# Admission Control and Radio Resource Allocation for Multicasting over High Altitude Platforms

by

Ahmed Mohamed Ali Ibrahim

A Thesis submitted to The Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

2016

**Abstract**

High Altitude Platforms (HAPs), are quasi-stationary aerial platforms that carry a wireless communications payload to provide wireless communications and broadband services. They are meant to be located in the stratosphere layer of the atmosphere at altitudes in the range 17-22 km and have the ability to fly on demand to temporarily or permanently serve regions with unavailable telecommunications infrastructure. In this thesis, optimization techniques are used for finding good solutions for a novel problem that considers user admission control and radio resource allocation in multicasting over OFDMA based HAPs. Power, frequency, space and time are the resources considered in the problem. These many different aspect combinations of the problem in multicasting over an OFDMA HAP system were not, to the best of our knowledge, addressed before. Due to the strong dependence of the total number of users that could join different multicast groups on the different way we allocate resources to the different multicast groups, it is of significant importance to consider a joint user to session assignments and radio resource management across the groups. From the service provider's point of view, it would be in its best interest to be able to admit as many users as possible, while satisfying their quality of service requirements.

The problem turns out to be mixed integer non-convex non-linear program for which branch and bound solution framework is guaranteed to solve the problem. Branch and bound can be also used to obtain sub-optimal solutions with known goodness. Even though branch and bound is guaranteed to find the optimal solution, the computational cost could be too high. Our work focuses on how to enhance the performance of the branch and bound algorithm such that the computational

effort and the tree size (which affects the memory requirements) are reduced. Two key things we rely on in this thesis for improving computational performance are reformulations and relaxations. In reformulations we reduce the problem into good structured problems for which efficient algorithms have been developed in the literature, which we use with some modification to suit our problem. Different relaxations are used to obtain good bounds that are used for early pruning of nodes in the branch and bound tree.

First, a system model in a multicellular high altitude platform system is considered, in which each user can receive any requested multicast session in its cell from no more than only one HAP antenna simultaneously. All the users have equal priority for admission. The users are selected to join the respective multicast groups and the power, subchannels and time slots are allocated such that the spectrum utilization is maximized while satisfying the quality of service requirements. We refer to this system model as the *primary system model*. Since Lagrangian relaxation is known to obtain good bounds for mixed integer non-linear programs, we use it with the subgradient algorithm to obtain solution bounds for the primary system model problem formulation. These bounds are used in the branch and bound algorithm for pruning of nodes. The tighter the bounds, the faster the branch and bound algorithm is, and the smaller the search tree is due to early pruning. The numerical results illustrate the goodness of the bounds for different constraint set dualizations and for different subgradient step size rules.

The system model is then extended to allow the multicast group users to receive a session's transmission from more than one antenna simultaneously at different frequencies. This also allows the user to receive multicast sessions transmitted in neighboring cells too, not just those transmitted in the cell which the user resides in.

The users have different priority levels of admission and the objective is to maximize the admission of highest priority users to the system. A much efficient formulation is obtained for the extended model in terms of size, as compared to the primary model. Linear outer approximation using McCormick underestimators are used for the relaxation of the mixed binary quadratically constrained problem. The solution method is based on branch and cut scheme in which cutting planes, domain propagation and heuristics are integrated. Various branching schemes are considered and a presolving reformulation linearization scheme for a specific set of quadratic constraints is considered. The numerical experiments compare the performances in terms of the duality gap, number of nodes, number of iterations, the number of iterations per node, the time needed to obtain the first feasible solution and the percentage of instances a feasible solution was found.

# Acknowledgements

This thesis with all its results would have not been achieved without God's help and care. Thanks to God for giving me the ability and strength to complete this work. I would like to express my special appreciation and thanks to my advisor Professor Dr. Attahiru Alfa, you have been an amazing mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a researcher. Your advice on both research as well as on my career have been priceless. I would also like to thank my committee members and examiners, professor Pradeepa Yahampath, professor Sherif Sherif, professor Subramaniam Balakrishnan and professor Xuemin Shen for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

I would like to thank my beloved wife for all her patience and help to me, thank you for being by my side in all the difficult times. Also, I would like to express my gratitude to my dear parents for supporting me in all the phases of my education since. Without your help and prayers I wouldn't have got to where I am now. I would also like to thank all of my friends who supported me during my research, and motivated me to strive towards my goal. I acknowledge the financial support from NSERC. I would like to thank the staff at the Department of Electrical and Computer Engineering. Special thanks to Amy Dario for her kind help.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AC | Admission control |
| AC-RRA | Joint admission control and radio resource allocation |
| AVG | Average |
| BER | Bit-error-rate |
| BET | Best effort traffic |
| BnB | Branch and Bound |
| BLP | Binary Linear Program |
| BS | Base Station |
| CAC | Call Admission Control |
| CTMC | Continuous Time Markov Chains |
| CDMA | Code Division Multiple Access |
| CPICH | Common Pilot Channel |
| E-SysMod | Extended System Model |
| FKS | Fractional Knapsack |
| FDMA | Frequency Division Multiple Access |
| FACH | Forward Access Channel |
| GEO | Geostationary Earth Orbit |
| GoS | Grade of Service |
| HAP | High Altitude Platform |

| | |
|---|---|
| HL | Hierarchical Layering |
| IDT | Information Decomposition Techniques |
| ITU | International Telecommunications Institute |
| ITU-R | International Telecommunications Institute-Radiocommunications |
| KKT | Karush-Kuhn-Tucker |
| LCG | Least Channel Gain |
| LEO | Low Earth Orbit |
| LMAP | Low-medium-altitude platforms |
| LP | Linear Program |
| L.H.S | Left Hand Side |
| LOS | Line-of-sight |
| LTE | Long Term Evolution |
| MBMS | Multicast Broadcast Multimedia Services |
| MBLP | Mixed Binary Linear Program |
| MBPCP | Mixed Binary Polynomial Constrained Program |
| MBPC | Mixed Binary Polynimilal Constraint |
| MBQCP | Mixed Binary Quadratically Constrained Program |
| MDC | Multiple Description Coding |
| MILP | Mixed Integer Linear Program |
| MINLP | Mixed Integer Non Linear Program |
| MIQCP | Mixed Integer Quadratically Constrained Program |
| MMF | Max-Min Fairness |
| MSF | Multicast Subgroup Formation |
| NBS | Nash Bargaining Solution |
| NLP | Non Linear Program |

| | |
|---|---|
| OFDM | Orthogonal Frequency Division Multiplexing |
| OFDMA | Orthogonal Frequency Division Multiple Access |
| ORA | Optimal Resource Allocation |
| ORA-LF | Optimal Resource Allocation with Local Fairness |
| ORA-GF | Optimum Resource Allocation subject to Global Fairness |
| P-SysMod | Extended System Model |
| PTF | Proportional Throughput with Fairness |
| QoS | Quality of Services |
| RF | Radio Frequency |
| R.H.S | Right Hand Side |
| RRA | Radio Resource Allocation |
| RA-RM | Resource Reservation- Random Model |
| RR-TS | Resource Reservation, Traffic Selection |
| SCIP | Solving Constraint Integer Programs |
| SINR | Signal-to-interference-niose ratio |
| SRA-LF | Simplified Resource Allocation with Local Fairness |
| STM | Strict Throughput Maximization |
| TDD | Time Division Duplexing |
| UL | Uplink |
| UMTS | Universal Mobile Telecommunications System |
| UT | User Terminal |
| WCDMA | Wide-band CDMA |

# List of Symbols

| | |
|---|---|
| $A(d_k)$ | is the attenuation due to clouds and rain that user $k$ experiences. |
| $\mathcal{B}$ | Set of all binary variables in $\mathcal{HAP}^{Eff}_{MBQCP}$. |
| $\Delta B$ | subchannel bandwidth. |
| $\ddot{c}_{dual}$ | Dual bound of a BnB node relaxation. |
| $\ddot{c}_{primal}$ | Primal bound of a BnB node relaxation. |
| $\ddot{c}_{opt}$ | Objective function value for the formulation $\mathcal{HAP}^{Eff}_{MBQCP}$. |
| $c_{sbiterquot}$ | maximal fraction of strong branching LP iterations compared to node relaxation LP iterations. |
| $\check{C}_{light}$ | Speed of light |
| $CN_{\mathcal{HAP}_2^{Lagrange}}$ | Number of constraints in $\mathcal{HAP}_2^{Lagrange}$. |
| $CN_{\mathcal{HAP}^{Eff}_{MBQCP}}$ | Number of constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$. |
| $\mathscr{C}$ | set of subchannels in HAP service area. |
| $\overline{\mathcal{C}}$ | Set of generated cutting planes. |
| $\tilde{\mathcal{C}}_{LP}$ | Set of cutting planes added to an LP. |
| $\mathfrak{C}$ | Cloud of optimal solutions to an LP relaxation. |
| $d_k$ | distance between the HAP and user $k$. |
| $E_c$ | chip energy in a CDMA system |

| | |
|---|---|
| $\mathbf{e}$ | column vector of ones. |
| $F$ | OFDMA frame length. |
| $F_{cand}$ | set of branching candidates. |
| $F(\mathfrak{C})$ | set of branching candidates from the cloud of solutions $\mathfrak{C}$. |
| $\overline{F}$ | a subset of the branching candidates. |
| $\mathcal{FRAC}$ | set of variables of binary type that have fractional values. |
| $g_{i,k,c,t}$ | channel gain between antenna $i$ and user $k$ on frequency-time slot $(c,t)$. |
| $G_k^u$ | Antenna gain of user $k$. |
| $G_H(\varpi_{i,k})$ | HAP antenna gain for $\varpi_{i,k}$. |
| $\mathcal{H}_{distance}$ | Hamming distance between two vectors. |
| $\mathcal{HAP}^{Init}$ | Inital formulation for the P-SysMod optimization problem. |
| $\mathcal{HAP}_2^{Lagrange}$ | Final formulation obtained for P-SysMod. |
| $\mathcal{HAP}^{Eff}$ | Initial formulation for E-SysMod. |
| $\mathcal{HAP}_{MBQCP}^{Eff}$ | Final formulation obtained for E-SysMod. |
| $I_o$ | interference power density in a CDMA system |
| $\mathscr{K}$ | Set of user terminals in the HAP service area. |
| $\ddot{\mathcal{L}}$ | Branch and bound queue of nodes. |
| $\mathscr{M}$ | set of multicast sessions in the HAP service area. |
| $N^{BiL}_{\mathcal{HAP}_2^{Lagrange}}$ | Number of bilinear terms in $\mathcal{HAP}_2^{Lagrange}$. |
| $N^{BiL}_{\mathcal{HAP}_{MBQCP}^{Eff}}$ | Number of bilinear terms in $\mathcal{HAP}_{MBQCP}^{Eff}$. |
| $N_{m,i}$ | set of UTs that get admitted to multicast session $m$ in cell $i$ for a given OFDMA frame. |
| $p_{m,i,c,t}$ | the HAP power level in $\mathcal{HAP}_{MBQCP}^{Eff}$ and $\mathcal{HAP}_2^{Lagrange}$, for session $m$ on antenna $i$ over frequency-time slot $(c,t)$. |

| | |
|---|---|
| $\mathbf{p}$ | column vector of the variables $p_{m,i,c,t}$. |
| $\tilde{p}^{L}_{m,i,c,t}$ | Local (at a given node) lower bound of the variable $p_{m,i,c,t}$. |
| $\tilde{p}^{U}_{m,i,c,t}$ | Local (at a given node) upper bound of the variable $p_{m,i,c,t}$. |
| $P^{Total}_{PF}$ | Total HAP available power. |
| $\mathcal{Q}_{relax}$ | Relaxation of BnB node $\mathcal{Q}$. |
| $\mathcal{Q}^{Strengthened}_{relax}$ | Strengthened relaxation of BnB node $\mathcal{Q}$. |
| $r^{j}_{relax}$ | reduced cost of $\ddot{x}^{relax}_{j}$. |
| $r_{m,i,c,t}$ | data rate of session $m$ defined as Shannon's capacity on antenna $i$ on frequency-time $(c,t)$. |
| $R^{min}_{m}$ | Minimum data capacity for session $m$. |
| $R^{max}_{m}$ | Maximum data capacity for session $m$. |
| $\mathbb{R}$ | set of real numbers. |
| $\mathbb{R}^{+}$ | set of positive real. |
| $s^{branch}_{j}$ | Branching score for the decision variable $\ddot{x}_j$. |
| $s_{m,i}$ | set of users that are tuned to receive session $m$ in cell (HAP antenna) $i$. |
| $s_{cut}\left(\overline{eff}_c, \overline{orth}_c, \overline{par}_{cut}\right)$ | Score of a cut as a function of its efficacy, orthgonality and parellism. |
| $\mathscr{S}$ | Set of HAP antennas. |
| $\mathscr{T}$ | set of time slots in the OFDMA frame in HAP service area. |
| $\mathbf{u}$ | row vector of dual variables for the Lagrangian relaxation (LR) in Chapter 3. |
| $VN_{\mathcal{HAP}_2^{Lagrange}}$ | Number of variables in $\mathcal{HAP}_2^{Lagrange}$ |
| $VN_{\mathcal{HAP}^{Eff}_{MBQCP}}$ | Number of variables in $\mathcal{HAP}^{Eff}_{MBQCP}$ |
| $\ddot{\mathbf{x}}$ | Vector of decision variables in $\mathcal{HAP}^{Eff}_{MBQCP}$. |

| | |
|---|---|
| $\ddot{x}_j^{relax}$ | decision variable in $\mathcal{Q}_{relax}$. |
| $\ddot{\mathbf{x}}^{\tilde{L}}$ | local lower bounds of $\ddot{\mathbf{x}}$. |
| $\ddot{\mathbf{x}}^{\tilde{U}}$ | local upper bounds of $\ddot{\mathbf{x}}$. |
| $\ddot{\mathbf{x}}_{opt}^{\mathcal{HAP}_{MBQCP}^{Eff}}$ | Optimal solution for the formulation $\mathcal{HAP}_{MBQCP}^{Eff}$. |
| $\ddot{\mathbf{x}}_{relax}^{opt}$ | Optimal solution of $\mathcal{Q}_{relax}$. |
| $\ddot{\mathbf{x}}_{BFS}$ | The incumbent solution for $\mathcal{HAP}_{MBQCP}^{Eff}$ during the course of BnB algorithm. |
| $y_{m,i,c,t}$ | is a binary variable in $\mathcal{HAP}_{MBQCP}^{Eff}$, that indicates whether the trio combination $(i,c,t)$ is assigned for session $m$. |
| $z_{m,i,k,c,t}$ | is a binary variable in $\mathcal{HAP}^{Init}$ and $\mathcal{HAP}_2^{Lagrange}$ indicating whether UT $k$ receives a transmission for session $m$ in cell $i$ in a given OFDMA frame slot $(c,t)$. |
| $\mathbf{z}$ | column vector of the variables $z_{m,i,k,c,t}$. |
| $\mathbb{Z}$ | set of integers. |
| $\nabla^2 f$ | Hessian matrix for the function $f$. |
| $\hat{\gamma}_{SB}^{max}$ | number of simplex iterations for for the strong branching done in $\mathcal{Q}$. |
| $\hat{\gamma}_{LP}$ | is the number of regular simplex iterations. |
| $\gamma_{k,N_{m,i}}^{c,t}$ | SINR of user $k$ admitted to $N_{m,i}$ on frequency-time slot $(c,t)$. |
| $\Delta T$ | time slot duration. |
| $\varsigma_j^0$ | unit bound change obtained by downwards branching on $\ddot{x}_j$. |
| $\varsigma_j^1$ | unit bound change obtained by upwards branching on $\ddot{x}_j$. |
| $\ddot{\sigma}_j^0$ | aggregate bound change obtained by downwards branching on $\ddot{x}_j$. |
| $\ddot{\sigma}_j^1$ | aggregate bound change obtained by upwards branching on $\ddot{x}_j$. |

| | |
|---|---|
| $\sigma^2$ | is the additive white Gaussian noise power per subchannel. |
| $\eta_j^0$ | number of nodes on which $\ddot{x}_j$ was selected for downwards branching. |
| $\eta_j^1$ | number of nodes on which $\ddot{x}_j$ was selected for upwards branching. |
| $\eta_{rel}$ | A reliability threshold on the number of BnB nodes. |
| $\varpi_{i,k}$ | angle between user $k$ and the HAP antenna's boresight axis. |
| $\Psi_j^0$ | pseudocost for branching on $\ddot{x}_j$ downwards. |
| $\Psi_j^1$ | pseudocost for branching on $\ddot{x}_j$ upwards. |
| $\ddot{\phi}_j^1$ | total deductions by branching on $\ddot{x}_j$ upwards. |
| $\ddot{\phi}_j^0$ | total deductions by branching on $\ddot{x}_j$ downwards. |
| $\lambda_{m,k}$ | is a binary constant in $\mathcal{HAP}_{MBQCP}^{Eff}$, that indicates whether user $k$ requests to join session $m$. |
| $\phi_{m,k}$ | is a binary variable in $\mathcal{HAP}_{MBQCP}^{Eff}$, that indicates whether a user $k$ gets assigned to receive multicast session $m$. |
| $\theta_m$ | is a binary variable in $\mathcal{HAP}_{MBQCP}^{Eff}$, that indicates whether session $m$ receives any resources, or equivalenty, whether any user gets assigned to receive the session's transmission. |
| $\rho_{m,k}$ | is a constant in $\mathcal{HAP}_{MBQCP}^{Eff}$ that represents priority for user $k$ on session $m$, and is a positive integer. |
| $\varrho$ | Duality gap. |
| $\overline{\varrho}$ | bound Goodness. |
| $\varphi_{k,c,t}$ | is the Ricean small scale gain in frequency-time slot $(c,t)$ for user terminal $k$. |
| $\omega_l$ | step size in iteration $l$ for the subgradient algorithm given |

in Chapter 3.

