the desired result. □

The COMP scheduling problem (5.6) can be reduced to the following maximization of DS function over a partition matroid.

$$\text{maximize}_{G\in\mathcal{I}} \quad \phi(G) - \phi'(G) \tag{5.12}$$

The problem of maximizing the difference of any two submodular functions generalizes the problem of maximizing a submodular function, and hence it is NP hard as well. Unfortunately, even obtaining an approximation algorithm for the difference of submodular maximization is known to be NP hard [35]. These hardness results provide a justification for use of an inspired heuristic algorithm for the difference of submodular maximization problem.

For solving problem (5.12), we use an approximate difference of submodular function maximization approach proposed for machine learning applications in [44], [35]. This approach is based on concave-convex procedure [64]. Concave functions are a continuous analogue of submodular functions, and therefore this is a natural approach for DS functions. The algorithm is a discrete analogue of successive convex approximation scheme that depends on discrete superdifferentials. Iyer et. al. in [37] show that many of the state-of-the-art submodular function optimization approaches are special cases of this procedure. Furthermore, this procedure is shown [37] empirically to work very well for submodular function optimization. Thus, we use the procedure for the problem (5.12).

As shown in *Step 2* of Fig. 5.1, we need to approximate the submodular function ϕ' by a modular function ϕ'_H, which is exact at a set $H \in \Omega$. The modular approximation should satisfy the following properties:

- $\phi'(H) = \phi'_H(H)$
- $\phi'(G) \leq \phi'_H(G) \;\; \forall G \subset \Omega$

Bilmes et. al. in [36, 38] showed that we can define supergradient $\partial\phi'(H)$ of a sub modular function ϕ' at H:

$$\begin{aligned}\partial\phi'(H) = &\{y \in R^n : \\ &\qquad \phi'(G) - y(G) \leq \phi'(H) - y(H); \forall G \subset \Omega\}\end{aligned}$$

A supergradient at H is defined as $\partial\phi'(H) = \phi'(j|H \setminus j)$ for $j \in H$ and $\partial\phi'(H) = \phi'(j|\emptyset)$ for $j \notin H$ in [37]. Using this supergradient we use a modular approximation given by

$$\phi'(H) + \partial\phi'(G) - \partial\phi'(H) \geq \phi'(G)$$

The modular approximation which satisfies the above properties, reduces to the

Table 5.1: **DS Algorithm**

```
1: Initialize with $G^0$, $t = 0$.
2: Repeat
3: $G^{t+1} = argmax_G \quad f(G)$, where $H = G^t$.
4: Update $t = t + 1$
5: Until $G^{t+1} = G^t$.
```

following:

$$\phi'_H(G) = \phi'(H) - \sum_{j \in H \setminus G} \phi'(j|H \setminus j) + \sum_{j \in G \setminus H} \phi'(j|\emptyset), \tag{5.13}$$

where $\phi(H|G) = \phi(H \cup G) - \phi(G)$. Note that the function $\phi'(G)$ is submodular iff $\phi'_H(G) \geq \phi'(G)$ [35].

As shown in Fig. 5.1 at *Step 3*, we solve the maximization problem (5.12) by an iterative procedure, which involves solving the following submodular maximization problem at each iteration:

$$\begin{aligned} &\text{maximize}_{G \in \mathcal{I}} \quad f(G) \\ &f(G) = \phi(G) - \phi'(H) + \sum_{j \in H \setminus G} \phi'(j|H \setminus j) - \sum_{j \in G \setminus H} \phi'(j|\emptyset) \end{aligned} \tag{5.14}$$

We solve for G using the iterative DS algorithm given in Table 5.1. Each iteration (line 3), requires the constrained maximization of submodular function $f(G)$, which is done using the simple greedy algorithm given in Table 5.2. Note that we perform approximate constrained submodular optimization in every iteration, and we are not guaranteed to monotonically increase the objective in every iteration.

Table 5.2: **Greedy maximization**

```
Initialize with $\hat{G} = \emptyset$
Repeat
Determine $(k', b')$ as the tuple in $\Omega$ which offers the best gain among all tuples $(k, b) \in \Omega$
    such that $\hat{G} \cup (k, b) \in I$.
Update $\hat{G} = \hat{G} \cup (k', b')$
Until gain is positive.
Output $\hat{G}$.
```

However, if we ensure that we move to the next iteration only when the objective does not decrease, we will restore monotonicity at every iteration. This follows from the following set of inequalities:

$$\phi(G^{t+1}) - \phi'(G^{t+1}) \geq^{a} \phi(G^{t+1}) - \phi'_{G^t}(G^{t+1}) \geq^{b} \phi(G^t) - \phi'_{G^t}(G^t) =^{c} \phi(G^t) - \phi'(G^t).$$

Here, (a) follows from the upper bound property of the modular approximation, (b) follows from the assumption that we move to the next step only when the objective does not decrease, and (c) follows from the tightness of modular approximation at G^t.

Let $K = |\mathcal{U}|$ and $B = |\mathcal{B}|$. Since the total number of the (user, TP) tuples is KB, and the maximum number of iterations of the greedy algorithm is B, complexity of the greedy maximization in the DS algorithm is $O(B^2K)$. We note that simulations reveal that for a medium-sized cluster ($KB \approx 100$) only very few DS iterations (6 or less) are needed to capture the available gains.

5.5 Relaxed problem

We now proceed to relax the COMP scheduling problem (5.6) and solve the relaxed problem using DC method by expressing the objective as a difference of two concave functions.

The MISO version of this relaxed problem has structure similar to the problem of power control, which aims to maximize the weighted sum of rates in a single transmit and multiple interfering links setup [39, 43]. The non-concave problem is very challenging as pointed out in [9, 27]. Among efforts to solve for the global optimal solution of the power control problem, [9, 27, 61] use branch and bound techniques or outer approximation techniques. However, these techniques are very computationally demanding. We aim to solve the problem efficiently and using a low complexity algorithm. Therefore, we solve the problem using a difference of concave function optimization approach as done in [39] for the power control problem. The successive convex approximation procedure as proved in [43] converges to a local optimum. Moreover, numerical results in [39] for small dimensions of $\mathbf{x}$ indicate that the algorithm converges to the global optimum in small number of iterations. As done in *Step 1* of Fig. 5.1, using (5.8) the objective of the relaxed problem can be rewritten as difference of concave and differentiable functions. The successive convex approximation procedure generates a sequence $\{\mathbf{x}_k\}$ of improved feasible solutions for the relaxed problem. Now, as done in *Step 2* of Fig. 5.1 in the k^{th} iteration, we can approximate $g(\mathbf{x})$ as an affine function $g_k(\mathbf{x})$, such that $f(\mathbf{x}) - g_k(\mathbf{x})$ is a concave function. The solution of the successive convex approximation procedure converges to a point which satisfies the first order optimality conditions for the relaxed problem. Further, it can be proved that the solution in the $k+1$ iteration $\mathbf{x}_{k+1}$ improves the solution of previous iteration $\mathbf{x}_k$.

The size of the unknown decision variable $\mathbf{x}$ in the relaxed problem is KB.

The complexity of a convex program is of cubic order in number of decision variables. Therefore, the complexity of the concave maximization is $O(K^3B^3)$. The simulations for a medium-sized cluster ($KB \approx 100$) show that we require around 40 DC iterations for $\epsilon = 0.004$. Therefore, DC algorithm is much more computationally demanding than the DS algorithm. Though the solution need not be feasible, it provides us with an efficient benchmark for an upper bound of the COMP scheduling problem.