# Publications

1. A.Ibrahim and A. Alfa "Multicasting Admission Control and Radio Resource Allocation over a HAP: A Branch and Bound Solution Framework submitted to IEEE Transactions on Communications and is under review.

2. A.Ibrahim and A. Alfa "Multicasting Admission Control and Radio Resource Allocation over a HAP: Applying Linearization and Branching Techniques submitted to IEEE Transactions on Aerospace and Electronic Systems and is under review.

3. A.Ibrahim and A. Alfa "Joint Admission Control-Radio Resource Allocation for Multicasting over High Altitude Platforms submitted to IEEE Transactions on Aerospace and Electronic Systems and is under review.

4. A. Ibrahim, and Attahiru S. Alfa. "Using Lagrangian relaxation for radio resource allocation in high altitude platforms." IEEE Transactions on Wireless Communications 14.10 (2015): 5823-5835.

5. A. Ibrahim, and Attahiru S. Alfa. "Solving binary and continuous knapsack problems for radio resource allocation over High Altitude Platforms." 2014 Wireless Telecommunications Symposium. IEEE, 2014.

6. A. Ibrahim, and Attahiru S. Alfa. "Radio resource allocation for multicast

transmissions over High Altitude Platforms." 2013 IEEE Globecom Workshops (GC Wkshps). IEEE, 2013.

# Chapter 1

# Introduction

## 1.1 The Emergence of High Altitude Platforms

Delivering high-capacity services over wireless medium presents challenges, since the spectrum is limited and the demand for its access is constantly growing. For terrestrial cellular networks, the solution is to decrease the transmission range of a base station (BS) and deploy more base stations which require backhaul interconnections. Clearly, this is a costly and difficult proposition, especially for areas with hostile geographical nature. This pressure on the radio spectrum requires moving higher in frequency to K/Ka bands (26-40 Ghz), which are less heavily congested and can provide significant bandwidth. The main problem with working in K/Ka bands is that line-of-sight (LOS) or quasi-LOS propagation is needed [1].

The visibility problem can be solved using satellite technology, which is a well-established alternative to terrestrial infrastructures that is able to serve wide areas with a cellular coverage, thus implementing frequency reuse paradigms. *Geostationary Earth Orbit* (GEOs) satellites are located at about 36 thousand kilometers away from the earth's surface. Due to the large distance from the earth's surface, GEOs

have huge antenna footprints that can cover entire continents providing services to millions of users. However, being far away from the earth's surface also has major drawbacks, mainly due to the very critical free-space path loss and large propagation delays. These problems require large antennas and sophisticated architectures and protocols at the customer receivers. Furthermore, technological constraints for on-board antennas prevent the possibility of optimizing the cell dimension on the ground, thus potentially lowering frequency reuse efficiency and, consequently, overall capacity.

These problems can be partially solved with the use of *Low Earth Orbit* (LEO) satellites; however, these suffer from the widely investigated issues related to the rapid appearance and disappearance in the sky portion visible to the receiver. These issues require that LEO-based systems must include an efficient handover protocol among cells and satellites.

A potential solution for these problems that has been adopted is carrying communications relay payloads and operating in a quasi-stationary position in the stratosphere layer of the atmosphere. LOS propagation paths can be provided to most users, with modest free space path loss and propagation delays, thus enabling services that take advantage of the best features of both wireless terrestrial and satellite communications. The platforms that carry these payloads were called *high altitude platforms* (HAPs) [2].

## 1.2 An Overview on HAPs

HAPs are quasi-stationary aerial platforms that are meant to be located at a height of 17-22 km above Earth's surface in the stratosphere layer. Many of their pros are a combination of those in both, terrestrial wireless and satellite communication

systems. Some of those pros are [3]:

- Their ability to fly on demand to temporarily or permanently serve regions with unavailable telecommunications infrastructure.
- A single HAP has a large area coverage that can go up to 150 km compared to a single terrestrial cellular base station (BS) whose maximum radius (macro cells) is in the range of 20-30 km.
- Low propagation delays compared to satellites which implies better perceived quality of service QoS by the users for real time applications like voice and video.
- Stronger received signal strengths as compared to satellites and hence user terminals need not be bulky.
- Deployment time is low since one platform and ground support are sufficient to start the service.
- Much less ground-based infrastructure compared to terrestrial cellular networks.

For the same allocated bandwidth in a specified area, terrestrial systems require large number of base stations. On the other hand, GEO satellites have cell size limitations due to large footprints on the Earth's surface and non-geostationary satellites face handover problems and the need to deploy the entire constellation , thus requiring high launching costs to place them in orbits. In this case, HAPs seem to be an attractive choice.

HAPs can be used to serve different scenarios such as broadcast/multicast HDTV signal, high-speed wireless access, navigation and position location systems, intelligent transportation systems, surveillance, remote sensing, traffic and environmental

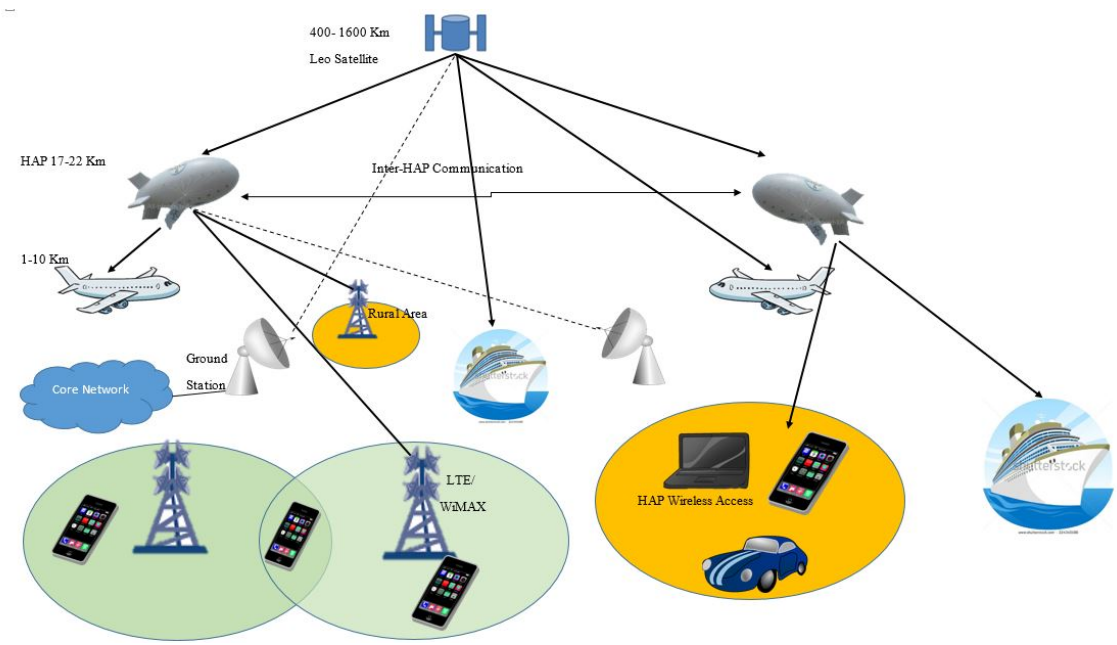



Figure 1.1: Integrated terrestrial/HAP/satellite networks

monitoring, emergency communications, disaster relief activities, and large-scale temporary events. In addition, one of the most promising capabilities of HAPs securing their role in fifth-generation (5G) networks is to provide high throughput backhaul links for ground-based pico and femto cells, thus minimizing the traffic burden in the mesh networks of the respective cellular systems. The ultimate long term aim is to obtain integration of terrestrial, HAPs and satellite networks. A generic framework for future generation communication networks based on integration of different communication infrastructures and employing an all IP-based core network is shown in Figure 1.1.

## 1.3 Types of HAPs

Basically, HAPs are classified into aerostatic and aerodynamic platforms [3], [4]. This is based on the underlying physical principle that provides the lifting force. Aero-

static platforms use buoyancy to float in the air, whereas aerodynamic platforms use propulsive forces created by jet engines.

Aerostatic platforms could be either balloons or airships. In order to provide buoyancy, they make use of a lifting gas in an envelope, most commonly hydrogen and helium. Balloons are usually unpowered platforms, and since the flight cannot be controlled easily they are usually manned. Airships, on the other hand, are normally unmanned powered platforms, capable of staying in the air for weeks and months. The main drawback of aerostatic platforms is their large size. A huge volume is needed to compensate for thin air in lower stratosphere. This causes dynamic drag during the course of a flight as well as difficulties for takeoff and landing. The large size of the platform has an advantage though, as it could accommodate larger and heavier payloads and its large area could be exploited for using solar cells to generate power.

Aerodynamic platforms, rely on aerodynamic lift for flying in the air. They cannot stay in the air unless they move forward, and therefore have to be above the coverage area to keep a quasi-stationary position. Due the density of the air at the operating altitudes of around 20 km, aerodynamic platforms require large size wings to obtain sufficient lift, making the radius of the circular flight of the platform in the range of a few kilometers. Due to the circular flight, the platform also requires some compensation for antenna pointing.

Neither of the two types of HAPs is yet available for delivering HAP-based communication services. Both are at different stages of development for operation in the stratosphere and are subject to different perceptions. They still need more modifications so that HAPs could move from research phase to operation phase in both the technological sense and in terms of flight regulations. It is expected that initial use

of HAP-based communication services can be for event servicing and disaster relief applications using existing manned and unmanned aircraft equipped with application specific modular payloads.

## 1.4 HAP Radio Regulations

HAP communications are subject to two main forms of regulations, radio-frequency (RF) regulations and aeronautical regulations. Since our thesis considers a telecommunications aspect in a HAP, we present the RF regulations only in this section. RF regulations are under the global control of the International Telecommunications Institute-Radiocommunications sector (ITU-R). ITU-R has allocated several frequency bands for HAPs to provide different broadband multimedia applications in millimeter-wave band and International Mobile Telecommunications (IMT)-2000 services in third generation (3G) frequency bands. ITU-R allocation specifies:

1. 300 MHz in each direction in the 47/48-GHz band worldwide on the secondary basis shared with satellites for all HAPs communications [5],

2. 300 MHz in each direction in the 31/28-GHz band also on the secondary basis in over 40 countries worldwide excluding all of Europe for fixed broadband services [6],

3. 2.1-GHz IMT-2000 band to be used for the provision of 3G services to users [7],

4. the band 5.85-7.075 GHz for HAP gateway links for fixed services [8].

## 1.5 Radio Resource Allocation and Admission Control for Multicasting

Just like any wireless system, HAP needs to manage its radio resources as efficiently as possible in order to gain the maximum desired benefit. This benefit could be the system data capacity, the system number of users, the fairness among the system's users etc. One of the aspects that *radio resource allocation* (RRA) has a direct impact on is the admission of users in the system. Simply, the availability of resources determines how many users can be admitted, or served in the system. The radio resources that need to be managed for a HAP having multiple antennas using *orthogonal frequency division multiple access* (OFDMA) are:

1. the radio power,
2. the frequency subchannels,
3. the time slots over the subchannels,
4. the antennas (antenna selection).

Choosing which users to admit into the system affects the total number admitted. This is because the users have different channel conditions due to their different positions and also due to the random nature of the radio channel. For example, if a user is in a location where the received signal quality is poor, and it is to be admitted into the system, it would need too much radio power to compensate for the channel attenuation. This could lead to little remaining power that is insufficient to admit other users. If that user would have not been admitted, the HAP might have been able to serve a larger number of users with good channel conditions. This is a simple

example considering power only. It grows much more complex when subchannels, time slots and antenna selections are to be allocated too.

Multicasting is the transmission of the same information to a group of users instead of transmitting the same information to each user individually (unicasting). This type of transmission saves a lot of radio resources as compared to unicasting, and is therefore, usually the method used to transmit same information to a group of users in any network. We can have more than one multicasting session in a HAP system and each user may want to join more than one session at the same time. Each multicast session transmits its data on the same set of subchannels, time slots and antennas with the same power level for all users in the multicast group. RRA is needed for *admission control* (AC) of multicast sessions so that efficient admission decisions are made for users wishing to join different multicasting groups.

Since aeronautically and mechanically reliable platforms are still in the development phase, the amount of published research for telecommunication services over HAPs, particularly RRA and AC, is limited compared to other wireless systems, let alone RRA and AC for multicasting in specific. Moreover, most of the big research projects for HAP like SHARP, Skynet, StratSat, HALO,CAPANINA, Helinet and HAPCOS [9–14] started their activities between 2000-2006, a time in which the most popular wireless interface in wireless telecommunications research was *code division multiple access* (CDMA) based *Universal Mobile Telecommunications System* (UMTS). Therefore, most of the published research in RRA and AC was for CDMA based HAPs. *Orthogonal Frequency Division Multiplexing* (OFDM) is one of the possible techniques to be used for transmission between the HAP and the users due to its well known capabilities in mitigating wireless channel impairments that result from high mobility and high transmission speeds [15]. Hence the multiple access scheme

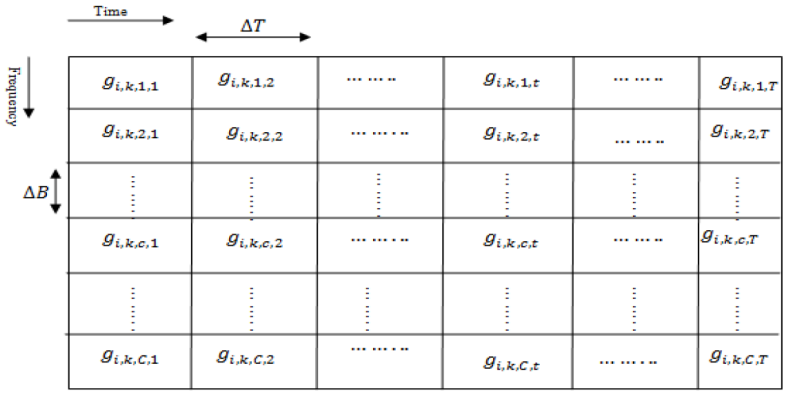



Figure 1.2: OFDMA frame.

that is expected to be used in HAPs is OFDMA. Therefore, we believe that more research in HAPs should be done considering this type of interface.

In OFDMA, the spectrum is divided into subchannels and time slots as shown in Figure 1.2. A user $k$ could have different gains for every subchannel $c$ and time slot $t$, therefore this gives many degrees of freedom as to which subchannel to allocate to a user in each cell (or HAP antenna) for a particular time slot and the power levels to allocate according to gain values $g_{i,k,c,t}$. Hence frequency diversity, time diversity, antenna diversity and multiuser diversity can be exploited.

## 1.6 Scope and Contribution of the Thesis

This thesis proposes a novel admission control and radio resource allocation problem to the HAP literature. Derivations for the mathematical formulations were done and suitable problem specific and structure oriented solution methodologies were used. The problem the thesis considers is joint AC-RRA for OFDMA based HAP system with multicasting, in the downlink, for admitting the users whose QoS requirements and their associated groups' requirements could be met. The QoS requirements con-

sidered in this thesis are the signal-to-interference-noise-ratio (SINR) of a multicast session for each user and the session's minimum and maximum data capacity constraints for all the multicast groups. First we consider maximizing the spectrum utilization by serving the largest number of users on all the available frequency-time slots. In an extended system model, we consider maximizing the number of highest priority users that are admitted to all multicast sessions in the HAP service area.

We consider two system models. A primary system model (P-SysMod), in which:

1. the concept of "cells" is adopted where each user falling within the foot print of antenna beam is associated with that antenna only. Hence a user can only receive from one antenna at most and any possible antenna beam overlaps are not taken into account.

2. A user can request, and hence can only receive sessions being transmitted in the cell in which the user resides,

3. all users assume the same level of priority,

4. the spectrum utilization, i.e. the number of users each frequency-time slot can serve, is the objective that is to be maximized.

An extended system model (E-SysMod) considers the following:

1. more flexibility by allowing transmission of a multicast session to the users in a group on more than one antenna at different frequency-time slots. Hence the users in the group can receive transmission on more than one antenna simultaneously.

2. a user can request, and hence receive, sessions being transmitted in any cell of the HAP service area,

3. all users are assumed to have heterogeneous priority levels for different multicast sessions,

4. the total number of admitted users with highest priorities in the HAP service area are maximized.

P-SysMod was the first part of our research work and is the topic of the first chapter of our research work in our thesis (Chapter 3), where the formulation, solution methodology and results were published in one journal and two conference papers. Since that system model is very rich and considers many different aspects that were not considered together, by other researchers in previous works for HAPs, we decide to go deeper in the same problem to see if we could achieve any improvement to what we have obtained for the problem. Since there could be many ways to formulate the same problem, we try to find a formulation that could be solved more efficiently than the one we obtained for P-SysMod. We were successful in obtaining a much smaller formulation which we believe is an important achievement as any algorithm's computational effort is always proportional to the formulated problem size. Since we are able to greatly reduce the problem size, we are able to extend the system model (to E-SysMod) while still having a far much smaller formulation than that we obtained and solved for in P-Sys Mod. Hence the aspect that we consider in comparing the two system models, P-SysMod and E-SysMod, is the formulation size for each which we illustrate in Chapter 4 to be in much efficient for E-SysMod. Therefore, the first part of our work is P-SysMod, its derived formulation and solution method. Then we improve the formulation and take the opportunity of the much efficient formulation to consider a little more complicated scenario in E-SysMod. We present this flow throughout the chapters 3-6.

Both system models involve too many degrees of freedom which make the task of

solving the problem complicated. We have not seen similar models in the HAP literature, probably due to their high complexity. However, by considering the primary and extended models, we decided to take a step in the direction of combining the following into one problem:

1. power allocation to multicast groups,
2. subchannel allocation,
3. time scheduling,
4. multiple antenna selection,
5. user to multicast group assignments,
6. heterogeneous user priorities and
7. reusing spectrum.

The work done by Zhao et al. in [16] is the closest to our work. They considered a unicasting unicellular system while we consider multicellular multiple-session multicasting system. Aside from power, subchannel and time allocation, which they have considered, we also consider multiple antenna assignments to multicast groups and frequency reuse across HAP cells. Finally, while the authors did not consider performing user admissions, we consider user to group admissions where each user can join more than one multicast group. Therefore, although the work done in [16] is the closest to ours, there are still many differences in the considerations and the system model. In Chapter 2, the last section provides an explanation of the how our work fits in the HAP RRA and AC and general multicasting literature.

For both system models an optimization problem formulation is developed. Although the extended system model is more complicated, we succeed in finding a

substantially more efficient formulation than that for the primary model. For the optimization problems, different operations research based optimization techniques are used. A branch and bound framework is used for the formulations obtained for both P-SysMod and E-SysMod. For P-SysMod, we use the Lagrangian relaxation and the subgradient algorithm to dualize different sets of constraints and solve easier problems to obtain bounds that can be used for pruning in a branch and bound scheme to solve the problem [17], [18]. The tighter the bounds of the solution, the larger the sections that can be pruned from a branch and bound tree, yielding quick good solutions with small memory requirements to store the tree leaves. The performance evaluation is obtained by computing the goodness of the different bounds in computational experiments.

For the extended model, E-SysMod, we use linear outer approximation by McCormick underestimators as a relaxation for the formulated mixed binary quadratically constrained program [19] and *mixed integer linear programming* techniques. Different branching schemes for the branch and bound scheme are used and their performances are evaluated by numerical experiments [20]. Also, a reformulation technique that linearizes a certain type of quadratic constraints in the formulation is used and experiments were conducted to evaluate the performance with and without the reformulation linearization scheme. Domain propagation methods, separating cuts and heuristics are also used for the formulation of E-SysMod [21]. The performance is evaluated for all of the methods and their different combinations using computational experiments. The parameters used for performance comparison are:

1. the duality gap,
2. the number of branch and bound nodes needed,
3. the number of iterations needed,

4. the average number of iterations per node,

5. the number of instances for which a feasible solution is found,

6. the time needed to find the first feasible solution,

7. the value of the objective function.

An important aspect in solving an optimization problem for AC-RRA over HAP is to find the best possible solution quickly before the radio channel changes its gains due to fast fading, which would lead to a change in the optimization problem's coefficients. Furthermore, due to the HAP's limited hardware resources on-board, it is desired to keep the memory usage as low as possible. The speed of obtaining the solution is measured by either the number of iterations, the time consumed or both. The memory usage is directly dependent on the size of the branch and bound tree. Therefore, the smaller the number of nodes, the smaller the memory needed. The quality of the solution is measured either using the bound goodness or the duality gap.

The Lagrangian relaxation removes troubling constraints from the formulation and adds their respective penalty in the objective function, creating a dual problem along with an easier primal problem to solve [22]. By easier we mean low computational effort. The dual problem is a convex problem with a linear piecewise objective function which uses a variant of the gradient techniques known as the subgradient techniques to find the optimal dual solution. Three different rules for finding the step size value which the algorithm moves in the subgradient directions are used for performance evaluation and comparison. Also, three different sets of constraints are dualized and structure specific algorithms for their respective simpler primal problems are used.

The branch and bound algorithm is a generic framework that is mostly used for

problems that involve binary or integer variables. It is a divide and conquer scheme that divides the problem into smaller simpler problems, known as nodes. The *root* node is the whole problem before division while the the rest of the nodes are smaller subproblems that have either been solved or still need to be solved. A *leaf* node is a subproblem that either has not yet been processed or one that has been processed and *pruned*. Any pruned node has no descendants. The *bounding* step avoids complete enumeration of potential solutions of the problem by pruning. A node gets pruned when it proves that none of its descendants can give a better primal feasible solution than the one found so far during the course of the algorithm.

The main purpose of separating cuts is to tighten the relaxation of a problem and hence produce sharper bounds that help in early pruning. The branching procedure is the selection of the variable to branch on to create the new descendant nodes. A branching scheme that yields best dual bounds is desirable as it leads to early pruning in the tree.

In this thesis, since the problem considered is quite different from the other problems in the HAP literature, and even the terrestrial wireless multicasting in OFDMA systems as Chapter 2 shows, we believe the comparison of our work with those would not provide us with sufficiently useful conclusions. Basically, the decisions to be made, the QoS constraints and even the objective function are all different. The comparisons that we perform however, are in the different formulations and the performance of different solution components like branching rules, subgradient step size formulas, the effect of introducing domain propagations separating cuts and heuristics etc. Comparison of different algorithm components options allows us to conclude the most efficient choices to use for the problem. For example, in Chapter 3, we conclude the most efficient step size formula and the most efficient selection of constraint sets to

dualize in terms of the bound goodness. We compare the two formulations in Chapter 4 and illustrate that the second one that we were able to derive later is more efficient and explain how. The second formulation, which also considered an extension to the first (primary) system model had a combination of algorithmic components proposed to solve it within a branch and bound framework. For those components, different choices and combinations are compared to find the most efficient choices in terms of solution goodness, speed(number of iterations) etc. At the end of chapters 3 and 6 we conclude with the most suitable choices to solve the two formulated problems. The main benefit of using branch and bound based solution methods is that if we decide to stop the algorithm before obtaining the optimum solution due to time limitations, we are left with the best solution found so far, which is a sub-optimal solution, and the information of how far that solution is from the optimum solution. Also, it is possible for the branch and bound algorithm to terminate once it obtains an acceptable sub-optimal solution that is within any desired % from the optimal. Hence we can say that the branch and bound based approach can be used to return sub-optimal solutions within an acceptable nearness to the optimal or, the best found solution in a maximum tolerable solution time.

## 1.7 Thesis Organization

Chapter 2 gives a review on previous related RRA and AC research in HAPs. Also, it provides a separate section for an overview for previous works done in multicasting RRA over OFDMA terrestrial cellular networks. In Chapter 3, the primary system model, P-SysMod, is presented and explained in details. An optimization problem is formulated, whose solution yields the best allocation of HAP resources such as radio power, sub-channels, and time slots. The problem also finds the best

possible frequency reuse across the cells that constitute the service area of the HAP. The objective is to maximize the spectrum utilization, of a given OFDMA frame, for admitting users to the multicast groups. A bounding subroutine in a branch and bound algorithm is obtained by decomposing it into two easier subproblems, due to its high complexity, and solving them iteratively. The first subproblem turns out to be a *binary linear program* (BLP) of no explicitly noticeable structure and therefore Lagrangian relaxation is used to dualize some constraints to get a structure that is easy to solve. Subproblem 2 turns out to be a linear program with a fractional knapsack problem structure. Hence a greedy algorithm is proposed to solve subproblem 2 to optimality. The subgradient method is used to solve for the dual variables in the dual problem to get the tightest bounds, for which different step size rules are used.

In Chapter 4, the extended system model E-SysMod is provided and a much more efficient formulation is derived. A comparison in the sizes of the two formulations show that the new formulation provided in Chapter 4 is much smaller. It is well known that when the number of variables, especially those of binary nature, and the number of constraints are significantly reduced, then the computational complexity is also greatly reduced (exponentially for binary variables). Therefore, efficient formulation has a large impact on the efficiency of the solution procedures. The problem formulation is a mixed binary quadratically constrained program and the constraints are proved to be nonconvex.

Chapter 5 shows the solution procedures used to solve the formulation for E-SysMod. An approach similar to [23] and [21] is used in which, an outer approximation is generated by linear underestimation of the non-convex quadratic constraints to relax the problem's feasible region. The problem becomes a *mixed binary linear program* (MBLP) and hence an LP solver can be used in a branch and cut algorithm.

The branch-and-bound (BnB) algorithm recursively splits the problem into smaller subproblems, thereby creating a branching tree. Different branching schemes are used for that purpose. At each node, domain propagation is performed to exclude further values from the variables' domains, and a relaxation may be solved to achieve an upper (dual) bound. The relaxation is then strengthened by adding further valid constraints, which cut off the optimum of the relaxation. Primal heuristics are integrated in the BnB procedure to improve the lower (primal) bound.

Finally, Chapter 6 explains the experiments performed on the formulation of E-SysMod in Chapter 4 and illustrates the numerical results obtained for the algorithmic procedures given in Chapter 5 to evaluate their performances. Three different experiment sets are provided in this chapter. The first experiment set compares the performance of activating-versus-deactivating a reformulation linearizion technique for a certain type of quadratic constraints, at the presolving phase. The second experiment set compares the performance of different branching techniques explained in Chapter 5. The third experiment set compares the performance of different combinations of domain propagation, cutting planes and heuristics. Chapter 7 gives the conclusions of the thesis and the possible future work.

# Chapter 2

# A Review on Related Work

In this chapter, the most relevant work done in the area of RRA and AC is presented. The chapter is divided into four main sections. Section 2.1 gives an overview on RRA and AC for CDMA based HAPs. Although we consider OFDMA in this thesis, however as pointed out in Chapter 1, most of the RRA and AC research work for HAPs considered CDMA as the wireless interface. In Section 2.2, RRA and AC for a few of the *Frequency Division Multiple Access* (FDMA)/OFDMA based HAPs are presented. Section 2.3 is dedicated to multicasting over terrestrial cellular networks. Although it does not consider HAPs, it is relevant to the work done in this thesis due to the nature of the transmission (multicasting) and the type of multiple access scheme. Finally, Section 2.4 explains the relevance of the work done in this thesis to the earlier work, as well as the concepts used and combined from the literature into the system models considered in this thesis.

## 2.1 RRA and AC in CDMA based HAPs

In [24] a *wide-band* CDMA (WCDMA) HAP for two types of traffic was considered. The first type had two service requirements, minimum transmission rate and bit-error-rate (BER). The second type was a best effort traffic (BET) type with BER as the only service requirement. The authors focused on the uplink since it posed a challenge, due to the constraints on each UT maximum transmit power. The aim was to find the optimum rate vector whose element values are the rate values that can be assigned to every UT depending on the QoS requirements of the radio connection. The schemes that were considered in [24] are:

1. *Optimal Resource Allocation* (ORA) strategy: which solves an optimization problem whose decision variables represent the transmission power and transmission rates for every user in order to maximize the throughput. This leads to a solution where BET UTs that only have the most favorable channel conditions transmitting to the HAP at the maximum allowable rate, while the others are forced to transmit at the lowest rate. Hence, the ORA scheme did not take into account the fairness criterion.

2. *Optimum Resource Allocation subject to Global Fairness* (ORA-GF): which achieves global fairness, by imposing equality constraints on the rate variables across all the HAP antennas for BET traffic.

3. *Optimal Resource Allocation with Local Fairness* (ORA-LF): maximizes the system throughput for uplink connections while preserving the fairness among UTs carrying the same traffic type only within each HAP cell, to achieve local fairness. To guarantee fairness among BET UTs at a cell level, equality constraints are imposed to force the rate decision variables for all BET traffic of all users

connected to a given HAP antenna to be equal.

4. *Simplified Resource Allocation with Local Fairness* (SRA-LF): is a heuristic scheme proposed by the authors to achieve a similar performance to the ORA-LF strategy but with lower complexity for real time implementations.

In [25], the unique characteristic of HAPs, that all base stations are collocated on the same platform was exploited. Unlike terrestrial cellular networks, this feature allows the exchange of information on the interference conditions within the cells between base stations with no signaling overheads. A reference scheme was used for comparison which processes the calls in parallel independently. If the total power at any BS is less than or equal the power outage threshold, the call gets accepted otherwise it gets blocked. The centralized proposed schemes of [25] are similar to the reference scheme except that they processes the calls centrally and sequentially instead of distributed and parallel. The central admission controller updated the BS total received power levels on call-by-call basis so that the admission decision for new calls can be made more accurately. Two centralized CAC schemes were given. The first one processed the calls in random order which could possibly admit a UT in a cell that already has high total received power leading to blockage of other subsequent calls in neighboring cells. The second one was based on priority, where highest priority for admission is given to a call request in the cell with the lowest total received power. The central admission controller ranked the calls accordingly. Both centralized schemes improved the *grade of service* (GoS) compared to the reference scheme. The GoS was best improved by the priority based scheme.

In [26] an extension for the work done in [25] was provided where a 'downlink' CAC scheme was proposed. A UMTS based HAP centrally manages the HAP power at the platform level and allocates power to the cells based on their demands. The

downlink and uplink CACs (proposed in [25]) work together to decide whether to admit a UT or not. Similar to the work done for the uplink in [25], a BS power based and a platform power based schemes were proposed. The basic idea for the BS based downlink CAC was to manage incoming calls according to the increase in the interference levels of the target cell as well as adjacent cells. Hence, with the admission of the new call, the forward link powers transmitted to all UTs must be increased to satisfy all UTs' signal-to-interference SIR requirements. A call was blocked if admitting the call would cause the UT's target base station as well as other neighboring base stations to exceed the maximum allowable output powers. The platform based scheme considers the total platform output power for the traffic channels as a single resource to be shared across the BSs based on demands rather than allocating fixed power for each BS. In this scheme, a call is admitted if the total platform power is not exceeded and the SIR thresholds are satisfied.

The authors of [25] and [26] extended their research work in [27] by considering a scenario of a UMTS system in which a HAP and a terrestrial cellular network are jointly deployed in a hierarchical system as illustrated in Figure 2.1. HAP was used to provide a macrocell coverage and the terrestrial towers are used for microcell coverage at a different frequency band, therefore, no cross-layer interference. A scheme was proposed that uses a combination of overflow and speed sensitive strategies to direct calls arriving within 'overlapped' service areas served by both HAP macrocell and terrestrial microcell layers to the appropriate layer. A speed threshold was used such that UTs with speeds greater than the threshold get directed to the HAP layer, while UTs with speeds lower than the threshold get directed to the microcell layer for admission. For UTs that request connection to the exclusive HAP macro cell layer, the CAC scheme proposed in [26] is used.

Since UTs in non-overlapped areas have higher blocking probability than the ones in overlapped areas, the CAC scheme in [27] was enhanced by centrally reserving a fraction of the platform power resource to accommodate new UTs arriving to the service area served exclusively by the HAP. The power reservation procedure was carried when the platform power utilization was near to its limit so that new UTs arriving to the exclusive HAP service area will have a smaller chance of being blocked. The power was reserved by handing down users in the overlapped areas from the HAP macrocell layer to the terrestrial tower-based microcell layer to free HAP resources. Two schemes of handing down users were proposed, one was random selection and the other gave priority to users with higher bit rate requirements. The proposed centralized resource reservation schemes achieved better GoS at the expense of higher handover rate. Also, it was found that by handing down mobiles in the order of decreasing bit rate, the total number of handing down executions was reduced leading to lower handover rates as compared to the case where users were handed down randomly.