5.6 Evaluation

Parameter	Value
Macro transmission power per PRB	1000 mW
Pico transmission power per PRB	0.1 mW
Path Loss Exponent	3.5
log-normal shadowing standard deviation	6 dB
Coverage radius of the macro TP	700 m
Coverage radius of the pico BS	50 m
Power spectral density of noise	-172 dBm/Hz
log-normal shadowing standard deviation	6 dB
BW of a frequency resource block	180 kHz
Number of macro TPs in a cluster	1
Number of antennae at each user	4
Number of antennae at each TP	4
Min. distance between macro and pico TP	50 m
Min. distance between any 2 pico TPs	30 m

Table 5.3: Default parameters

We present an evaluation of the proposed DS algorithm. Table 5.3 contains the default parameters used in our numerical studies and simulations unless stated otherwise. We evaluate the COMP scheduling algorithms over a cluster topology consisting of a macro TP and multiple pico TPs. Pico TPs are distributed uniformly in the coverage area of the macro TP. Furthermore, pico TP user locations are uniformly distributed within the coverage area of picos. In the 2-tier network topology, the remaining users' locations are uniformly distributed outside the pico coverage area and within the coverage area of the macro TP. At the beginning of each realization or sample path, we generate mutually independent log-normal random variables to model the shadow fading between all (user, BS) pairs. We compute the signal strength for each user according to the path loss to each BS based on the parameters given in Table 5.3 and the realized random variables for shadow fading. Channel matrix between user i and TP b, $H_{i,b}$ is generated assuming Rayleigh fast fading, and can be represented as $U_{i,b} \sum_{i,b} V_{i,b}^H$ using SVD transform, where $\sum_{i,b}$ is a diagonal matrix and $V_{i,b}$ is a unitary matrix. For simplicity, we assume precoder matrix between user i and TP b, $W_{i,b} = V_{i,b}$.

In this section, firstly using simulations we show that the DS algorithm performs at par with the much more computationally complex DC and the optimal solution (calculated using brute force). Secondly, we study the DS solution with variation in different topology parameters and compare its performance with respect to the following algorithms:

- Weighted SNR (wSNR): Each TP b selects a user to serve based on the criterion $arg_i \quad \{max \ w_i \cdot \log(det(H_{i,b}W_{k,b}W_{k,b}^H H_{i,b}^H P_b + \sigma^2 I_{n_r}))$, where the term inside logarithm considers channel between user i and TP b, and neglects the interference from other TPs. Though this criterion is simple, it takes the parameter α into account as it depends on the user weight w_i, unlike schemes that are only based on the channel between user and TP.

- Long-term COMP (LCOMP): [22] In the first step, the recommended transmission set for each user is calculated. The set includes TP that provides the user highest signal strength and is called the anchor TP. In addition this set includes TPs whose long term signal strength is within a threshold (6 dB) from anchor TP. In the second step, the COMP coordination cluster is partitioned into COMP non-overlapping transmission sets. In the third step, for each transmission set, from the set of COMP users with matching recommended transmission set, search for optimal user maximizing the PF utility $w_i r_i$, where r_i is given by (5.3). Note that we assume $K_i^n = \sigma^2 I_{n_r}$ to limit the complexity of the algorithm. If we include the interference term in K_i^n, the computational complexity of the algorithm would be exponential as users served (and hence their precoders) by interfering TPs are not known in the third step. We repeat steps two and three over different partitions and choose the partition corresponding to the maximum weighted sum of user utility.

The initialization point for the DC and DS algorithms is based on wSNR criterion in the simulations. The convex optimization program in DC algorithm is implemented using CVX toolbox [34]. To contain the complexity of the DC algorithm, the maximum number of iterations is limited to 20 and $\epsilon = 0.1$.

We begin by evaluating the DS algorithm with respect to the optimal solution and DC solution, by examining their returned utility function values for proportional fairness utility ($\alpha = 1$). We consider a scenario which contains two pico TPs

Algorithm	Average System Utility
Optimal	11.36
DC	9.95
DS	9.89
wSNR	8.59
LCOMP	7.37

Table 5.4: PF performance of different algorithms.

distributed uniformly inside the macro coverage radius. Each pico TP contains six users placed uniformly in its coverage area. Ten users are distributed uniformly inside the macro coverage area and outside the pico coverage radius. We calculate the average system utility by taking an average of 80 realizations, where in each realization TP positions and user positions vary. As seen in Table 5.4, DS algorithm performs on average within 14.35% of the optimal and performs within 1.1% of DC algorithm. Though we limit complexity of DC algorithm in the simulations, because of its cubic complexity in KB, it still takes far more computation time than DS (and wSNR, LCOMP).

In the rest of this section, we focus on the DS algorithm. We study its performance as (i) the distance of pico TPs from macro, (ii) the number of users per pico, and (iii) the number of picos are varied. We compare the system utility obtained using DS algorithm for PF ($\alpha = 1$) to that of the baseline wSNR and existing LCOMP scheme.

Effects of distance between macro and pico TPs : We begin with a scenario in which there are two pico TPs inside the macro TP coverage radius. Each pico TP has six users placed uniformly inside its coverage radius. The coordinates of the

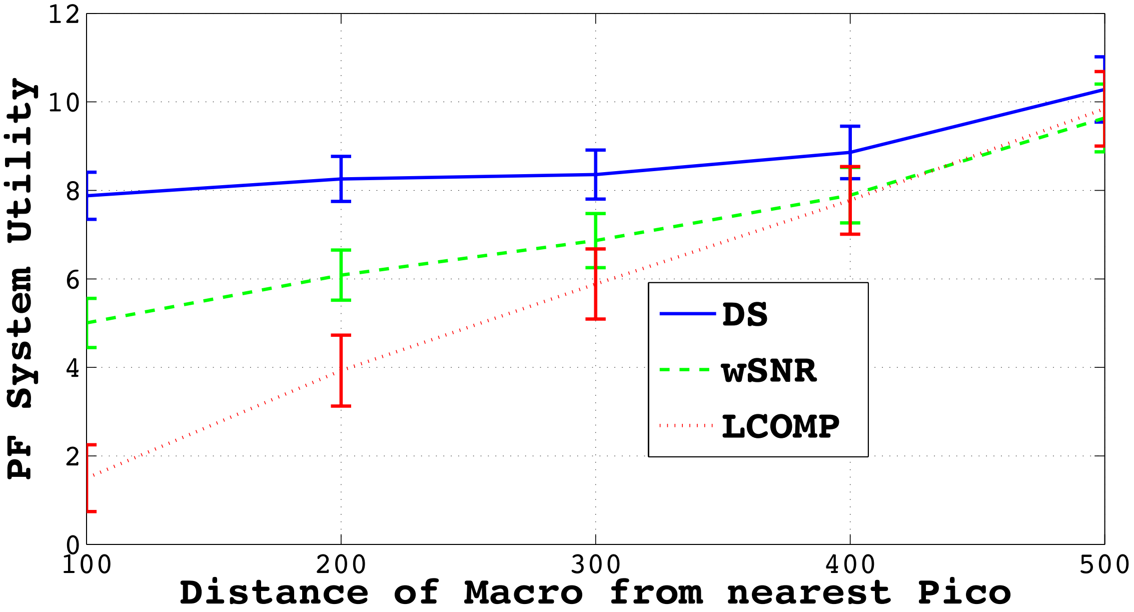


Figure 5.2: PF system utility with varying distance of closest pico TP from macro TP.

macro TP are (0,0) and those of the nearest pico and other pico are (100,0) and (200,0), respectively. We fix the inter-pico distance to 100 m in this experiment, and vary distance between the closest pico and macro TP, by moving both the pico TPs along x axis to generate different points in Fig. 5.2. Each point in Fig. 5.2 is an average of 80 realizations, where in each realization TP positions are fixed and user positions vary.

Unlike the wSNR algorithm, the DS algorithm can switch off TPs to reduce interference faced by users in a subframe. As a result, in Fig. 5.2, DS algorithm outperforms the wSNR algorithm by 60% and LCOMP algorithm substantially, when closest pico faces excessive interference (from macro) at a distance of 100 m from macro TP. Note that both the wSNR and LCOMP algorithms do not take into account the inter-TP interference, unlike the DS algorithm. As the distance of closest pico from macro TP is increased, the interference from the macro faced by picos decreases, and performance of wSNR and LCOMP algorithms approach that of DS scheme. In particular, the LCOMP algorithm performance suffers severely when pico TPs are close to macro TP, as the scheme limits the association of users based on average SNR.