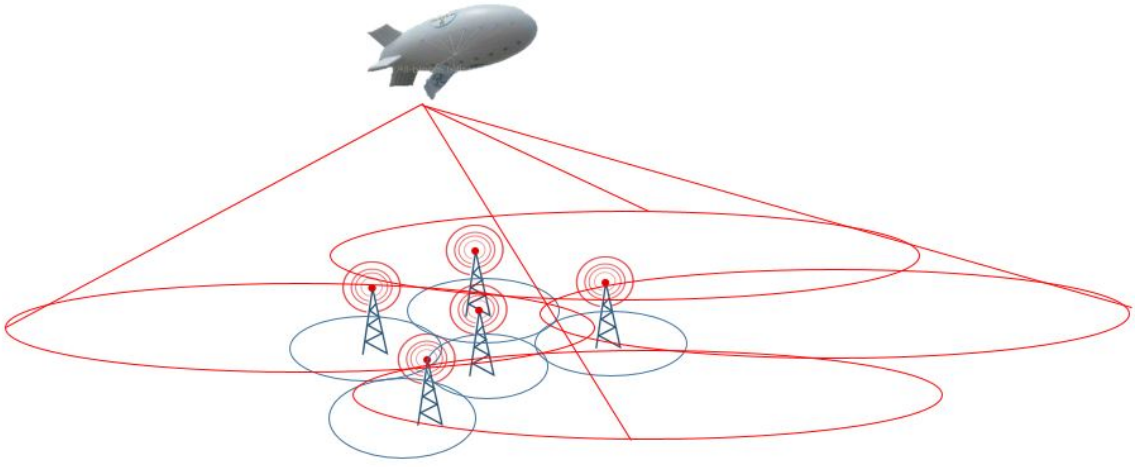


Figure 2.1: A hierarchical HAP and terrestrial cellular system

The work done in [25], [26], [27] were further extended in [28] by the same authors who proposed a speed and direction based CAC scheme for a UMTS based standalone HAP system. For this scheme, the system continuously tracked the rate of change of the difference between the energy per chip to interference power density ratio $E_c/I_o$ received from the mobile's serving base station's Common Pilot Channel (CPICH) and the next strongest $E_c/I_o$ received from the mobile's neighboring base stations' CPICH signals. It was used to derive the speed and direction of the mobile relative to the rest of the mobiles. Based on this information, the new call admission thresholds in each cell were dynamically adjusted to ensure that all cells reserved sufficient resources for handoff users that were predicted to enter their service areas from other neighboring cells. The objective was to reduce the handoff call dropping probability as much as possible since forced termination is more undesired than new call blocking.

In [29], an integrated system of HAP and terrestrial cellular UMTS like the one shown in Figure 2.1 was considered for *Multicast Broadcast Multimedia Services* (MBMS) applications. The authors' aim was to efficiently allocate transmission resources to multicast traffic streams by suitably selecting terrestrial and/or HAP channels while still preserving the desired GoS of unicast traffic. The authors proposed two different RRA approaches, one took into account the "number" of users in each multicast group to drive the RRA policies; while the second one based the RRA behavior on the "distribution of" users belonging to each multicast group. The "Number of Multicast Users" policy moved the multicast connections which belonged to the largest multicast group, onto the HAP channel since doing the opposite is expected to degrade the performance due to congestion in the terrestrial network. "Distribution of Multicast Users" achieved an improvement in performance in certain situations by assigning the *forward access channel* FACH of the HAP to the multicast group that

is most scattered across the cells. This freed more FACH of the terrestrial system for other multicast groups. The performance improvement achieved was both in terms of cell resource utilization and GoS of unicast connections.

## 2.2 RRA and AC for FDMA/OFDMA based HAPs

In [30], a heterogeneous network with two HAPs was considered where UTs with a limited HAP choice (labeled Group L) and users with a full HAP choice (labeled Group F) coexist in the same system as Figure 2.2 shows. Group F users have access to both HAP1 and HAP2 by smart or steerable antennas, while Group L users only have access to one of the HAPs due to some physical constraints such as fixed antennas. The multiple access scheme considered was FDMA. The authors modeled the channel allocation process as a birth-death two dimensional *Continuous Time Markov Chain* (CTMC). In order to improve the potentially inferior GoS of Group L due to relatively poorer HAP availability, the authors imposed restriction to Group F to deliberately limit the availability of channels for the group. The restriction mechanism used was for equalizing the blocking probability of Group F and Group L. Simply stated, the restriction blocks some Group F users to reserve more channels for Group L users which had more limited HAP availability. Using this compensation effect, a balanced blocking probability was achieved.

A restriction function of the number of occupied channels prior to the user's arrival on the chosen HAP was used, assuming there is spare capacity on a HAP when a Group F user arrives at one HAP. The restriction function is defined in terms of the probability of access to a channel when in a particular state of the Markov chain. If a group F user gets denied from accessing the chosen HAP as a result of the restriction mechanism, it does not get considered for access to the other HAP. If a group F user

got inhibited from accessing the chosen HAP because all the channels are occupied, the new Group F user would access the other HAP subject to the restriction function based on the number of occupied channels on the other HAP unless they are also fully occupied. Different ways were used to restrict Group F users. A constant restriction function was used, that applies equal probability of restriction to Group F users independent of the channel occupancy in the system. The authors also tried linear, quadratic and step restriction functions that placed higher emphasis on restriction when the channel occupancy was at higher levels. It was shown in Figure 2.2 that the step function was the most suitable to provide a balanced low blocking probability performance to both user groups simultaneously.

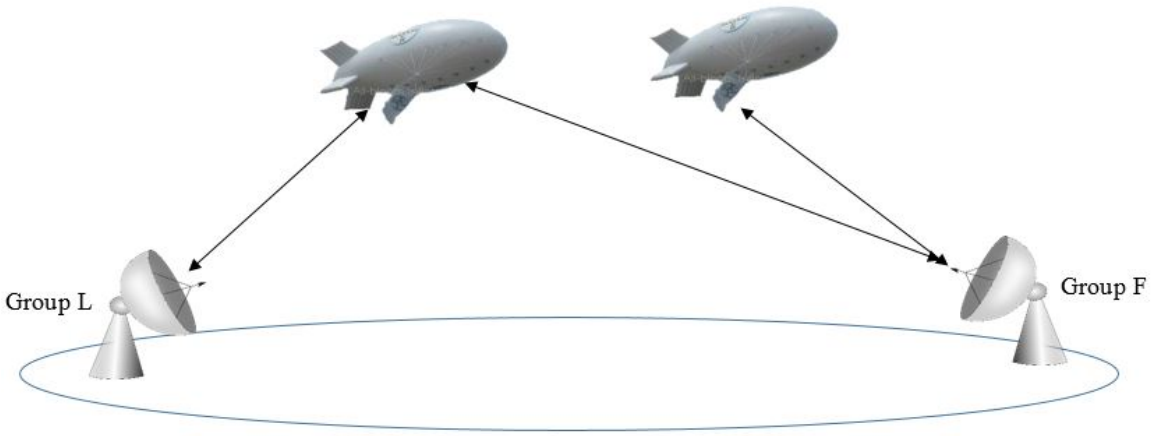



Figure 2.2: A system of two HAPs and two groups of users with different access choices

In [16], low-medium-altitude platforms (LMAPs) are considered, which are temporary platforms at altitudes lower than HAPs (around 2 Km) that provide telecommunications services for territorial users. The authors suggest the use of WiMAX based air interface and carry out joint radio resource allocation for time, frequency and power simultaneously in the downlink. The problem was modeled as a coopera-

tive game to make a good trade-off between throughput and fairness. The objective of the formulated problem was to achieve the Nash Bargaining Solution (NBS) of the cooperative game. Due to the high complexity of the problem, it was decomposed into two optimization subproblems, one for the time-frequency block allocation and the other for power allocation, hence reducing the complexity. Lagrangian multiplier method and Karush-Kuhn-Tucker (KKT) conditions were used in both subproblems to obtain the solution. Similar work was done by the authors in [31] and [32] for *time division duplex-Long Term Evolution* (TDD-LTE) based LMAP for both the uplink and downlink to maximize the ratio of the attainable data rate to the maximum required data rate and power-efficiency respectively.

## 2.3 Related Work on RRA for Multicasting in OFDM Systems

While lots of research exist on RRA in unicast terrestrial multiuser OFDM systems, research on multicast RRA is still emerging in broadband wireless systems in general. There are two types of multicast transmissions, *single-rate* transmissions and *multi-rate* transmissions. In single-rate, the transmission to all users in each multicast group is with the same rate irrespective of their non-uniform achievable capacities. In multi-rate however, the transmission to each user in each multicast group can be at different rates based on what each user can handle.

There are three simple schemes that have been adopted for single-rate transmissions [33]:

1. *Predefined fixed default group rates*: CDMA 2000 1x EV-DO networks for example, use 204.8 kbps for multicast transmission and assign resources equally to all users in cyclic round-robin [34].

2. *Least Channel Gain (LCG) User Rate*: This scheme adaptively sets the group transmission rate according to the user with the poorest channel conditions [35], [36], [37].

3. *Average (AVG) Group Throughput*: The transmission rate to each group is based on the long-term moving average throughput of the group. There are various techniques for the group averaging, for e.g. in [38] the median throughput is selected among the achievable instantaneous throughput of the users in the group such that half of the members can be supported. Also, in [39] the single transmission rate is selected based on the exponential moving average received throughput of each user in the group.

For multi-rate multicast transmissions two techniques currently exist in the literature:

1. *Information Decomposition Techniques* (IDT): where high rate multimedia content is split into multiple sub-streams where users can subscribe to the amount of data that can reliably be received. It exploits user multichannel diversity and the perceived quality improves as the user receives more substreams. There are two types of IDT available in the literature, *Multiple Description Coding* (MDC) [40–42] and *Hierarchical Layering* (HL) [43, 44]. In both schemes, multicast data is split to multiple substreams and each subcarrier is allocated to one substream for transmission to the multicast group. A base substream is transmitted to all users. Users with higher channel gain have potential for more throughput and can therefore also receive additional enhancement substreams to improve quality of the base substream. The difference between MDC and HL is in the order of reception of the substreams. In HL, for any enhancement substreams to be successfully decoded, all the lower enhancement and base substreams must be successfully received. In MDC however, all substreams have

equal priority and any combinations of the received substreams can be decoded independently.

2. *Multicast Subgroup Formation* (MSF): This involves splitting multicast groups into smaller subgroups based on the intra-group users' channel qualities and single multicast transmission rate is defined for each subgroup. It combines the simplicity of single rate and higher capacity of IDT. The scheme however, lacks precise coexistence definitions to guarantee reliable transmission to LCGs. Some coexistence ideas that have been proposed for use in MSF are LCG users trade-off [45], subgroup resource sharing [46,47] and cooperative data relay [48].

Single rate multicast transmission is popular for its implementation simplicity and low complexity. This scheme guarantees reliable multicasting to user terminals with poor link conditions. However it puts severe restriction on the achievable throughput. Multi-rate, on the other hand, has been receiving more attention lately because of necessity to achieve user throughput differentiation such that improved system spectral efficiency is attained. Multi-rate multicasting addresses the sub-optimality that exists in single-rate transmission considering the intrinsic heterogeneous channel characteristics of each user terminal in a multicast group. Its aim is to alleviate the problem of intra-group unfairness in single-rate transmissions caused by large differences in channel conditions for different users in a multicast group. However it exhibits higher coding and synchronization complexities, requires high computational power and requires heavy retransmission overhead.

There are three main categories for multicasting RRA based on the desired objectives, those are:

1. *Strict Throughput Maximization* (STM): which often has been considered for multiple multicast resource allocation problems to manage inter-group coexis-

tence to obtain an optimum system spectral efficiency. It selects the transmission rate of a multicast group to be that of the group's user with best channel conditions. Hence it achieves poor fairness among the groups' users. Therefore, we can say that it achieves significant throughput for inter-group RRA at the expense of intra-group performance, especially when there are a lot of users in a group experiencing poor channel conditions. Some of the work in which STM has been considered is [49] and [50].

2. *Max-Min Fairness* (MMF): which tries to overcome the intra-group poor fairness by giving priority to users with poorest channel conditions to achieve their maximum possible rates. MMF based RRA algorithms maximize a threshold-rate, that is dependent on an intra-group single-rate scheme, iteratively until a Pareto optimal solution is reached. Pareto optimality is a state at which there is no other way to improve the throughput of the system without decreasing throughput of other users. In [45] and [51], the threshold is defined by the AVG of the group.

3. *Proportional Throughput with Fairness* (PTF): which attempts to balance a multicast group aggregate throughput while reducing resource starvation and providing fair QoS to all groups. This objective was considered in [52], [51] and [43].

For each category, various algorithms and optimization techniques have been proposed, many of have been summarized in [33].

## 2.4 Relation of the Work Done in this Thesis with the Previous Works

Figure 2.3 illustrates the categorization map of the different previous works done for RRA and AC that are both relevant to the communications system we are considering, which is HAP, and the type of transmission that we are considering, which is multicast transmissions. There are three different colors used under RRA and AC in the figure. The blue color represents the research works explained in Section 2.1, the green represents those covered in Section 2.2 and the red represents the research works covered by Section 2.3. Also, in the figure we can see some salient shapes, which indicate the key areas that we combine together in this thesis. We can see that there was no multicasting considered for OFDMA based HAP in the literature as indicated in its corresponding shape in the green section. Recognizing this main area of research to be missing, to the best of our knowledge, in HAP RRA and AC motivated us to consider it as our research problem.

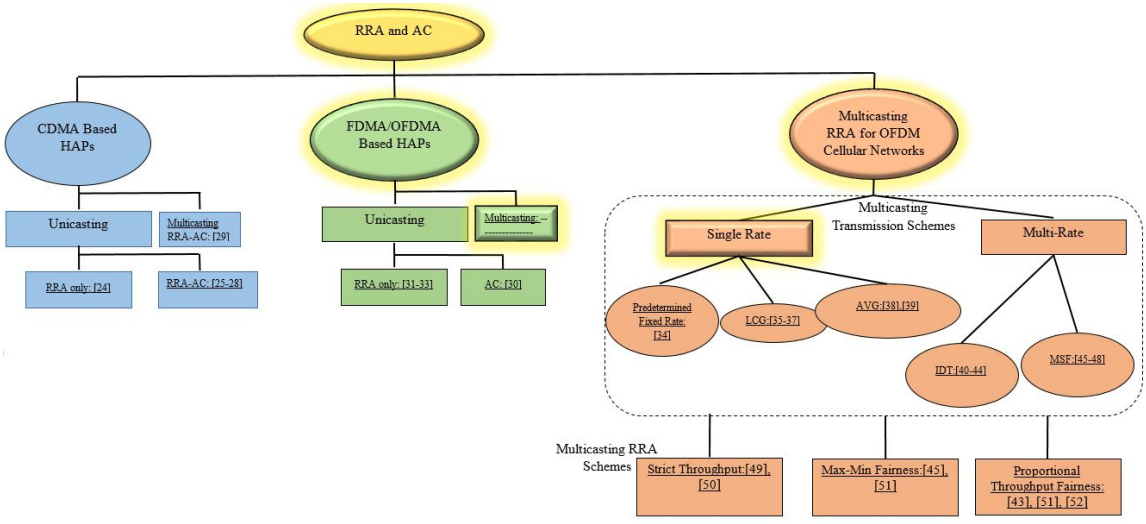



Figure 2.3: Illustration of the classification of RRA and AC in previous related work.

We combine three concepts from the literature:

1. central power management at the HAP level instead of at the BS level from [25],
2. frequency, time and power resource allocation for OFDMA from [16] and
3. single rate LCG multicast transmissions [33].