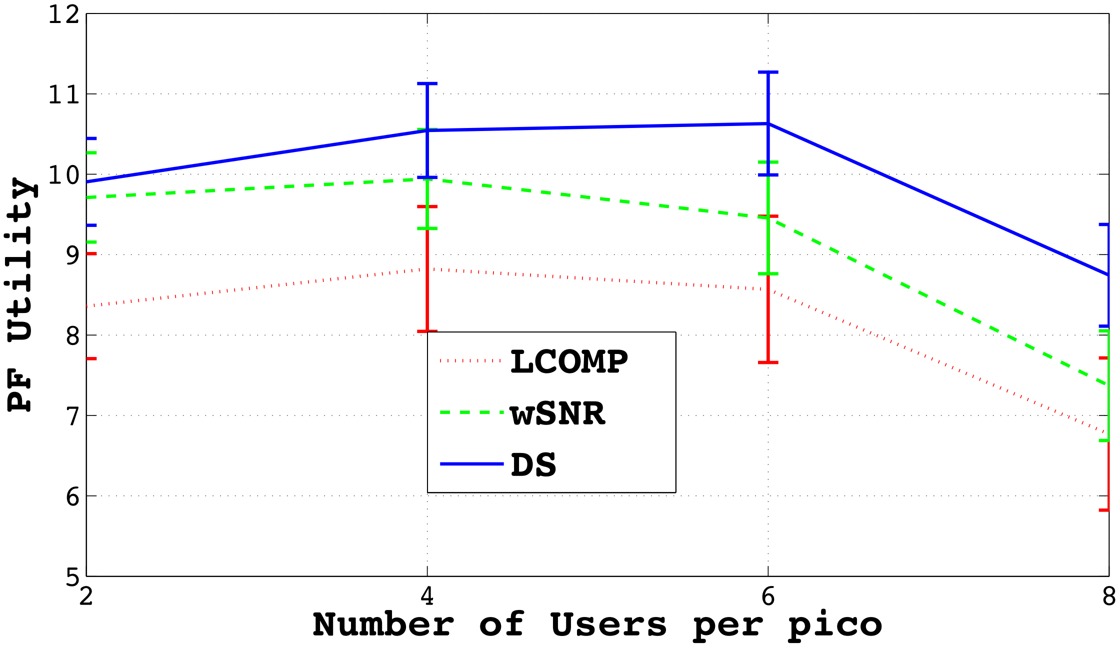


Figure 5.3: PF system utility with varying number of users per pico TP.

Effects of the number of users per pico : We plot the PF system utility as the number of users in each pico TP coverage region increases, where the users are placed uniformly in pico coverage area. In each realization, there are ten users placed outside the pico coverage area and inside macro coverage area. Also, two pico TPs are placed uniformly inside the macro TP coverage area in each realization. We compute an average of 80 such realizations in this experiment (Fig. 5.3).

Fig. 5.3 shows that as the number of users in pico coverage area increases, the performance gain of the DS algorithm over the wSNR algorithm increases from 2% up to 18.9%. This is due to the fact that DS algorithm, using pico TP on-off, controls the pico TP interference faced by users that are in pico coverage area but are connected to macro TP. Note that the performance of LCOMP algorithm with respect to wSNR algorithm improves with an increase in users in pico TP coverage area. The system fairness suffers using the LCOMP algorithm, due to limitation of user association based on long term signal strength. However, we observe that the LCOMP algorithm performs well in terms of the total user throughput.

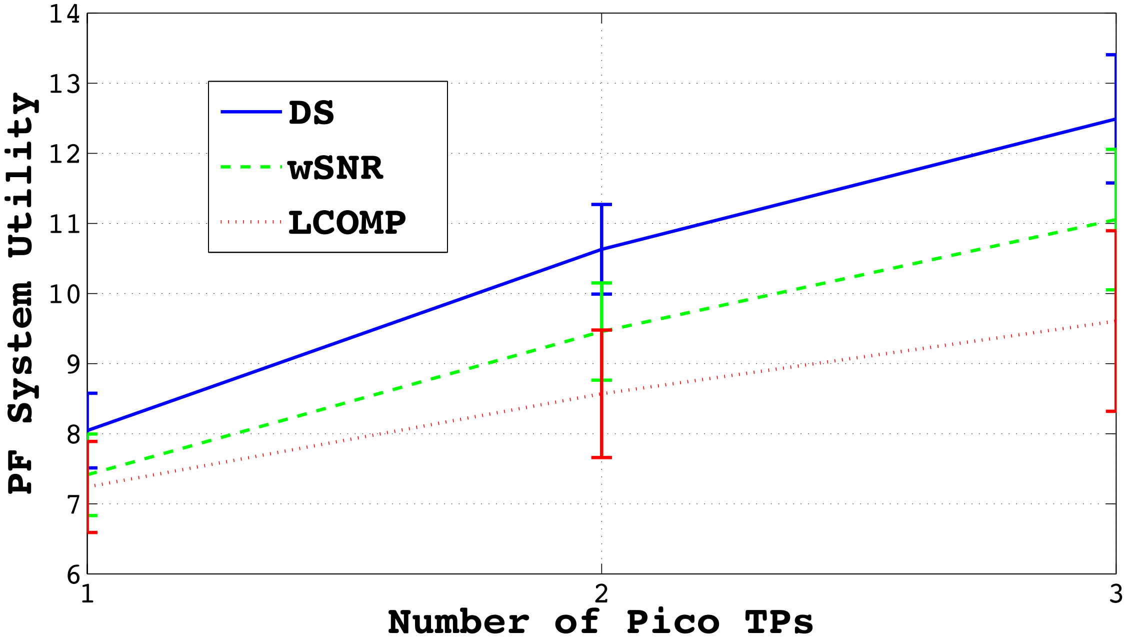


Figure 5.4: PF system utility with varying number of pico TPs in the cluster.

Effects of increasing network size : We consider a scenario in which there are ten users inside macro TP coverage radius but outside the pico TP coverage radius, and vary the number of pico TPs. Each pico TP has six (other) users inside its coverage radius, and the total number of users in the system increases with the number of pico

TPs. The picos are distributed uniformly in macro coverage area and we compute the average of 80 realizations.

With an increasing number of pico TPs, the PF system utility increases. This is because the number of users served by picos increases as the number of picos increases. However, the average inter-TP interference also increases. As shown in Fig. 5.4, the DS algorithm is able to handle increase in interference better than the other two algorithms. The DS algorithm performs 8% better than the wSNR algorithm and 11.1% better than the LCOMP scheme in a single pico TP scenario. As the number of picos is increased to 3, DS algorithm performs 13.2% better than wSNR algorithm and 31.5% better than LCOMP scheme.

Chapter 6: Conclusion

In Chapter 3, we have introduced and evaluated a new algorithm for frequency allocation in 2-tier heterogeneous networks. Our algorithm is completely distributed, requires little coordination between various BSs, and is able to effectively share frequency resources between an MBS and PBSs. Our algorithm is more efficient than previously known distributed schemes, and performs nearly on par with centralized LP-based optimal solutions if the PBS deployments allow for spatial frequency reuse.

There remain many open issues we have not investigated here. For instance, we have not explored the possibility of joint association and resource allocation and design of distributed algorithms. We plan to examine these open issues in the future and study how much benefit we can achieve through joint design of association and allocation schemes.

We also plan to study the comparison of our approach in which we minimize frequency resources while guaranteeing a minimum throughput to all users, and then use the remaining frequency resources for network utility maximization; with respect to the approach where network utility is maximized guaranteeing a minimum throughput to each user.

In Chapter 4, we introduced and evaluated a new joint association and activation fraction optimization algorithm for maximizing the α-fairness utility in HetNets. We derived meaningful performance guarantees and demonstrated the significant benefit of the joint algorithm over a practical HetNet topology.

Note that we have separately studied the frequency domain and time domain interference solution in Chapter 3 and Chapter 4, respectively. We plan to examine and compare the performance of solutions in both the domains. In future, we also plan to study a joint frequency and time domain solution to the interference problem.