These ideas were there individually in different works in the literature, and we combine them for our system models P-SysMod and E-SysMod in this thesis. Moreover, each of those concepts are considered in a different context in our thesis. Speaking more specifically, the following are different contexts for each of the individual concepts used in this thesis:

1. We use the central power management at HAP level from [25] in OFDMA based HAP instead of CDMA based. We apply the concept not only to power, but to frequency and time slots.
2. Frequency, time and power resource allocation for OFDMA in [16] were considered for single cell unicasting, but we consider the idea for multicellular multicasting in this thesis.
3. Single rate LCG in the literature was obtained for the user with the least channel gain (as the technique's name suggests) in the multicast group. In this thesis, a similar but slightly different concept is used. Instead of using the lowest channel gain among all users in a group, we use the lowest SINR instead as it is more accurate in describing the users' quality of received signals. A user with good channel gain but too high interference would have a poor quality received signal. Moreover, in E-SysMod we define the concept of a group's highest data rate and use the highest SINR for all users in a group for a given frequency-time

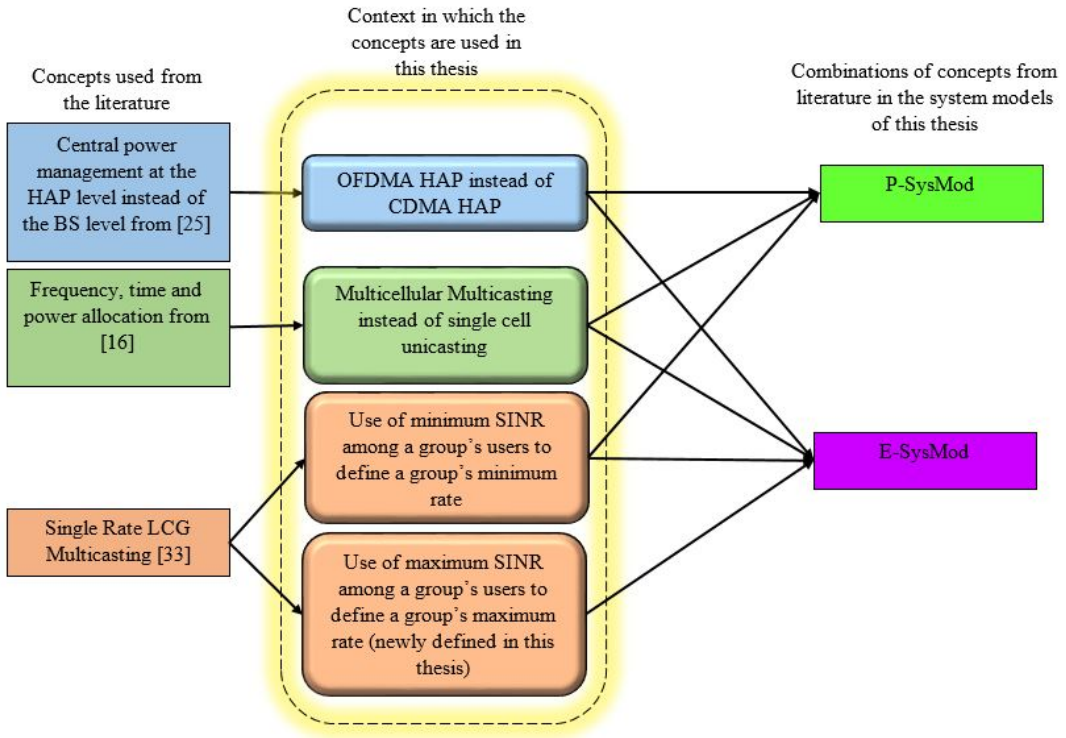



Figure 2.4: Illustration of the concepts combined from the literature and the context in which they are used for the system models in this thesis.

slot to calculate it. It is well known that variable bit-rate multicast streams, such as video, have a minimum and maximum range in which they vary. We do not want to allocate resources that yield a capacity higher than the maximum possible bit-rate of the multicast session. These resources would better be used for other groups with larger number of users with poor SINR conditions.

Figure 2.4 illustrates a summary for the combinations of concepts from the literature used in the system models in this thesis. The colors used in the figure for the different concepts used from the literature are the same as the colors of their respective categories in Figure 2.3. We choose single rate transmissions for its low complexity advantages, especially that our resource allocation problem is a complex one involv-

ing allocation of many different types of resources and making admission decisions in five dimensions. Hence the low complexity feature in single rate transmission is desirable. Moreover, in both P-SysMod and E-SysMod we are trying to maximize the spectrum utilization and the number of highest priority users admitted respectively, rather than maximizing the throughput or fairness. Hence, the drawback of throughput restriction in the single rate transmissions mentioned earlier is not much of a concern for us in both our system models P-SysMod and E-SysMod.

Furthermore, existing studies in radio resource allocation for multicast services in OFDMA systems have mainly investigated a single multicast session. Those studies, along with the few that considered multiple multicast sessions, mostly considered a single cell system [33]. As an effort to fill in this gap, we combine multiple muticast sessions with a multicellular HAP system in both P-SysMod and E-SysMod.

# Chapter 3

# Multicasting in a Single HAP System: Primary System Model, Formulation and Solution Bounds

In this chapter, we provide our primary system model, P-SysMod, for multicasting joint AC-RRA in an OFDMA based single HAP system such that the spectrum utilization is maximized. We define the spectrum utilization as the amount of users a single frequency-time slot conveys multicasting data to. The reason we consider that to be our objective function is simply because for a telecommunications service provider, the larger the number of users it can serve in a given bandwidth, the larger the revenue it can obtain. Assuming each unit of bandwidth could give a certain amount of revenue, then in multicasting, the revenue is directly proportional to the number of users served on all subchannels over all the slots of the OFDMA frame. The work in this chapter was published in [53], [54] and [55].

We exploit an important feature that HAPs have over terrestrial wireless communications networks, which is central power management at the platform level rather

than at the base station (HAP antenna) level [26]. This yields a greater flexibility in power allocation. Furthermore, we also use central management for all the frequency-time slots of the OFDMA frame at the HAP level and dynamically reuse them across the cells that constitute the HAP service area provided an SINR threshold is satisfied.

The rest of this chapter is organized as follows. Section 3.1, explains in details the primary system model P-SysMod. In Section 3.2, an optimization problem is formulated with the objective of maximizing the total number of users receiving transmission in all the frequency-time slots of the OFDMA frame in all the cells in the system. The QoS requirements are the BER, which correspond to SINR, and the data capacity rates of all the multicast sessions. In Section 3.3 a scheme for obtaining solution bounds based on decomposition, Lagrangian Relaxation [22], [17] and *subgradient* algorithm [18] is presented. The section is divided into three subsections, two subsections correspond to the two subproblems resulting from decomposition and the last subsection is dedicated for the dual problem. Section 3.4 explains the computational experiments conducted and presents the results for bound goodness of different dualized sets of constraints, different stepsize rules and the convergence of the solution bounding scheme. Finally, Section 3.5 concludes the chapter.

## 3.1 Primary System Model

We consider the scenario of a single OFDMA based HAP providing cellular like coverage to a number of users in a given geographical area. Unlike terrestrial cellular systems, all the cells in the service area have one common platform power source with a total power $P_{PF}^{Total}$ which allows for better and flexible power utilization. Also, besides power, the HAP centrally manages the other radio resources which are frequency subchannels and the time slots on those subchannels across all the cells of its service

area. Figure 3.1 illustrates the system model considered for AC-RRA multicasting across all the users in the HAP's service area. The radio power for each antenna is allocated at different frequency-time slots of the OFDMA frame. The users that request to join certain multicast sessions are assigned to the same set of frequency-time slots, inside their respective cells. The frequency-time slots are reused across the cells such that the SINR requirements of all users are satisfied. Please note that for the rest of this chapter, the terms 'antenna' and 'cell' are used interchangeably since a cell is defined by the corresponding HAP antenna footprint, which is why we use the same colors for the cells and their respective antennas in Figure 3.1. A multicast group for P-SysMod represents the users receiving the same multicast session $m$ and are located in the same cell $i$.

The OFDMA frame consists of a fixed number of subchannels $C$ and an equal number of time slots $T$ on each subchannel. Therefore, a frequency-time slot refers to one subchannel $c$ in time slot $t$, i.e. a slot $(c,t)$, with $\Delta B$ as its bandwidth and $\Delta T$ its time duration. The gains $g_{i,k,c,t}$ for each slot $(c,t)$ for a user $k$ in cell $i$ are independent and we assume that their values are known at the HAP side (see Figure 1.2). The channel gains depend upon the instantaneous values of large scale fading and small scale fading. In a HAP system, large scale fading is a result of free space path loss and attenuation due to rain and clouds [56]. Small scale fading is acceptably modeled as Ricean fading due to the presence of line of sight rays from the HAP to most of the locations in the HAP service area [1]. The channel gain $g_{i,k,c,t}$ between base station (antenna) $i$ and user $k$ on the frequency-time slot $(c,t)$ can hence be given as:

$$g_{i,k,c,t} = \left(\frac{\check{C}_{light}}{4\pi d_k f_c}\right)^2 .G_H\left(\varpi_{i,k}\right).G_k^u.\frac{1}{A\left(d_k\right)}.\varphi_{k,c,t} \tag{3.1}$$

where

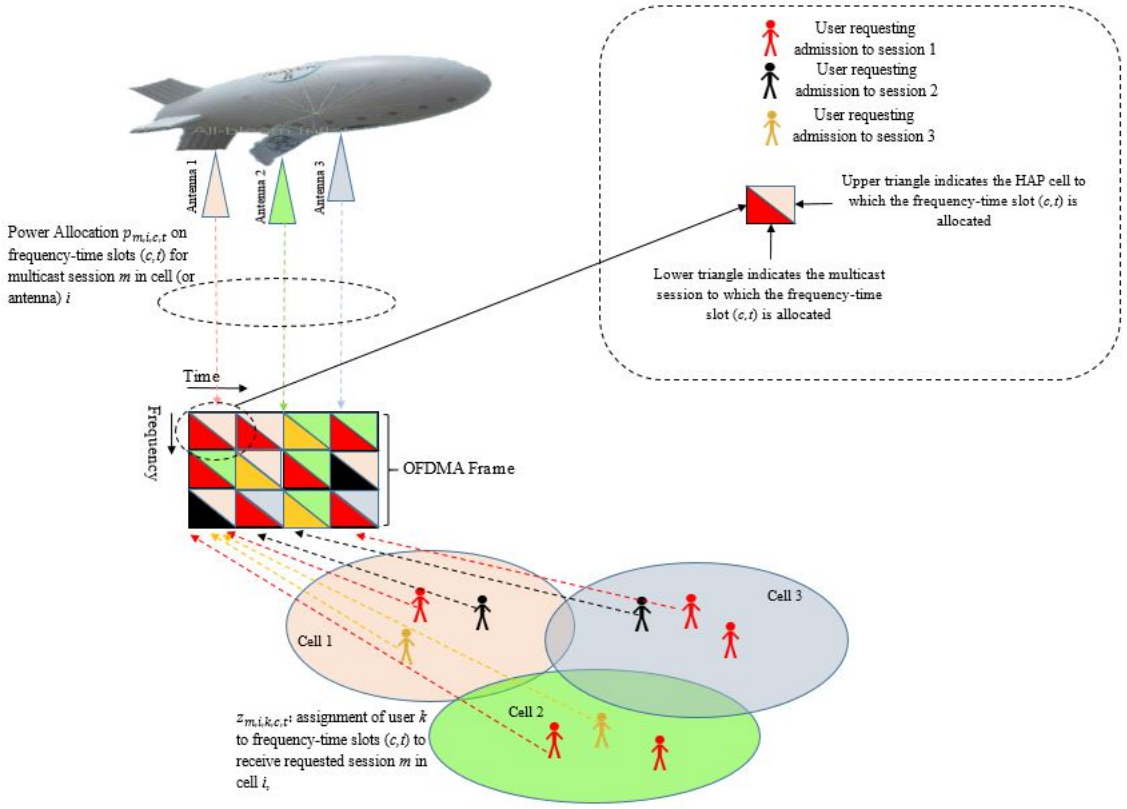



Figure 3.1: Primary System Model (P-SysMod)

- $G_H\left(\varpi_{i,k}\right)$ is the gain seen at an angle $\varpi_{i,k}$ between user terminal $k$ and antenna $i$ boresight axis.

- $d_k$ is the distance between the HAP and user terminal $k$, $\check{C}_{light}$ is the speed of light and $f_c$ is the carrier frequency.

- $A\left(d_k\right)$ is the attenuation due to clouds and rain. This depends on the distance between the HAP and each user $k$ in the service area.

- $G_k^u$ the antenna's gain of user terminal $k$.

- $\varphi_{k,c,t}$ is the Ricean small scale gain in frequency-time slot $(c,t)$ for user terminal $k$.

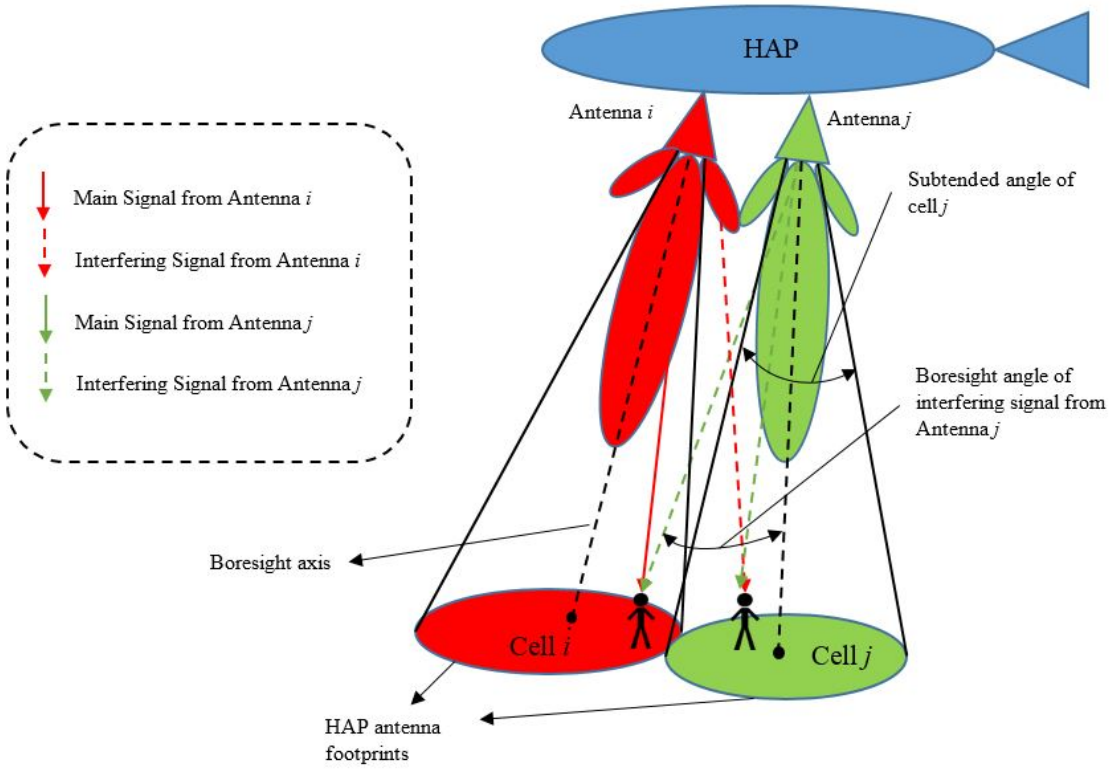



Figure 3.2: Interference in a single HAP system

The beams of the antennas co-located on the HAP interfere with each other, as illustrated in Figure 3.2 for a single HAP system. The interference to a user in a particular cell is due to the reception of unwanted transmissions at boresight angles greater than angles that subtend the neighbor cell footprints through the mainlobes and side lobes of their antennas [57]. A frequency-time slot is reused across different cells for any multicast sessions as long as an SINR threshold $\gamma^{th}$ for the users satisfy an acceptable BER in that slot. However within any cell $i$, a frequency-time slot cannot be assigned to more than one multicast session $m$.

Let the set of users that get admitted to multicast session $m$ in cell $i$ for a given OFDMA frame be $N_{m,i}$. The transmission rate $r_{m,i,c,t}$ in a frequency-time slot $(c,t)$ in cell $i$ for session $m$ has to be supported by the users with the worst SINR among

all the users $k \in N_{m,i}$ to ensure they can all reliably receive the multicast session. We define $r_{m,i,c,t}$ as Shannon's capacity on slot $(c,t)$ for the user with the poorest link conditions in $N_{m,i}$:

$$r_{m,i,c,t} = \frac{\Delta B \Delta T}{F} \log_2 \left(1 + \min_{k \in N_{m,i}} \gamma^{c,t}_{k,N_{m,i}}\right), \tag{3.2}$$

where $F$ is the OFDMA frame duration and $\gamma^{c,t}_{k,N_{m,i}}$ is the SINR of user $k$ on slot $(c,t)$ and is defined as

$$\gamma^{c,t}_{k,N_{m,i}} = \frac{g_{i,k,c,t} p_{m,i,c,t}}{\sum_{m=1}^{M} \sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}, \tag{3.3}$$

where $\mathscr{S}$ is the set of cell indices, $p_{m,i,c,t}$ is the power allocated for session $m$ in cell $i$ over slot $(c,t)$ and $\sigma^2$ is the noise power per subchannel.

In each cell $i$ in the HAP service area, there is a set of users that are tuned to receive (or request to receive) a multicast session $m$, let this set be $s_{m,i}$. Our interest is to find the following such that the number of users served by all frequency-time slots (i.e. spectrum utilization) is maximized:

- Which users from $s_{m,i}$ to assign to the set $N_{m,i} \subseteq s_{m,i}$.
- Which frequency-time slots to assign to those users in $N_{m,i}$, for all sessions in all cells.
- The power level for each multicast session $m$ in each cell $i$ for each frequency-time slot $(c,t)$.

The QoS requirements which must be satisfied are:

- Minimum SINR requirement for each user $k$ receiving a multicast transmission $m$ in cell $i$ over a slot $(c,t)$ i.e.$\gamma^{c,t}_{k,N_{m,i}} \geq \gamma^{th}$.

- Minimum and maximum capacity requirements ($R_m^{min}$and $R_m^{max}$) for multicast session $m$ in cell $i$ given by

$$R_m^{min} \leq \sum_{c=1}^{C}\sum_{t=1}^{T} \frac{\Delta B \Delta T}{F} \log_2\left(1 + \min_{k \in N_{m,i}} \gamma_{k,N_{m,i}}^{c,t}\right) \leq R_m^{max}.$$

The optimization problem is therefore a joint AC-RRA scheme to maximize the spectrum efficiency, which we define to be number of users receiving multicast transmissions in all frequency-time slots of an OFDMA frame in the entire HAP service area.

## 3.2 Optimization Problem Formulation for P-SysMod

In this section, we present our optimization problem for the RRA. Table 3.1 gives the definitions of the different variables, constants and notations used for the formulation. The optimization problem is given as:

The optimization problem can be given as :

$$\max_{z_{m,i,k,c,t}, p_{m,i,c,t}, x_{m,i,c,t}} \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{t=1}^{T} z_{m,i,k,c,t} \qquad (\mathcal{HAP}^{Init})$$

s.t.

$$C1 : z_{m,i,k,c,t} = \begin{cases} \{0,1\} & if\ \lambda_{m,i,k} = 1, \\ 0 & \text{otherwise,} \end{cases} \quad \forall m, i, k, c, t$$

Table 3.1: Notation Definitions for the Primary System Model

| **Notation** | **Definition** |
|---|---|
| $\mathscr{S}=\{1,2,...,S\}$ | set of cells in the HAP service area. |
| $\mathscr{M}=\{1,2,...,M\}$ | set of sessions being multicasted in the HAP service area. |
| $\mathscr{C}=\{1,2,...,C\}$ | set of subchannels available for the HAP service area. |
| $\mathscr{T}=\{1,2,...,T\}$ | set of time slot indices on a given subchannel. |
| $\mathscr{K}=\{1,2,...,K\}$ | set of user indices in HAP service area. |
| $s_{m,i}\subseteq\mathscr{K}$ | set of users that request session $m$ in cell $i$ . |
| $\lambda_{m,i,k}$ | a binary constant indicating, if user $k$ resides in cell $i$, and whether it requests to receive a session $m$ being transmitted in cell $i$. |
| $z_{m,i,k,c,t}$ | is a binary variable indicating whether user $k$ receives a transmission for session $m$ in cell $i$ in a given OFDMA frame slot $(c,t)$. |
| $p_{m,i,c,t}$ | is a non-negative variable indicating the power level in slot $(c,t)$ of cell $i$ for session $m$. |
| $x_{m,i,c,t}$ | is a binary variable that indicates whether slot $(c,t)$ is assigned to session $m$ in cell $i$. |

$$C2: z_{m,i,k,c,t,} = z_{m,i,k,c',t'}, \qquad \forall c,t: x_{m,i,c,t}=1,$$
$$\forall c',t': x_{m,i,c',t'}=1, \forall m,i,k$$

$$C3: \gamma^{c,t}_{k,N_{m,i}} \geq \gamma^{th} \quad \forall m,i,k,c,t: z_{m,i,k,c,t} \neq 0,$$

$$C4: R^{min}_m \leq \sum_{c=1}^{C}\sum_{t=1}^{T} r_{m,i,c,t} \leq R^{max}_m \quad \forall k: z_{m,i,k,c,t}\neq 0, \forall m,i$$

$$C5: x_{m,i,c,t} = \begin{cases} 1 & p_{m,i,c,t}>0, \\ 0 & \text{otherwise}, \end{cases} \quad \forall m,i,c,t$$

$$C6: \sum_{m=1}^{M} x_{m,i,c,t} \leq 1, \quad \forall i,c,t$$

$$C7: \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{c=1}^{C} p_{m,i,c,t} \leq P^{Total}_{PF}, \quad \forall t$$

$$C8: p_{m,i,c,t} \geq 0 \quad \forall m,i,c,t$$

The following is a brief explanation of the objective function and the constraints:

- We construct a single aggregate objective function by summing the users on all frequency-time slots receiving the same sessions across all cells. This reflects the spectrum utilization, which increases proportionally with the number of users that get accepted to receive transmissions of the corresponding requested sessions in the available spectrum of the OFDMA frame.

- Constraint set $C1$ suggests that a user terminal $k$ may be assigned to receive a multicast transmission for session $m$ in cell $i$ in an OFDMA frame slot $(c, t)$ only if:

  - user $k$ is in cell $i$,
  - session $m$ is being transmitted in cell $i$
  - a request has been received from user $k$ to join the multicasting group of session $m$ in cell $i$.

- Constraint set $C2$ ensures that user terminals assigned to receive session $m$ in cell $i$ over a set of slots in the OFDMA frame should be the same in each of those assigned slots, since all users receiving a session share the same resources.

- In constraint $C3$ the selection of user $k$ residing in cell $i$ to receive a session $m$ in a particular slot $(c, t)$ (i.e. the value of $z_{m,i,k,c,t}$ ) depends on whether there is enough power for that user to join the multicast group $m$ and receive transmission in slot $(c, t)$ of cell $i$ while satisfying the minimum SINR value. This is important since different users have different channel gains on each slot $(c, t)$ .

- In constraint set $C4$ the value of $z_{m,i,k,c,t}$ depends on whether there is enough power for that user to satisfy the minimum and maximum rate constraints.

- Constraint set $C5$ suggests that a binary assignment variable $x_{m,i,c,t}$ for frequency-time slot $(c,t)$ to be 1 if any power $p_{m,i,c,t}$ is allocated for session $m$ in cell $i$ on the frequency-time slot $(c,t)$. If no power is allocated for session $m$ in cell $i$ on frequency-time slot $(c,t)$, then this means that the slot is not to be assigned to session $m$ in cell $i$, which would force the corresponding assignment variable $x_{m,i,c,t}$ to zero.

- Constraint set $C6$ implies that the power $p_{m,i,c,t}$ allocated for a session $m$ in cell $i$ on a frequency-time slot $(c,t)$ can only be non zero for at most one session in a particular slot $(c,t)$ for a cell $i$.

- Constraint set $C7$ guarantees that for a given time slot, the total power of the HAP is not exceeded and $C8$ guarantees that the power values are always non-negative.

Using some algebraic and logical manipulation, we were able to transform the problem to an easier formulation with the variables $z_{m,i,k,c,t}$ and $p_{m,i,c,t}$ only. The reformulation is obtained by: 1) replacing the variable $x_{m,i,c,t}$ in $\mathcal{HAP}^{Init}$ and its corresponding constraint set $C6$ with a set of linear constraints in $z_{m,i,k,c,t}$ and $p_{m,i,c,t}$ 2) getting around the difficulty with the capacity constraint $C4$ by replacing it with non-logarithmic constraint sets in $z_{m,i,k,c,t}$ and $p_{m,i,c,t}$. Either the lower capacity bound constraint or the SINR constraint can be selected and the other can be ignored due to its redundancy. The following constraints in $\mathcal{HAP}^{Init}$ can be rewritten as follows:

- $C1$ can be written as:

$$0 \leq z_{m,i,k,c,t} \leq \lambda_{m,i,k}, \quad \forall m,i,k,c,t$$

- $C2$ can be written as:

$$z_{m,i,k,c,t} + z_{m,i,k',c',t'} \leq 1 + z_{m,i,k,c',t'}, \ \ \forall m,i; \ \forall k,k^{'} : k \neq k^{'}; \forall\,(c,t) \neq \left(c^{'},t^{'}\right)$$

- $C5$ and $C6$ can be replaced by:

$$z_{m^{'},i,k^{'},c,t} \leq 1 - z_{m,i,k,c,t}, \quad \forall m,k : m^{'} \neq m, k^{'} \neq k; \ \forall i,c,t$$

and

$$p_{m,i,c,t} \leq P_{PF}^{Total} \sum_{k=1}^{k=K} z_{m,i,k,c,t}, \quad \forall m,i,c,t$$

We also found that the following constraints are sufficient to satisfy $C3$ and $C4$ in $\mathcal{HAP}^{Init}$:

- $z_{m,i,k,c,t} \geq \frac{A_m - \Omega}{\frac{g_{i,k,c,t} p_{m,i,c,t}}{\sum_{m=1}^{M} \sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} - \Omega}, \ \forall m,i,k,c,t$
- $g_{i,k,c,t} z_{m,i,k,c,t} \leq B_m \left( \sum_{m=1}^{M} \sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t} P_{m,i',c,t} + \sigma^2 \right), \ \forall m,i,k,c,t$
- $\sum_{c=1}^{C} \sum_{t=1}^{T} z_{m,i,k,c,t} \geq z_{m,i,k,c,t} y_m^{min}, \qquad \forall m,i,k,c,t$
- $\sum_{c=1}^{C} \sum_{t=1}^{T} z_{m,i,k,c,t} \leq y_m^{max}, \qquad \forall m,i,k$

where $A_m = max\left(\left(2^{\frac{F \times R_m^{min}}{y_m^{min} \Delta B \Delta T}} - 1\right), \gamma^{th}\right)$, $B_m = 2^{\frac{F \times R_m^{max}}{y_m^{max} \Delta B \Delta T}} - 1$, $\Omega$ is a constant strictly greater than $A_m$, $y_m^{min}$ and $y_m^{max}$ are predetermined integer constants defined

as the minimum and maximum number of frequency-time slots, respectively, that can be allocated to a multicast session.

Finally our optimization problem can be expressed as :

$$\max_{z_{m,i,k,c,t},p_{m,i,c,t}} \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{t=1}^{T} z_{m,i,k,c,t} \qquad (\mathcal{HAP}_2^{Lagrange})$$

s.t.

$$D1 : z_{m,i,k,c,t} \leq \lambda_{m,i,k}, \quad \forall m,i,k,c,t$$

$$D2 : z_{m,i,k,c,t} + z_{m,i,k',c',t'} \leq 1 + z_{m,i,k,c',t'}, \quad \forall m,i;\forall k,k^{'} : k \neq k^{'};\forall\ (c,t) \neq \left(c^{'},t^{'}\right)$$

$$D3 : z_{m,i,k,c,t} \geq \frac{A_m - \Omega}{\frac{g_{i,k,c,t}p_{m,i,c,t}}{\sum_{m=1}^{M}\sum_{\forall i' \in \mathscr{S}\ i' \neq i} g_{i',k,c,t}p_{m,i',c,t}+\sigma^2} - \Omega}, \quad \forall m,i,k,c,t$$

$$D4 : z_{m,i,k,c,t} \leq \frac{B_m\left(\sum_{m=1}^{M}\sum_{\forall i' \in \mathscr{S}\ i' \neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2\right)}{g_{i,k,c,t}p_{m,i,c,t}}, \quad \forall m,i,k,c,t$$

$$D5 : z_{m',i,k',c,t} \leq 1 - z_{m,i,k,c,t}, \quad \forall m,k : m^{'} \neq m, k^{'} \neq k;$$

$$D6 : \sum_{c=1}^{C}\sum_{t=1}^{T} z_{m,i,k,c,t} \geq z_{m,i,k,c,t}y_m^{min}, \quad \forall m,i,k,c,t$$

$$D7 : \sum_{c=1}^{C}\sum_{t=1}^{T} z_{m,i,k,c,t} \leq y_m^{max}, \quad \forall m,i,k$$

$$D8 : z_{m,i,k,c,t} \in \{0,1\}, \quad \forall m,i,k,c,t$$

$$D9 : \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{c=1}^{C} p_{m,i,c,t} \leq P_{PF}^{Total}, \quad \forall t$$

$$D10 : p_{m,i,c,t} \leq P_{PF}^{Total}\sum_{k=1}^{k=K} z_{m,i,k,c,t}, \quad \forall m,i,c,t$$

$$D11 : p_{m,i,c,t} \geq 0, \quad \forall m,i,c,t.$$

The formulation $\mathcal{HAP}_2^{Lagrange}$ turns out to be a *mixed integer non-linear program*

(MINLP) for which a branch and bound (BnB) algorithm can be used for solving the problem [58]. In the next section, we the propose a technique to obtain solution bounds that could be used in BnB for pruning.

## 3.3 Proposed Solution Techniques

In this section, solution bounds for $\mathcal{HAP}_2^{Lagrange}$, are obtained by constructing a Lagrangian relaxation problem LR. The relaxed constraints are replaced with a penalty term in the objective function involving the amount of violation of the constraints and their dual variables. For the minimization part, the *subgradient* algorithm is used to solve for the dual variable vector $\mathbf{u}$ in LR [17, 22].

$$f(\mathbf{u}, \mathbf{z}, \mathbf{p}) = \min_{\mathbf{u}} \left\{ \max_{\mathbf{z},\mathbf{p}} \left( \mathbf{e}^T.\mathbf{z} + \mathbf{u} \left( \mathbf{b}_1(\mathbf{z}, \mathbf{p}) - \mathbf{b}_2(\mathbf{z}, \mathbf{p}) \right) \right) \right\} \qquad \text{(LR)}$$

s.t.

$$C^*1 : \mathbf{u} \geq \mathbf{0},$$

$$C^*2 : \mathbf{z}, \mathbf{p} \in F.S.,$$

$$C^*3 : z_{m,i,k,c,t} \in \{0, 1\}, \quad \forall m, i, k, c, t$$

$$C^*4 : \mathbf{p} \geq \mathbf{0}.$$

where $\mathbf{u}$ is the dual variable row vector, $\mathbf{z}$ is a column vector whose elements are $z_{m,i,k,c,t}$, $\mathbf{p}$ is a column vector whose elements are $p_{m,i,c,t}$ ,$F.S.$ is the feasible region of the relaxed problem, $\mathbf{e}$ is a column vector of ones, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the right hand side (R.H.S) and left hand side (L.H.S) column vectors respectively of the dualized constraints and are functions in the vectors $\mathbf{z}$ and $\mathbf{p}$.

For the maximization part, we solve two separate subproblems for the variable

vectors $\mathbf{z}$ and $\mathbf{p}$. We propose three different configurations for dualization of constraint sets in solving for the vector $\mathbf{z}$. Each configuration exploits the structure of the relaxed problem resulting from the corresponding relaxed constraint sets. The basis of the approach is to solve the problem in two phases iteratively where in Phase 1, we solve subproblem 1 for $\mathbf{z}$ while using a constant $\mathbf{p}$ obtained from the previous iteration of the Lagrangian relaxation problem, and in Phase 2, the opposite is done by solving subproblem 2 for $\mathbf{p}$ while fixing $\mathbf{z}$ at the values obtained from Phase 1. For both phases, we have linear optimization problems with special structures that we could exploit to solve easily. The first subproblem is function in $\mathbf{z}$ (Phase 1) and considers joint scheduling, subchannel allocation and user-to-group admission. The second subproblem is function in $\mathbf{p}$ (Phase 2) and considers the power allocation for each multicast session across the entire HAP service area. Both subproblems are a decomposition of the Lagrangian relaxation problem LR.

Solving iteratively on two phases can enhance the solution of the maximization part in the Lagrangian problem LR or at least does not worsen it. The larger the value of the objective function, for a given dual vector $\mathbf{u}$, the smaller the duality gap, which implies better 'bound goodness'. We label this procedure, in the rest of the chapter as the *primal solution algorithm*. For a given dual vector $\mathbf{u}^l$ in a *subgradient* algorithm iteration $l$, solving for $\mathbf{z}$ in Phase 1 of the first iteration requires fixing $\mathbf{p}$ to an arbitrary value $\mathbf{p}^*$ that satisfies the constraints $C^*2$ and $C^*4$ of problem LR. The obtained solution for $\mathbf{z}^*$ can then be substituted in LR to obtain subproblem 2 to find the value for $\mathbf{p}$ in Phase 2, which is either a better solution or in the worst case a solution that yields the same value for the objective function obtained from the previous phase. We believe that the new solution for $\mathbf{p}$ will not be worse than the older one since both solutions are feasible to the Lagrangian relaxation problem

LR, hence either the new solution is better than the older one or at least is equal to it.

The *primal solution algorithm* continues solving Phase 1 and Phase 2 iteratively until one of the stopping criteria is satisfied:

1. The duality gap given by :

$$\varrho = 100 - 100 \left( \frac{f^*_{i,j}(\mathbf{u}^l, \mathbf{z}, \mathbf{p})}{f_{i,j}(\mathbf{u}^l, \mathbf{z}, \mathbf{p})} \right) \leq \epsilon_o \tag{3.4}$$

   where $f_{i,j}(\mathbf{u}^l, \mathbf{z}, \mathbf{p})$ is the value of the Lagrangian relaxation objective function of LR after solving Phase $i$ in iteration $j$ of the proposed technique, $f^*_{i,j}(\mathbf{u}^l, \mathbf{z}, \mathbf{p})$ is the best feasible solution obtained up to that point and $\epsilon_o$ is the acceptable duality gap in percent.