Finally in Chapter 5, we have proposed a difference of submodular function optimization algorithm for the problem of COMP MIMO system utility maximization over the set of user and TP pairs. The discrete algorithm (DS) performs at par and is much less computationally complex than its continuous analogue (DC algorithm). Also, the DS algorithm performs within a reasonable percentage of the optimal solution. There remain many open issues we have not investigated here. For instance, we have not explored the possibility of joint precoder and TP-user pairs optimization. We plan study how much benefit we can achieve through joint design of precoder and TP-user pairs algorithms.

In this dissertation, we have designed algorithms for interference management assuming communication amongst TPs either at superframe level or subframe level. However, there could be practical deployment scenarios where neighboring TPs belong to different service providers and TPs operated by one service provider cannot communicate with TPs operated by other service provider. Therefore, in the future, we plan to extend our study of interference management algorithms to the schemes which do not require communication with TPs operated by other service providers.

Chapter : Appendix

Appendix A

Proposition 3.1. *Algorithmic constraints are equivalent to the set of original constraints.*

Proof. Suppose, BS-a_1 and BS-a_2 are neighbors in the interference graph. Let **S** denote a subset of BSs which interfere with BS-a_1 and BS-a_2. Both the FRB types (a_1: a_2,**S**) and (a_2: a_1,**S**) denote the sharing amongst same set of BSs, i.e; BS-a_1, BS-a_2 and BSs in the set **S**. Therefore, both the FRB types are compatible with each other. BS-Z is a neighbor of BS-a_1 but not a neighbor of BS-a_2. Hence, any FRB type of (a_1:a_2,**S**,Z) is compatible with type (a_2:a_1,**S**).

A set of constraints is introduced to couple the number of FRB types in the compatible sets for each pair of the neighboring BSs, in order to maximize the use of shared FRB types. For every pair of matching FRB types in any two neighboring BSs, the constraints equate total number of FRBs of different types present in the compatible set of each matching FRB. All the constraints generated using this criterion constitute an original constraint set.

Suppose **H1** denotes the set of BSs which interfere with BS-a_1 but not BS

a_2. **H2** denotes the set of BSs which interfere with BS-a_2 but not with BS-a_1. For every neighboring BSs a_1 and a_2 and $\mathbf{S} = \{a_3, a_4, ..., a_j\}$ in the interference graph, the original set of constraints corresponding to the FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and matching FRB type $(a_2 : a_1, a_3, a_4, ...a_j)$ is given by

$$\sum_{\mathbf{H'} \in P(\mathbf{H1})} N(a_1 : a_2, \mathbf{S}, \mathbf{H'}) = \sum_{\mathbf{H'} \in P(\mathbf{H2})} N(a_2 : a_1, \mathbf{S}, \mathbf{H'}) \tag{.1}$$

Note that P denotes the power set. $N(a_1 : a_2, \mathbf{S})$ represents the number of FRBs of type $(a_1 : a_2, \mathbf{S})$.

However, we note that the original set of constraints are equalities for a pair of matching FRB types. The set of equality constraints for all the matching FRB types can be given by the following set of equations. The equations are valid for any matching FRB of the type (a_1:a_2,$\mathbf{S}$). **H3** denotes the set of BSs that interfere with BS-a_1 but do not interfere with at least one BS which owns the matching FRB of type (a_1:a_2,$\mathbf{S}$). Similarly, **H4** denotes the set of BSs which interfere with BS-a_2 but do not interfere with at least one BS which owns the matching FRB of type (a_2:a_1,$\mathbf{S}$). For every neighboring BSs a_1 and a_2 in the interference graph, the original set of constraints corresponding to the FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and matching FRB type $(a_2 : a_1, a_3, a_4, ...a_j)$ is given by

$$\sum_{\mathbf{H'} \in P(\mathbf{H3})} N(a_1 : a_2, \mathbf{S}, \mathbf{H'}) = \sum_{\mathbf{H'} \in P(\mathbf{H4})} N(a_2 : a_1, \mathbf{S}, \mathbf{H'}) \tag{.2}$$

Let S_{SC} denote the set of FRBs which have their FRB type as SC. $N(SC) =$

$|S_{SC}|$. $\cup_{SC \in (a_1 : a_2, \mathbf{S}, P(\mathbf{H3}))} S_{SC}$ is the union of all the FRBs having the FRB type $(a_1 : a_2, \mathbf{S}, P(\mathbf{H3}))$ and are the physical FRBs of the types present in the extended compatible set of FRB type $(a_1 : a_2, \mathbf{S})$.

a) From algorithmic constraint to the corresponding original constraint.

Consider FRB types $(a_1 : a_2, a_3, a_4, ...a_j)$ and $(a_2 : a_1, a_3, a_4, ...a_j)$. The physical FRBs of the types present in extended compatible set of type $(a_1 : a_2, a_3, a_4, ...a_j)$ is denoted as ψ and physical FRBs of the types present in the extended compatible set of FRB type $(a_2 : a_1, a_3, a_4, ...a_j)$ is denoted as ϕ. We assume *(Assumption 1)* each element in the set of BSs $\{a_{j+1}, a_{j+2}..., a_{j+n}\}$ interferes with both BS-a_1 and BS-a_2. *(Assumption 2)* However, each element of this BS set does not interfere with at least one BS that owns the matching of FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$. The physical FRBs of the types present in the extended compatible set of type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+i})$ is denoted by ψ_i. Moreover, physical FRBs of the types present in the extended compatible set of type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+k_1}, a_{j+k_2}..a_{j+k_m})$ (where $1 <= k_1, k_2, .., k_m <= n$) is denoted by $\psi_{k_1,k_2,..,k_m}$.

$M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k}))$ is the set of FRB types which match with the FRB type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})$.

$\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})))$ is the set of owner BSs of the FRB types which match with the FRB type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})$.

We aim to derive the original constraint in equation (.1) corresponding to the FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and matching FRB type $(a_2 : a_1, a_3, a_4, ...a_j)$ from the algorithmic constraint of the two FRB type in equation (.2). Firstly, we prove Lemma 1 and obtain a corollary that $\psi_{k_1,k_2..k_m} \subseteq \psi$. Further, using cardinality of union of sets property and Lemma 2 we obtain the corresponding original constraint.

Lemma 1: The extended compatible set of FRB type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})$ is a subset of extended compatible set of FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ where $1 \leq k$. Therefore, $\psi_{1,2,..k} \subseteq \psi$.

Proof:

a) Note that $\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})))$ is the set of BSs whose elements belong to $\{a_2, a_3..., a_{j+k}\}$, such that each of the element BSs interferes with the BSs a_i where, $1 \leq i \leq j + k$ (of course excluding itself).

b) $\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j)))$ is the set of BSs whose elements belongs to $\{a_2, a_3..., a_j\}$, such that each of the element BSs interferes with the BSs a_i where, $1 \leq i \leq j$ (of course excluding itself)

We know that $1 \leq k$.

Note that $\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})))$ is a subset of $\{a_2, a_3..., a_j\}$. This is because we assumed (in *Assumption 2*) every BS element in the set $\{a_{j+1}, a_{j+2}, .., a_{j+n}\}$ does not interfere with atleast one BS element of $\mathcal{O}(M(a_1 : a_2, a_3, a_4, ...a_j))$ (which itself is the subset of $\{a_2, a_3..., a_j\}$ from statement b. Therefore, further from statements a) and b),

$$\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k}))) \\ \subseteq \mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j))) \tag{.3}$$

The FRB types present in the extended compatible set of FRB type $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})$ are of the form $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k}, P(Y_k))$ where Y_k is the set of base stations that interfere with a_1 but do not interfere with at least one BS present in the set $\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_{j+k})))$.

Similarly, the FRB types present in the extended compatible set of FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ are of the form $(a_1 : a_2, a_3, a_4, ...a_j, P(Y))$ where Y is the set of base stations that interfere with a_1 but do not interfere with at least one BS present in the set $\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j)))$. Therefore, following from equation (.3) we have $P(Y_k) \subseteq P(Y)$. Moreover, from the assumption (in *Assumption-2*) $\{a_{j+1}, a_{j+2}, a_{j+3}...., a_{j+n}\} \subseteq Y$. Since Y includes the set of BSs $\{a_{j+1}...a_{j+k}\}$, and $P(Y_k) \subseteq P(Y)$, we have $\psi_{1,2,..,k} \subseteq \psi$.

Hence, **Lemma 1 is proved**

Corollary 1: Suppose $BS_2 \subseteq BS_1$. Then

$$\mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, BS_1)))$$
$$\subseteq \mathcal{O}(M((a_1 : a_2, a_3, a_4, ...a_j, BS_2)))$$

Corollary 2:$\psi_{k1,k2,..,km} \subseteq \psi$

From the algorithmic constraint of FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and type $(a_2 : a_1, a_3, a_4, ...a_j)$, it follows that

$$|\psi| = |\phi| \tag{.4}$$

Using Corollary 2 of Lemma 1, equation (.4) can be written as

$$\begin{aligned} &|\psi \cup \psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{12} \cup \psi_{1n} ... \cup \psi_{123} .. \cup \psi_{123..n}| \\ &= |\phi \cup \phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{12} \cup \phi_{1n} ... \cup \phi_{123} .. \cup \phi_{123..n}| \end{aligned} \tag{.5}$$

In equation (.5), LHS is the cardinality of union of physical FRBs of the types present in the extended compatible set of all the FRB types $(a_1 : a_2, a_3, a_4, ...a_j, a_{j+1}, ...a_j, P(Y))$, where $Y = \{a_{j+1}, ..., a_{j+n}\}$.

We need to obtain the original constraint for the FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and type $(a_2 : a_1, a_3, a_4, ...a_j)$ from (.5). Therefore, we need to get rid of FRB type elements which contain any of the BSs in the set $\{a_{j+1}, ..., a_{j+n}\}$, as these BSs interfere with both BS-a_1 and BS-a_2 which should not be present in the original constraint (.1) corresponding to FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and type $(a_2 : a_1, a_3, a_4, ...a_j)$. Following this motivation,

$$\begin{aligned} &|\psi \cup \psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{1,2} \cup \psi_{1,n} ... \cup \psi_{1,2,3} .. \cup \psi_{1,2,3..,n}| \\ &= |\psi \setminus \psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{1,2} \cup \psi_{1,n} ... \cup \psi_{1,2,3} .. \cup \psi_{1,2,3..,n}| \\ &\quad + |\psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{1,2} \cup \psi_{1,n} ... \cup \psi_{1,2,3} .. \cup \psi_{1,2,3..,n}| \end{aligned} \tag{.6}$$

Also

$$\begin{aligned} &|\phi \cup \phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{1,2} \cup \phi_{1,n} ... \cup \phi_{1,2,3} .. \cup \phi_{1,2,3..n}| \\ &= |\phi \setminus \phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{1,2} \cup \phi_{1,n} ... \cup \phi_{1,2,3} .. \cup \phi_{1,2,3..n}| \\ &\quad + |\phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{1,2} \cup \phi_{1,n} ... \cup \phi_{1,2,3} .. \cup \phi_{1,2,3..,n}| \end{aligned} \tag{.7}$$

The first term in the RHS of equation (.6) is equivalent to $\sum_{\mathbf{H'} \in P(\mathbf{H1})} N(a_1 : a_2, a_3.., a_j\mathbf{H'})$ where **H1** denotes the set of BSs which interfere with BS-a_1 but not BS-a_2. The first term in the RHS of equation (.7) is equivalent to $\sum_{\mathbf{H'} \in P(\mathbf{H1})} N(a_2 : a_1, a_3.., a_j\mathbf{H'})$. **H2** denotes the set of BSs which interfere with BS-a_2 but not with BS-a_1. Therefore, if we are able to prove the equality of second term in the RHS of equation (.6) and second term in the RHS of equation (.7), using (.5) we obtain the desired original constraint for FRB type $(a_1 : a_2, a_3, a_4, ...a_j)$ and type $(a_2 : a_1, a_3, a_4, ...a_j)$.

The problem of obtaining the desired original constraint is reduced to proving

$$\begin{aligned} &|\psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{1,2} \cup \psi_{1,n} ... \cup \psi_{1,2,3} .. \cup \psi_{1,2,3,..n}| \\ &= |\phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{1,2} \cup \phi_{1,n} ... \cup \phi_{1,2,3} .. \cup \phi_{1,2,3..,n}| \end{aligned} \tag{.8}$$

By the cardinality of union of sets property, for any n sets $A_1; A_2 ...; A_n$ we know that :-

$$\begin{aligned} |\cup_{i=1}^{n} A_i| &= \textstyle\sum_{i=1}^{n} |A_i| - \sum_{1 \le i < j \le n}^{n} |A_i \cap A_j| \\ &+ \textstyle\sum_{1 <= i < j < k <= n}^{n} |A_i \cap A_j \cap A_k| ... \\ &+ (-1)^{n-1} |\cap_{i=1}^{n} A_i| \end{aligned} \tag{.9}$$

Let $\psi^1 = \psi_1$,...,$\psi^n = \psi_n$,$\psi^{n+1} = \psi_{1,2}$,...,$\psi^{2^n} = \psi_{1,2,3,4..,n}$

Similarly, let $\phi^1 = \phi_1$,...,$\phi^n = \phi_n$, $\phi^{n+1} = \phi_{1,2}$,...,$\phi^{2^n} = \phi_{1,2,3,4..n}$.

Using equation (.9)

$$\begin{aligned}\left|\cup_{i=1}^{2^n}\psi^i\right| = \textstyle\sum_{i=1}^{2^n}\left|\psi^i\right| - \sum_{1\leq i<j\leq 2^n}^{2^n}\left|\psi^i \cap \psi^j\right| \\ +\textstyle\sum_{1<=i<j<k<=2^n}^{2^n}\left|\psi^i \cap \psi^j \cap \psi^k\right| ... \\ +(-1)^{2^n-1}\left|\cap_{i=1}^{2^n}\psi^i\right|\end{aligned} \tag{.10}$$

$$\begin{aligned}\left|\cup_{i=1}^{2^n}\phi^i\right| = \textstyle\sum_{i=1}^{2^n}\left|\phi^i\right| - \sum_{1<=i<j<=2^n}^{2^n}\left|\phi^i \cap \phi^j\right| \\ +\textstyle\sum_{1<=i<j<k<=2^n}^{2^n}\left|\phi^i \cap \phi^j \cap \phi^k\right| ... \\ +(-1)^{2^n-1}\left|\cap_{i=1}^{2^n}\phi^i\right|\end{aligned} \tag{.11}$$

Lemma 2: Each ψ term in the RHS of equation (.10) can be mapped to the corresponding ϕ term in in the RHS of equation (.11), i.e., $|\psi^{i_1} \cap \psi^{i_2} .. \cap \psi^{i_k}| = |\phi^{i_1} \cap \phi^{i_2} .. \cap \phi^{i_k}|$ where $1 \leq i_1 < i_2 .. < i_k \leq 2^n$

Proof: Suppose ψ^{i_m} is the set of physical FRBs of the types present in the extended compatible set of FRB type $(a_1 : a_2 ... a_j, BS_{i_m})$, for $1 \leq m \leq k$ where $BS_{i_m} \subseteq \{a_{j+1}, ..., a_{j+n}\}$. The FRB types in the set ψ^{i_m} are of the form $(a_1 : a_2 ... a_j, BS_{i_m}, P(Y_{i_m}))$. Let $BS_i = \cup_{m=1}^{k} BS_{i_m}$. Let the FRB type A is $(a_1 : a_2 ... a_j, BS_i)$ and FRB type B is$(a_2 : a_1 ... a_j, BS_i)$. Note that if $\cap_{m=1}^{k}\psi^{i_m} \neq \phi$ then this set contains the FRB of type $(a_1 : a_2 ... a_j, BS_i, P(X))$. From the definition of X, we have $X = \cap_{m=1}^{k} Y_{i_m}$. X can be partitioned into X_1 and X_2 such that $X = X_1 \cup X_2$, where X_1 is the set of BSs which are contained in the set X and interfere with every element of $\mathcal{O}(M(A))$. X_2 is the set of BSs which do not interfere with at least one element of $\mathcal{O}(M(A))$.