2. The solution does not improve in a specified number of iterations $N$.

3. The maximum iteration limit, $J$, is reached.

If there is no improvement achieved after a number of iterations $N$ for the *primal solution algorithm*, or if the improvement is slow such that the number of iterations reaches the maximum allowable iterations $J$ before an acceptable duality gap is achieved, the *subgradient* algorithm is invoked can move on to the next iteration $l + 1$ to calculate the value of the new dual variable vector $\mathbf{u}^{l+1}$ in its path to find the optimal solution $\mathbf{u}^{**}$ of the dual problem. A the end of this section, the flowchart in Figure 3.4 illustrates the entire solution bounding algorithm involving the *primal solution algorithm* and the subgradient algorithm. In the next subsection we show the different dualization configurations and the corresponding solution methods for subproblem 1 that solves the Lagrangian relaxation problem LR for $\mathbf{z}$ (Phase1) of the *primal solution algorithm*.

### *3.3.1 Proposed Solution Methods for Phase 1 and their Corresponding Dualizations*

The following are the three different proposed dualization configurations and their corresponding proposed solution methods for the Phase 1 subproblem to solve for **z**

- **Method1**: We dualize all the constraint sets in $\mathcal{HAP}_2^{Lagrange}$ except constraint $D7$,

$$\sum_{c=1}^{C}\sum_{t=1}^{T} z_{m,i,k,c,t} \leq y_m^{max}, \quad \forall m,i,k$$

  and $D8$,$z_{m,i,k,c,t} \in \{0,1\}$, $\quad \forall m,i,k,c,t$. The Lagrangian relaxation problem can then be further broken into $M \times S \times K$ independent sub-subproblems as show in Figure 3.3 that can all be solved separately. Those sub-subproblems can each be solved to optimality using a heap sort algorithm to find the highest $y_m^{max}$ objective function coefficients of the Lagrangian relaxation problem. The strategy is to go through the coefficients once, and keep a list of the highest $y_m^{max}$ elements found up to that point. This is done efficiently by always knowing the smallest element in this top-$y_m^{max}$, so that it can be possibly replaced by a larger one. The heap structure makes it easy to maintain this list without wasting any effort. This way it does enough of the sort to find the smallest element, and that is why it is fast. The worst case complexity is $O\left(MSKCT \log\left(\max_m (y_m^{max})\right)\right)$ [59].

- **Method2**: Dualizes all constraints except $D1$ and either $D2$ or $D3$ in problem LR or both. The Lagrangian problem can then be solved by setting $z_{m,i,k,c,t}$ to '1' those cases which satisfy:

  i $\lambda_{m,i,k} = 1$ and

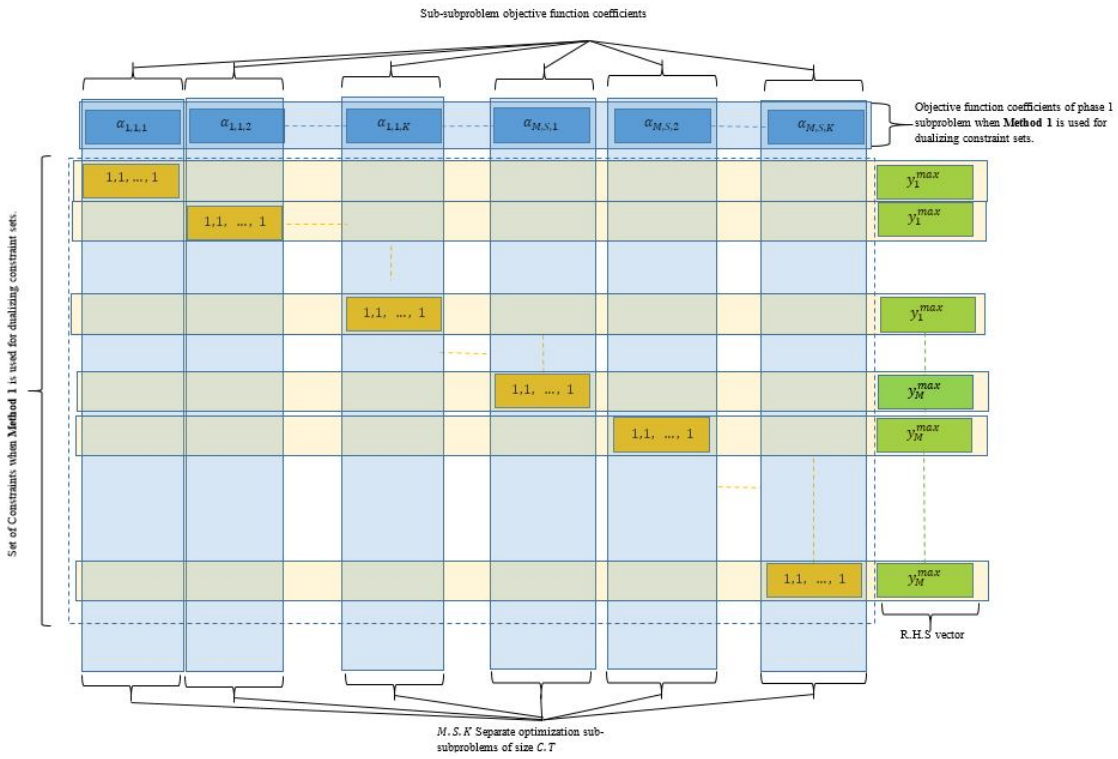



Figure 3.3: Structure of Phase 1 subproblem when Method 1 is used for dualizing constraint sets.

ii $g_{i,k,c,t}p_{m,i,c,t} \leq B_m(\sum_{m=1}^{M}\sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2)$ and/or

iii $g_{i,k,c,t}p_{m,i,c,t} \geq A_m(\sum_{m=1}^{M}\sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2)$.

This can be done for all the $z_{m,i,k,c,t}$ variables separately and the solution to the Lagrangian relaxation problem will still be optimal. The worst case complexity is derived to be $O\left(MS^2KCT\right)$.

- **Method3**: Relaxes only one constraint set of those which have any of the variables $p_{m,i,c,t}$, $D2$ and/or $D3$ in problem $\mathcal{HAP}_2^{Lagrange}$, to bring in $p_{m,i,c,t}$ in the objective function such that in Phase 2, the subproblem2 becomes a linear program (LP) in $p_{m,i,c,t}$. This problem can be solved using branch and bound with LP relaxation to obtain the bounds at each node in the tree [58]. The BnB

algorithm may stop either after the optimal solution to Phase 2 subproblem is obtained or at an acceptably close to optimal feasible solution. The worst case complexity of BnB occurs when it requires examination of all the nodes in its tree which makes its worst case complexity $O(2^{MSKCT})$.

### *3.3.2 Power Allocation Subproblem (Phase 2) and its Proposed Solution Method*

Solving for the power vector $\mathbf{p}$ could be done using the simplex algorithm [58] for any of the three proposed dual configurations. This is because for the fixed value of the vector $\mathbf{z}$ obtained from Phase 1, the Lagrangian relaxation LR is a linear program in $\mathbf{p}$. However for dualization configuration 1 (Method 1), a greedy algorithm can be used to solve a special structured linear program, which will be shown in this section to be a set of separate fractional knapsack (FKS) problems. Greedy algorithms are well known to have low polynomial (usually quadratic) worst case complexities which are computationally quite efficient compared to the general simplex algorithm, whose worst case complexity is exponential with respect to the size of the subproblem.

For a *subgradient* algorithm iteration $l$ and a given dual variable vector $\mathbf{u}^l$, the power allocation linear optimization problem is given by:

$$\max_{\mathbf{p}} \quad \left\{ \mathbf{e}^T.\mathbf{z}^* + \mathbf{u}^l.\mathbf{b}.\left(z^*_{m,i,k,c,t}\right) - \mathbf{u_1}^l.\widehat{\mathbf{A}}_1.\mathbf{z}^* - \mathbf{u}_2^l.\widehat{\mathbf{A}}_2^p\left(z^*_{m,i,k,c,t}\right).\mathbf{p} \right\} \qquad (\mathbf{PW})$$

s.t.

$$L1^*: \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{c=1}^{C} p_{m,i,c,t} \leq P_{PF}^{Total}, \qquad \forall t$$

$$L2^*: p_{m,i,c,t} \geq 0, \qquad \forall m,i,c,t$$

where

$$\mathbf{b}\left(z_{m,i,k,c,t}\right) = \begin{bmatrix} \widehat{\mathbf{b}_1} \\ \widehat{\mathbf{b}_2}\left(z_{m,i,k,c,t}\right) \end{bmatrix},$$

$$\widehat{\mathbf{b}}_1 = \begin{bmatrix} \tilde{\mathbf{b}}_1 \\ \tilde{\mathbf{b}}_2 \\ \tilde{\mathbf{b}}_5 \\ \tilde{\mathbf{b}}_6 \end{bmatrix}, \widehat{\mathbf{b}}_2\left(z_{m,i,k,c,t}\right) = \begin{bmatrix} \tilde{\mathbf{b}}_3 \\ \tilde{\mathbf{b}}_4 \\ \tilde{\mathbf{b}}_{10}\left(z_{m,i,k,c,t}\right), \end{bmatrix}$$

$$\widehat{\mathbf{A}}_1 = \begin{bmatrix} \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_2 \\ \tilde{\mathbf{A}}_5 \\ \tilde{\mathbf{A}}_6 \end{bmatrix}, \widehat{\mathbf{A}}_2\left(z_{m,i,k,c,t}\right) = \begin{bmatrix} \tilde{\mathbf{A}}_3\left(z_{m,i,k,c,t}\right) \\ \tilde{\mathbf{A}}_4\left(z_{m,i,k,c,t}\right) \\ \tilde{\mathbf{A}}_{10} \end{bmatrix}, \mathbf{u}^T = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \end{bmatrix}.$$

$\mathbf{e}$ is a column vector of ones, $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \tilde{\mathbf{b}}_5, \tilde{\mathbf{b}}_6$, are the right hand side (R.H.S) vectors of constraints $D1, D2, D5, D6$ in $\mathcal{HAP}_2^{Lagrange}$ and $\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2, \tilde{\mathbf{A}}_5, \tilde{\mathbf{A}}_6$ are their corresponding left hand side (L.H.S) constraint coefficient matrices. The vectors $\tilde{\mathbf{b}}_3^p, \tilde{\mathbf{b}}_4^p, \tilde{\mathbf{b}}_{10}^p\left(z_{m,i,k,c,t}\right)$ are R.H.S vectors of the constraints $D3, D4, D10$ in $\mathcal{HAP}_2^{Lagrange}$ when written in the form:

$$D3^p: -g_{i,k,c,t}p_{m,i,c,t}z_{m,i,k,c,t} + A_m\left(\sum_{m=1}^{M}\sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2\right) \leq 0, \quad \forall m,i,k,c,t \tag{3.5}$$

$$D4^p : g_{i,k,c,t} z_{m,i,k,c,t} p_{m,i,c,t} - B_m \left( (\sum_{m=1}^{M} \sum_{\substack{\forall i' \in \mathscr{S} \\ i' \neq i}} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2 \right) \leq 0, \quad \forall m, i, k, c, t \tag{3.6}$$

$$D10^p : p_{m,i,c,t} \leq P_{PF}^{Total} \sum_{k=1}^{k=K} z_{m,i,k,c,t}, \quad \forall m, i, c, t \tag{3.7}$$

and $\tilde{\mathbf{A}}_3^p \left( z_{m,i,k,c,t} \right), \tilde{\mathbf{A}}_4^p \left( z_{m,i,k,c,t} \right), \tilde{\mathbf{A}}_{10}^p$ are their L.H.S coefficient matrices respectively.

Swapping columns of the functional constraint matrix of **PW** and its objective function terms as expressed by the indices of $p_{m,i,c,t}$ to $p_{t,m,i,c}^{swapped}$ gives us $T$ independent and separate linear programs that can be solved in parallel. Furthermore, by inspection for each of the $t$ subproblems, if the L.H.S and R.H.S of each of the functional constraints are divided by the R.H.S of the constraint, which is $P_{PF}^{Total}$ , the functional constraint for subproblem $t$ can be rewritten as $\sum_{m=1}^{M} \sum_{i=1}^{S} \sum_{c=1}^{C} \tilde{p}_{t,m,i,c}^{swapped} \leq 1, \quad \forall t$ where $\tilde{p}_{t,m,i,c}^{swapped} = \frac{p_{t,m,i,c}^{swapped}}{P_{PF}^{Total}}$. We define the swapped-normalized power decision vector as:

$$\tilde{\mathbf{p}}^{swapped} = \left[ \tilde{\mathbf{p}}_1^{swapped}, \tilde{\mathbf{p}}_2^{swapped}, ..., \tilde{\mathbf{p}}_t^{swapped}, ..., \tilde{\mathbf{p}}_T^{swapped} \right]^T \tag{3.8}$$

where $\tilde{\mathbf{p}}_t^{swapped}$ is a column vector whose elements are $\tilde{p}_{t,m,i,c}^{swapped}$. The objective function coefficients for **PW**, also get swapped according to the new arrangement of the power variables in vector $\tilde{\mathbf{p}}^{swapped}$. For the sake of brevity, let us refer to the swapped objective function coefficients of **PW** by the row vector $\Upsilon$, which is defined as:

$$\Upsilon = [\Gamma_1, ..., \Gamma_t, ..., \Gamma_T] \tag{3.9}$$

where $\Gamma_t$ is a row vector that comprises the objective function coefficients of the $t^{th}$ power allocation subproblem. The column swapping modification yields an optimization problem that has a similar structure as that shown earlier in Figure 3.3, with slight differences of an all ones R.H.S vector and the objective function vector $\Upsilon$.

Then a *Merge-Sort* algorithm sorts the coefficients of each of the vectors $\Gamma_t$ separately in descending order. For each sub-subproblem, the variable in $\tilde{\mathbf{p}}_t^{swapped}$ which corresponds to the first sorted coefficient, is set to 1 while the rest are set to zeros. Hence the complexity encountered here is only the complexity of the *Merge-Sort* algorithm which is known to be $O\left(nlog\left(n\right)\right)$ [60], where $n = M.S.C$ for each sub-subproblem. All the $t$ sub-subproblems could be solved simultaneously in parallel if the HAP has more than $T$ processors on-board.

### *3.3.3 Solving The Dual Problem*

To solve for the dual variable vector $\mathbf{u}$ in LR, the *subgradient* algorithm is used to calculate the search direction and step size $\omega_l$ at each iteration $l$. As in [22] and [17], the direction is given by the vector $\left(\mathbf{b}_1\left(\mathbf{z},\mathbf{p}\right) - \mathbf{b}_2\left(\mathbf{z},\mathbf{p}\right)\right)$ obtained from the relaxed constraints. The formula used to determine the sequence of values of the dual variable vector $\mathbf{u}^l$ in every iteration is given by [17]:

$$\mathbf{u}^{l+1} = max\left\{0, \mathbf{u}^l - \omega_l\left(\mathbf{b}_1\left(\mathbf{z}^*,\mathbf{p}^*\right) - \mathbf{b}_2\left(\mathbf{z}^*,\mathbf{p}^*\right)\right)\right\}. \tag{3.10}$$

For the step size $\omega_l$, we compare three different step size rules, these are:

1. **Step size rule 1**:Square summable but not summable [18] which satisfies :

$$\sum_{l=1}^{\infty}\omega_l^2 < \infty \;\; and \;\; \sum_{l=1}^{\infty}\omega_l = \infty$$

where $\omega_l$ is the step size at the $l^{th}$ iteration. Therefore we choose

$$\omega_l = \frac{1}{l}. \tag{3.11}$$

2. **Step size rule 2**:Nonsummable diminishing [18] which satisfies :

$$\lim_{l\to\infty} \omega_l = 0 \quad and \quad \sum_{k=1}^{\infty} \omega_l = \infty$$

and hence we choose

$$\omega_l = \frac{1}{\sqrt{l}}. \tag{3.12}$$

3. **Step size rule 3**:The formula that Fisher suggested in his papers [22] and [17]:

$$\omega_l = \frac{\rho_l \left( f\left(\boldsymbol{u}^l\right) - f^* \right)}{\left\| \mathbf{b}_1\left(\mathbf{z},\mathbf{p}\right) - \mathbf{b}_2\left(\mathbf{z},\mathbf{p}\right) \right\|^2} \tag{3.13}$$

where $\rho_l$ is a scalar satisfying $0 < \rho_l \leq 2$.

The termination conditions for the sub-gradient algorithm are chosen to be any of the following that get encountered first:

1. $f(\boldsymbol{u}) < f^*$, where $f^*$ is the best feasible solution found and $f(\boldsymbol{u})$ is the upper bound.