$\mathcal{O}(M((a_1 : a_2...a_j, BS_i))) \subseteq \mathcal{O}(M((a_1 : a_2...a_j, BS_{i_m})))$, $\forall m$ from corollary of Lemma 1.

Further,

$$\mathcal{O}(M((a_1 : a_2...a_j, BS_i))) \subseteq \cap_{m=1}^{k} \mathcal{O}(M((a_1 : a_2...a_j, BS_{i_m}))).$$

Physical FRBs of types present in the extended compatible set of A are represented by EX(A), which contains the FRBs of type $(a_1 : a_2...a_j, BS_i, P(X_2))$. Since, $\cap_{m=1}^{k} \psi^{i_m}$ contains the FRBs of type $(a_1 : a_2...a_j, BS_i, P(X))$ and $X_2 \subseteq X$, we have $EX(A) \subseteq \cap_{m=1}^{k} \psi^{i_m}$.

Therefore,

$$\begin{aligned} |\psi^{i_1} \cap \psi^{i_2} .. \cap \psi^{i_k}| = & |\psi^{i_1} \cap \psi^{i_2} .. \cap \psi^{i_k} \setminus EX(A)| \\ & + |EX(A)| \end{aligned} \tag{.12}$$

Similarly,

$$\begin{aligned} |\phi^{i_1} \cap \phi^{i_2} .. \cap \phi^{i_k}| = & |\phi^{i_1} \cap \phi^{i_2} .. \cap \phi^{i_k} \setminus EX(B)| \\ & + |EX(B)| \end{aligned} \tag{.13}$$

This is because, $|EX(A)| = |EX(B)|$ using the algorithmic constraint for FRB type A and type B. To prove lemma 2 our aim is to prove:

$$\left| \cap_{m=1}^{k} \psi^{i_m} \setminus EX(A) \right| = \left| \cap_{m=1}^{k} \phi^{i_m} \setminus EX(B) \right| \tag{.14}$$

For a FRB to belong to the set $\cap_{m=1}^{k} \psi^{i_m}$ but not belong to $EX(A)$, the FRB should be of type $(a_1 : a_2...a_j, BS_i, Z)$ where set Z($Z \subseteq X$) contains at least one BS which interferes with every element of $\mathcal{O}(M(A))$. At least one element of Z should belong to the set X_1 and the rest of the elements may belong to X_2. We try to classify all such FRBs which belong to $\cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$ and then use the algorithmic constraint to show (.14).

Suppose, BS-v interferes with every element of $\mathcal{O}(M(A))$ and $v \in X_1$, FRB of type A_1 $(a_1 : a_2...a_j, BS_i, v) \in \cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$. Also note that $\mathcal{O}(M(A)) = \mathcal{O}(M(A_1)$ as BS- v interferes with every owner of $\mathcal{O}(M(A))$. Therefore, every owner BS in the set $\mathcal{O}(M(A))$ also owns the corresponding matching in $M(A_1)$. Since X_2 is the set of BS which do not interfere with at least one element of $\mathcal{O}(M(A))$, the extended compatible set of FRB type A_1 contains the FRBs of the type $(a_1 : a_2...a_j, BS_i, v, P(X_2))$. If physical FRBs of type $A_1 \in \cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$ then $EX(A_1) \in \cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$.

Now we can follow the same arguments for $V_1 \in P(X_1)$ (same argument for v in previous paragraph) and the type of FRBs $(a_1 : a_2...a_j, BS_i, V_1))$. The physical FRB of type $(a_1 : a_2...a_j, BS_i, V_1) \in \cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$. Moreover, all the FRB types present in the extended compatible set of this FRB type are of the form $(a_1 : a_2...a_j, BS_i, V_1, P(X_2)) \in \cap_{m=1}^{k}\psi^{i_m} \setminus EX(A)$. Also note that $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_1)) = \mathcal{O}(M(A))$.

Suppose, $V_2 \in P(X_1)$ and $V_2 \neq V_1$. We have $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_2)) = \mathcal{O}(M(A))$. This is because, $V_2 \neq V_1$, and all elements of V_2 and V_1 interfere with $\mathcal{O}(M(A))$, and $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_2)) = \mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_1)) =$

$O(M(A))$. If $V_2 \subseteq V_1$, then the physical FRBs in the set $EX((a_1 : a_2...a_j, BS_i, V_2) \cap EX((a_1 : a_2...a_j, BS_i, V_1)$ are of the type $(a_1 : a_2...a_j, BS_i, V_1, ..)$. But every element of $V_1 \backslash V_2$ interferes with every element of $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_2))$. Therefore, FRB type $(a_1 : a_2...a_j, BS_i, V_1, ..)$ cannot be present in the set $EX((a_1 : a_2...a_j, BS_i, V_2)$. We have $EX((a_1 : a_2...a_j, BS_i, V_2) \cap EX((a_1 : a_2...a_j, BS_i, V_1) = \phi$. If neither of the sets V_1 or V_2 is a subset of the other, then also

$$EX((a_1 : a_2...a_j, BS_i, V_2) \cap EX((a_1 : a_2...a_j, BS_i, V_1) = \emptyset \tag{.15}$$

since each element BS of the set $V_1 \setminus V_2$ interferes with every element of $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_2))$ and each element BS of the set $V_2 \setminus V_1$ interferes with every element of $\mathcal{O}(M(a_1 : a_2...a_j, BS_i, V_1))$. Neither of the sets $EX((a_1 : a_2...a_j, BS_i, V_2))$ or $EX((a_1 : a_2...a_j, BS_i, V_2))$ can have an element FRB type $(a_1 : a_2...a_j, BS_i, V_2 \cup V_1, ..)$. Similarly by symmetry, $EX((a_2 : a_1...a_j, BS_i, V_2) \cap EX((a_2 : a_1...a_j, BS_i, V_1) = \phi$. By algorithmic constraints, we know that

$$|EX(a_1 : a_2...a_j, BS_i, V)| = |EX(a_2 : a_1...a_j, BS_i, V)| , \forall V \in P(X_1) \tag{.16}$$

The FRB types present in the extended compatible set of FRB type $(a_1 : a_2...a_j, BS, V)$ are of the form $(a_1 : a_2...a_j, BS, V, P(X_2))$. We also know from (.15) that the extended compatible sets for distinct V_1 and V_2 are non-intersecting. Therefore, if we add (.16) for all the $V \in P(X_1)$ we obtain

$$\begin{aligned} &\textstyle\sum_{V \in P(X_1), S \in P(X_2)} N(a_1 : a_2 ... a_j, BS_i, V, S) \\ = &\textstyle\sum_{V \in P(X_1), S \in P(X_2)} N(a_2 : a_1 ... a_j, BS_i, V, S) \end{aligned} \tag{.17}$$

Since $X_1 \cup X_2 = X$ and $\cap_{m=1}^{k} \psi^{i_m}$ contains the FRB of type $(a_1 : a_2 ... a_j, BS_i, P(X))$ we obtain:-

$$|\psi^{i_1} \cap \psi^{i_2} .. \cap \psi^{i_k}| = |\phi^{i_1} \cap \phi^{i_2} .. \cap \phi^{i_k}| \tag{.18}$$

Hence, **Lemma 2 is proved**.

From Lemma 2 and equations (.10) and (.11), the equation (.8) follows. This proves

$$\begin{aligned} &|\psi \setminus \psi_1 \cup \psi_2 .. \cup \psi_n \cup \psi_{12} \cup \psi_{1n} ... \cup \psi_{123} .. \cup \psi_{123..n}| \\ = &|\phi \setminus \phi_1 \cup \phi_2 .. \cup \phi_n \cup \phi_{12} \cup \phi_{1n} ... \cup \phi_{123} .. \cup \phi_{123..n}| \end{aligned} \tag{.19}$$

We obtain the desired original constraint corresponding to FRB type $(a_1 : a_2, a_3, a_4, ...a_j$ and type $(a_2 : a_1, a_3, a_4, ...a_j)$ from the algorithmic constraint set.

b) *From original constraint set to the corresponding algorithmic constraint.*

We need to prove that we can obtain the algorithmic constraints for the FRB types $(a_1 : a_2 ... a_j)$ and $(a_2 : a_1 ... a_j)$ from the set of corresponding original constraints. **H3** denotes the set of BSs that interfere with BS-a_1 but does not interfere with at least one BS element of $\mathcal{O}(M((a_1 : a_2 ... a_j)))$. Similarly, **H4** denotes the set of BSs which interfere with BS a_2 but does not interfere with at least one BS element of $\mathcal{O}(M((a_2 : a_1 ... a_j)))$. The algorithmic constraint we aim to obtain is given by

$$\sum_{\mathbf{H'} \in P(\mathbf{H3})} N(a_1 : a_2, ..a_j, \mathbf{H'}) = \sum_{\mathbf{H'} \in P(\mathbf{H4})} N(a_2 : a_1, ..a_j, \mathbf{H'})$$

We first approach the problem by partitioning the set **H3** into two sets of BSs. The first one is a set of BSs in which every element BS interferes with both a_1 and a_2 and each element of the other set of BSs interferes with a_1 but not a_2. We do the same thing for **H4**. Let $H3_x$ be the set of BSs which interfere with a_1 but do not interfere with a_2. Further, $H3_y$ is the set of BSs which interfere with both a_1 and a_2 and $H3_x \cup H3_y = \mathbf{H3}$. Similarly, $H4_x$ be the set of BSs which interfere with a_2 but do not interfere with a_1. Further, $H4_y$ is the set of BSs which interfere with both a_2 and a_1 and $H4_x \cup H4_y = \mathbf{H4}$. Note that from the definition of **H3** and **H4**, the BSs which interfere with both a_1 and a_2 are same in both sets **H3** and **H4**. Therefore $H3_y = H4_y$.

Our approach is to use the original constraints corresponding to FRBs of type $(a_1 : a_2, ..a_j, P(H3_y))$ and $(a_2 : a_1, ..a_j, P(H4_y))$ and then add them up to obtain the algorithmic constraint. The set of original constraints is given by

$$\begin{aligned} &\sum_{S \in P(H3_x)} N(a_1 : a_2, ..a_j, V, S) \\ &= \sum_{S \in P(H4_x)} N(a_2 : a_1, ..a_j, V', S); \forall V \in P(H3_y), V = V'. \end{aligned} \tag{.20}$$

Adding all the original constraints in the equation (.20) for $\forall V \in P(H3_y)$. We obtain

$$\sum_{V \in P(H3_y), S \in P(H3_x)} N(a_1 : a_2, ..a_j, V, S)$$

$$= \sum_{V' \in P(H4_y) S \in P(H4_x)} N(a_2 : a_1, ..a_j, V', S); V = V'. \tag{.21}$$

Since $H3_x \cup H3_y = \mathbf{H3}$ and $H4_x \cup H4_y = \mathbf{H4}$. (.21) is the same as (.20). Hence, we have obtained the algorithmic constraint (.20) from the set of original constraints in (.20). $\square$

Appendix B

Recall that each BS is interested in allocating the FRBs in the *agreed list* at the k-th priority level to its users. Let us denote the FRB of type t allocated to user u by $y_{u,t}$, and define $\mathbf{y}_u = (y_{u,t}; t \in \mathcal{T}_b)$ and $\mathbf{y} = (\mathbf{y}_u; u \in \mathcal{N}_b)$. Similarly, let

$$\tilde{y}_{u,t} = \begin{cases} req'_u / r_{u,t} \text{ if } t = L_u(k), \\ \quad 0 \qquad \text{otherwise,} \end{cases}$$

$\tilde{\mathbf{y}}_u = (\tilde{y}_{u,t}; t \in \mathcal{T}_b)$, and $\tilde{\mathbf{y}} = (\tilde{\mathbf{y}}_u; u \in \mathcal{N}_b)$. Then, we can allocate the available FRBs according to the solution to the following optimization problem:

$$\text{minimize}_{\mathbf{y} \geq \mathbf{0}} \qquad ||\mathbf{y} - \tilde{\mathbf{y}}||_2^2$$

$$\text{subject to algorithmic constraints in (5)}$$

It is clear that the objective function is a convex function of $\mathbf{y}$, and the feasible set is a convex set because if both $\mathbf{y}_1$ and $\mathbf{y}_2$ satisfy the algorithmic constraints, then so does any convex combination of $\mathbf{y}_1$ and $\mathbf{y}_2$. Hence, this optimization problem can be solved efficiently using convex optimization tools and is guaranteed to produce a solution that meets the algorithmic constraints.

This optimization can also be changed in different ways. For instance, we can weigh entries in the difference vector by different constants. Similarly, we can impose a constraint so that no user is allocated more FRBs than necessary to meet its minimum throughput requirement.

Appendix C

We capture some basic definitions that are used in the dissertation

Given a ground set Ω, we define its power set (i.e., the set containing all the subsets of Ω) as 2^Ω. Then, a real-valued function defined on the subsets of Ω, $h : 2^\Omega \to R$ is called a *submodular* set function if and only if

$$h(B \cup a) - h(B) \leq h(A \cup a) - h(A),$$
$$\forall\, A \subseteq B \subseteq \Omega \;\&\; a \in \Omega \setminus B$$

A real valued set function $h : 2^\Omega \to R$ is a *monotonic* non-decreasing set function if and only if it satisfies, $h(A) \leq h(B)$, $\forall\, A \subseteq B \subseteq \Omega$.

Bibliography

[1] Picocell mesh : Bringing low-cost coverage, capacity and symmetry to mobile WiMAX. Tropos Networks.

[2] Wireless backhaul solutions for small cells. Ceragon Networks White Paper.

[3] Mobile Backhaul: Fiber vs. Microwave. Ceragon Networks White paper, 2009.

[4] Summary of the description of candidate eicic solutions. R1-104968 3GPP Std., 2010.

[5] E-utra and e-utran overall description. TS36.300 V10.8.0,, 2012.

[6] Study on small cell enhancements for e-utra and e-utran physical-layer aspects. TR36.872 V12.0.0, 2013.

[7] 3GPP. Evolved universal terrestrial radio access (E-UTRA) - further advancements for (E-UTRA) physical layer aspects. *TR36.814 V9.0.0*, Mar. 2010.

[8] 3GPP. Study on small cell enhancements for E-UTRA and E-UTRAN physical-layer aspects. *TR36.872 V12.0.0*, Sept. 2013.

[9] H. Al-Shatri, S.Shi, and T.Weber. Optimizing power allocation in interference channels using d.c. programming. In *Proc. 2010 Int. Symp. Modeling Optimization Mobile, Ad Hoc Wireless Netw.*, pages 380–386.

[10] S. Ali and V. Leung. Dynamic frequency allocation in fractional frequency reused ofdma networks. *IEEE Trans.on Wireless Comm.*, 8(8):4286–4295, Aug. 2009.

[11] M. Y. Arslan, J. Yoon, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee. Fermi: a femtocell resource management system forinterference mitigation in ofdma networks. In *Proc. MobiCom'11.*

[12] K. Balachandran, J. H. Kang, K. Karakayali, and K. Rege. Cell selection with downlink resource partitioning in heterogeneous networks. In *Proc. IEEE Workshop Int. Conf. on Comm.*, June 2011.

[13] A. Bedekar and R. Agrawal. Optimal muting and load balancing for eICIC. In *Proc. IEEE WiOPT*, 2013.

[14] S. Borst, S. Hanly, and P. Whiting. Throughput utility optimization in hetnets. In *IEEE VTC*, 2013.

[15] S. Borst, M. Markakis, and I. Saniee. Distributed power allocation and user assignment in ofdma cellular networks. In *Proc. 43rd Allerton*, 2011.

[16] T. Bu, L. Li, and R. Ramjee. Generalized proportional fair scheduling in third generation wireless data networks. *IEEE Infocom*, 2006.

[17] G. Cao, D. Yang, R. An, X. Ye, R. Zheng, X. Zheng, and X. Zhang. An adaptive sub-band allocation scheme for dense femtocell environment. In *Proc. IEEE Wireless Comm. and Net. Conf.*, pages 102–107, Mar. 2011.

[18] I. Caragiannis, A. Fishkin, C. Kaklamanis, and E. Papaioannou. A tight bound for online coloring of disk graphs. *Structural Information and Communication Complexity*, 3499:78–88, 2005.

[19] V. Chandrashekhar and J. G. Andrews. Spectrum allocation in tiered cellular networks. *IEEE Trans.on Comm.*, 57(10):3059–3068, Oct. 2009.

[20] V. Chandrashekhar, J. G. Andrews, and A. Gatherer. Femtocell networks: A survey. *IEEE Commun. Mag.*, 46(9):59–67, Sep. 2008.

[21] C. Chen and F. Baccelli. Self-optimization in mobile cellular networks: Power control and user association. In *Proc. IEEE ICC*, 2010.

[22] R. Chen and R. B. et. al. Single and multi-cell scheduling in coordinated multi-point distributed antenna systems. In *Proc. IEEE Global Telecomm. Conf.*, Dec. 2012.

[23] Y. Cheng, M. Pesavento, and A. Philipp. Joint network optimization and downlink beamforming for comp transmissions using mixed integer conic programming. *IEEE Trans. on Signal Processing*, 61(16), 2013.

[24] S. Christensen, R. Agarwal, E. Carvalho, and J. Cioffi. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design. *IEEE Trans. Wireless Commun.*, 7(12):1–8, 2008.

[25] H. Claussen. Femtocell coverage optimization usingswitched multi element antenna. *IEEE ICC, Dresden,Germany*, June 2009.

[26] T. P. Dinh and L. T. H. An. Convex analysis approach to d.c. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

[27] K. Eriksson, S.Shi, N. Vucic, M.Schubert, and E. Larsson. Global optimal resource allocation for achieving maximum weighted sum rate. In *Proc. 2010 IEEE Global Telecomm. Conf.*

[28] T. Erlebach and J. Fiala. On-line coloring of geometric intersection graphs. *Elsevier, Computational Geometry*, 23:243–255, Sept. 2002.

[29] S. W. et al. Transmission capacity of wireless ad hoc networks with successive interference cancellation. *IEEE Trans. Info. Theory*, 53(8):2799–2814, Aug 2007.

[30] G. Fodor, C. Koutsimanis, A. Racz, N. Reider, A. Simonsson, and W. Muller. Intercell interference coordination in ofdma networks and in 3gpp long term evolution system. *IEEE Journal on selected areas in comm.*, 4(7), Aug. 2009.

[31] A. Ghosh, A. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visosky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan. Heterogeneous Cellular Networks: From Theory to Practice. *IEEE Comm. Mag.*, 50(6):54–64, June 2012.

[32] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed. *Fundamentals of LTE*. Prentice Hall., 2011.

[33] P. Goundan and A. Schulz. Revisiting the greedy approach to submodular set function maximization. *manuscript*, June 2007.

[34] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[35] R. Iyer and J. Bilmes. Algorithms for approximate minimization of the difference between submodular functions. In *Proc. 2012 Uncertainity in Artificial Intelligence*, August 2012.

[36] R. Iyer and J. Bilmes. The sub modular bregman and lovasz-bregman divergences with applicationns. In *NIPS*, 2012.

[37] R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *30th Proc. International Conf. on Machine Learning*, 2013.

[38] S. Jegelka and J. Bilmes. Submodularity beyond sub modular energies: coupling edges in graph cuts. In *CVPR*, 2011.

[39] H. Kha, H. Tuan, and H.Nguyen. Fast global optimal power allocation in wireless networks by local d.c. programming. *IEEE Wireless Comm.*, 11(2), Feb. 2012.

[40] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *STOC*, 2009.

[41] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *Proc. IEEE INFOCOM*, 2005.

[42] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji. Cell association and interference coordination in heterogeneous LTE-A cellular networks. *IEEE Journal on Selected areas in Comm.*, 28(9):1479–1489, Dec. 2010.

[43] M.Chiang, C. Tan, D. Palomar, D. Neill, and D. Julian. Power control by geometric programming. *IEEE Trans.on Wireless Comm.*, 6(7), July 2007.

[44] M. Narasimhan and J. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv*, Jul. 2012.

[45] M. Necker. Towards frequency reuse 1 cellular FDM/TDM systems. In *Proc. ACM, MSWiM.*, Oct. 2006.

[46] M. Necker. Local interference coordination in cellular OFDMA networks. In *Proc. IEEE VTC*, pages 1741–1746, Oct. 2007.

[47] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Math. Operations Research*, 1978.

[48] Y. Peng and F. Qin. Exploring het-net in LTE-Advanced system. In *Proc. IEEE Workshop Int. Conf. on Comm.*, June 2011.

[49] D. Perez, I. Guvenc, G. Roche, M. Kountouris, T. Quek, and J. Zhang. Enhanced intercell interference coordination challenges in heterogeneous networks. *IEEE Wireless Communications*, 18(3):22–30, 2011.

[50] D. Perez, A. Juttner, and J. Zhang. Dynamic frequency planning versus frequency reuse schemes in OFDMA networks. In *Proc. IEEE VTC*, pages 1–5, April. 2009.

[51] D. L. Perez, A. Ladanyi, A. Juttner, H. Rivano, , and J. Zhang. Optimization method for the joint allocation of modulation schemes, coding rates, resource blocks and power in self-organizing lte networks. In *Proc. IEEE Infocom. Conf.*, Apr. 2011.

[52] D. L. Perez, A. Valcarce, G. de la Roche, and J. Zhang. OFDMA femtocells: A roadmap on interference avoidance. *IEEE Commun. Mag.*, 47(9):41 – 48, Sep. 2009.

[53] S. Sardellitti, G. Scutari, and S. Barbarossa. Joint cell selection and radio resource allocation in mimo small cell networks via successive convex approximation. In *Proc. 2014 IEEE ICASSP*.

[54] Y. Shi, A. B. MacKenzie, L. A. Dasilva, K. Ghaboosi, and M. Latva-aho. On resource reuse for cellular networks with femto and macrocell coexistence. In *Proc. IEEE Global Telecomm. Conf.*, Dec. 2010.

[55] I. Siomina and D. Yuan. Analysis of cell load coupling for LTE network planning and optimization. *IEEE Trans. on Wireless Comm.*, June 2012.

[56] K. Son, S. Chong, and G. Veciana. Dynamic association for load balancing and interference avoidance in multi-cell networks. *IEEE Trans. Wireless Comm.*, 2009.

[57] A. L. Stolyar. On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation. *Operations Res.*, 2005.

[58] A. Stoylar and H. Viswanath. Self-organizing dymanic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination. In *Proc. IEEE INFOCOM*, 2009.

[59] K. Sundaresan and S. Rangarajan. Efficient resource management in OFDMA femtocells. In *Proc. ACM Intl. Symp. Mobile Ad Hoc Netw. Comput.*, pages 33–42, May 2009.

[60] A. Tajer, N. Prasad, and X. Wang. Robust linear precoder design for multi-cell downlink transmission. *IEEE Trans. on Signal Processing*, Jan. 2011.

[61] C. Tan, S. Friedland, and S. Low. Spectrum management in multi-user cognitive wireless networks: optimality and algorithm. *IEEE Journal on Selected areas in Comm.*, 29:421–430, Feb. 2011.

[62] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, New York, NY, USA, 2005.

[63] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews. User association for load balancing in heterogeneous cellular networks. *IEEE Trans. on Wireless Comm.*, June 2013.

[64] A. Yuille and A. Rangarajan. The concave-convex procedure (cccp). In *Advances in Neural Information Processing Systems 2002*.

[65] J. Zhao, T. Quek, and Z. Lei. Coordinated multipoint transmission with limited backhaul data transfer. *IEEE Trans. Wireless Comm.*, 12(6), Jun. 2013.