2. An acceptable duality gap (%).

3. A maximum iteration limit $L$.

The entire procedure used to find the upper solution bound for the problem $\mathcal{HAP}_2^{Lagrange}$ is summarized by the flowchart in Figure 3.4 where the *primal solution algorithm* is the iterative procedure in which phases 1 and 2 are invoked and the *subgradient* algorithm is used to obtained the optimal solution for the dual variables. The orange portion of the flowchart represents the *primal solution algorithm* while the blue portion represents the *subgradient* algorithm. In one iteration of *subgradient*

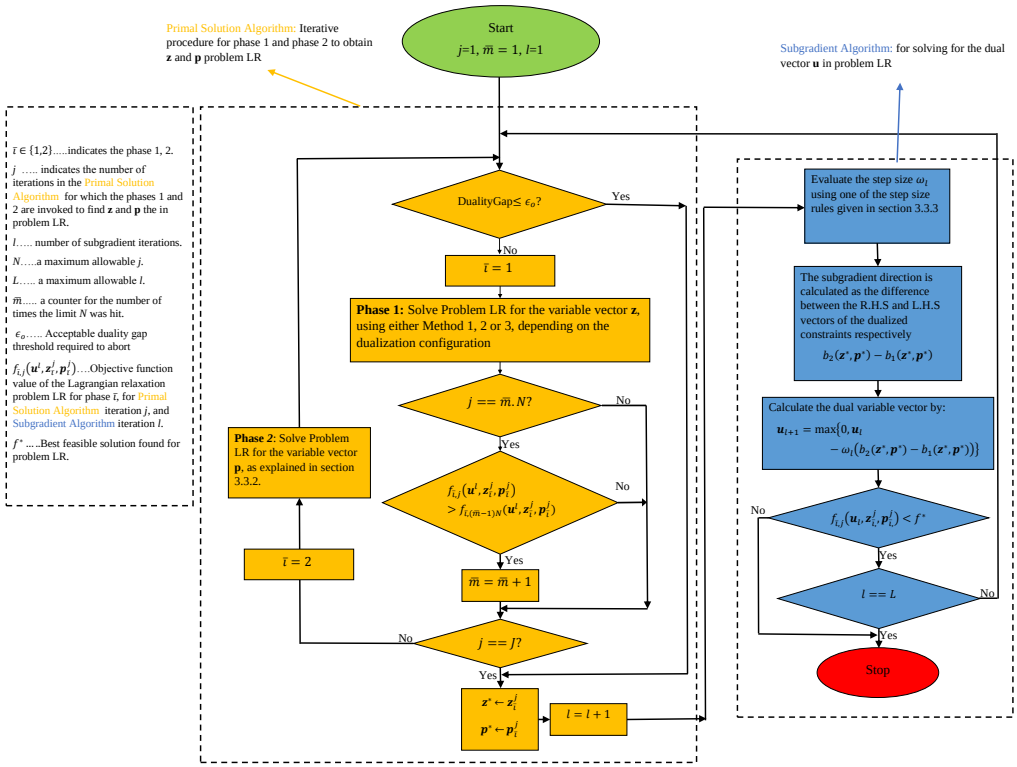



Figure 3.4: Solution bounding subroutine flowchart.

algorithm, the step size and direction are calculated and the *primal solution algorithm* is executed until any of the stopping criteria mentioned earlier in this section are encountered.

## 3.4 Numerical and Simulation Results

In this section we present some illustrative numerical and simulation results for both phases 1 and 2. First we perform some numerical experiments with parameters that yield a medium sized optimization problem. The aim of these experiments is to numerically compare the bounds obtained by different solution methods, find the

most suitable step size formula to use and to find out whether the initial values of the dual variable vectors affect the bounds. Based on the obtained results, we use the step size formula that performed best in a realistic simulation scenario that is described in detail later to numerically observe and compare the results of the proposed Lagranigan relaxation based solution methods in terms of bounds.

For initial numerical experiments for phase 1, the bound goodness is calculated as

$$\overline{\varrho} = 100 \times \left( \frac{f^*}{f\left(\mathbf{u}, \mathbf{z}, \mathbf{p}^*\right)} \right), \tag{3.14}$$

where

- $f\left(\mathbf{u}, \mathbf{z}, \mathbf{p}^*\right)$ is the upper bound value for Phase 1 subproblem for a power vector fixed at $\mathbf{p} = \mathbf{p}^*$ ,
- $f^*$ is the best feasible solution obtained for Phase 1 subproblem.

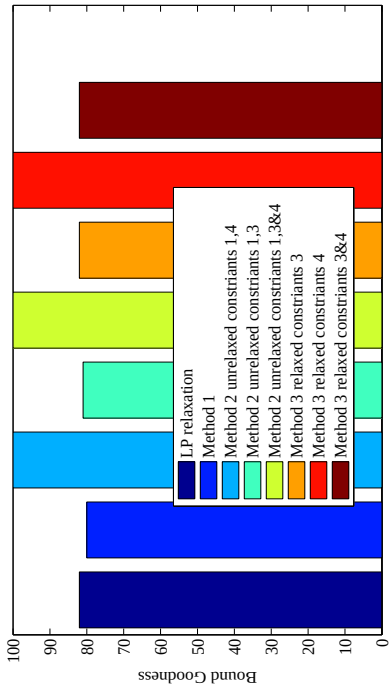
The optimization problem for $M = 2$, $S = 2$, $K = 4$ ,$C = 2$, $T = 2$, $y_m^{min} = 1\ \forall m$, $y_m^{max} = 3\ \forall m$ is considered and the rest of the numerical parameters of the problem are shown in Table 3.2. The parameter $\rho_l$ of **step size rule 3** begins with a value of 2 and is reduced by half if the bound does not decrease for 500 iterations. LP relaxation is used as a reference scheme, where the integer constraint of $z_{m,i,k,c,t}$ is relaxed to take any values in the range $0 \leq z_{m,i,k,c,t} \leq 1$.

Figure 3.5a shows the goodness of the bounds obtained using Methods 1, 2 and 3 by using **step size rule 1** compared to that of LP relaxation when the initial dual variable vector $\boldsymbol{u}$ is set to $\mathbf{0}$. For all the three methods, the *subgradient* algorithm converged to bounds with the goodness shown on the chart and remained at those values for thousands of iterations before the algorithm reached its maximum iteration limit. They all seem to perform almost at least as the LP relaxation in the worst case.

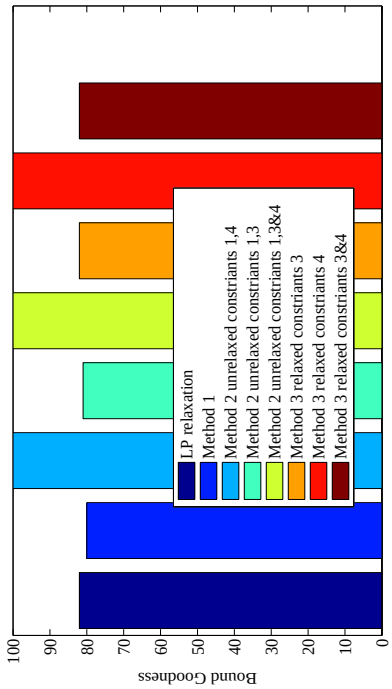
Table 3.2: Experimental Values For Model Parameters

| Model Parameters | Values |
|---|---|
| Noise power spectral density ($N_o$) | -173 dBm/Hz |
| OFDMA frame length ($F$) | 10 ms |
| Minimum capacity requirements ($R_m^{min}$) | $R_1^{min}$=0.5 Mbps,$R_2^{min}$=0.75 Mbps |
| Maximum capacity requirements ($R_m^{max}$) | $R_1^{max}$=6 Mbps, $R_2^{max}$=7 Mbps |
| Total Bandwidth | 1MHz |
| Total HAP Power | 120 Watt |
| Target BER | $10^{-6}$ |
| Constellation Size | 2 |
| Carrier Frequency | 28 GHz |
| $\lambda_{m,i,k}$ | $\lambda_{1,1,1}$= $\lambda_{1,1,2}$=$\lambda_{2,2,3}$= $\lambda_{2,2,4}$=1 and zero for others |

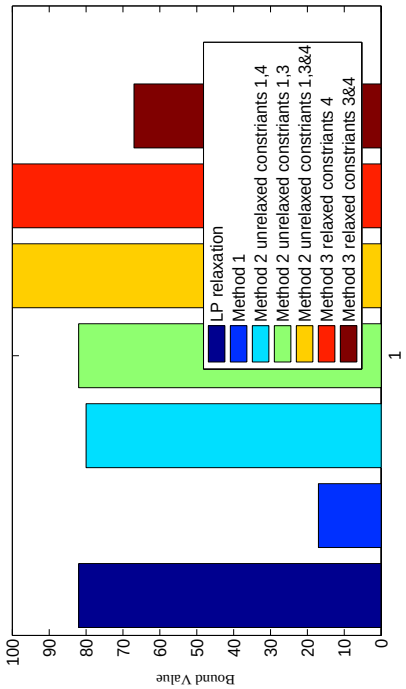
Figures 3.5b and 3.5c are similar charts for **step size rules 2 and 3**, respectively. We can see that the bounds provided by **step size rule 3** are the worst. This is because there are no guarantees that they converge to the optimal solution $f(\boldsymbol{u}^*)$ since according to [22], **step size rule 3** does not satisfy the sufficient conditions of correct convergence given by $\lim_{l\to\infty}\omega_l = 0$ and $\sum_{l=1}^{\infty}\omega_l = \infty$. As a result, some of the relaxations perform much poorer than LP relaxation using this step size rule as the chart in Figure 3.5c shows. The charts show that among the three proposed solution methods, Method 1 has the lowest bound goodness and, depending upon the constraint sets relaxed, the highest is either Method 2 or Method 3. However, it is worth mentioning that according to the worst case complexity mentioned previously in this chapter, Method 1 has the lowest complexity. Therefore, there is a trade-off between Methods 1 and 2 in terms of bound goodness and worst case complexity.




(b) Step size rule no. 2.




(a) Step size rule no. 1.




(c) Step size rule no. 3.

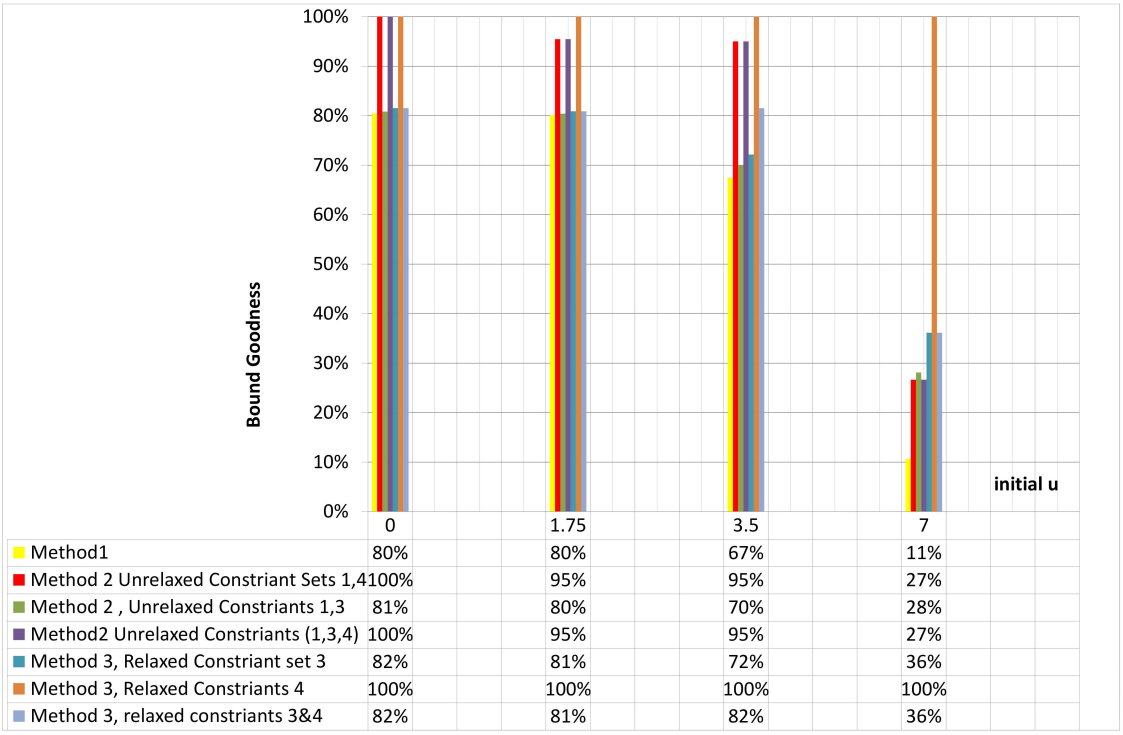
Figure 3.5: Goodness of bounds for the three proposed Lagrangian relaxation based solution methods for Phase 1.

Figures 3.6a, 3.6b and 3.6c show the bound goodness for each of the proposed solution methods versus the initial values of the dual variables for step size rules 1, 2 and 3 respectively. For Figure 3.6a, the difference in bound goodness for different initial values of **u** is relatively large because the convergence using step **size rule 1** is not as fast as that in Figure 3.6b (**step size rule 2**). The faster convergence for step size rule 2 compared to that of 1 is expected due to the higher dampness. Figures 3.7a and 3.7b support our argument which give the absolute value of the obtained bounds versus the number of iterations of the *subgradient* algorithm for step size rules 1 and 2 respectively. For those initial points that do not lead to convergence in the iteration limit mentioned previously, convergence is expected for a larger iteration limit.

For **step size rule 3** in Figure 3.6c, the difference in the goodness of bounds due to different initial dual variable values is huge. The reason is that, as Figure 3.7c shows, convergence is too slow to the extent that leads us to question if it ever converges to the same solution. We believe that this is due to the fact that it does not satisfy the sufficient conditions for convergence as Fisher mentioned in his paper [22]. **Step size rule 2** seems to converge quickly to the same solution even if the initial point differs and hence is the best choice for our problem.

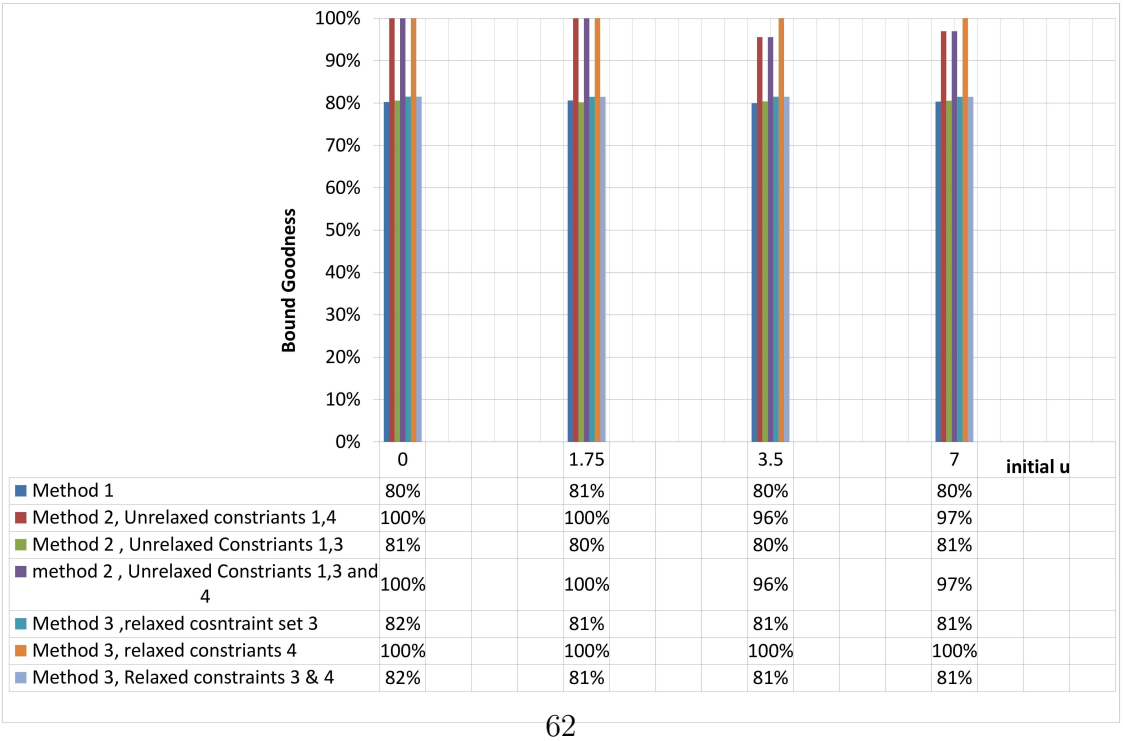
We then vary the optimization problem size by varying the number of users terminals ($K$) in the range 3-7. All the other model parameters that we use remain the same as those in Table 3.2. Consider **step size rules 1 and 2**, Figures 3.8a and 3.8b show that proposed Method 2 performs better than others in terms of bound goodness. For **step size rule 3**, the bound goodness is poor compared to **step size rules 1 and 2** as Figure 3.8c shows. These results are consistent with the previous results.

A more realistic simulation was conducted for a cell surrounded by its six first tier




| | 0 | 1.75 | 3.5 | 7 |
|---|---|---|---|---|
| Method1 | 80% | 80% | 67% | 11% |
| Method 2 Unrelaxed Constriant Sets 1,4 | 100% | 95% | 95% | 27% |
| Method 2 , Unrelaxed Constriants 1,3 | 81% | 80% | 70% | 28% |
| Method2 Unrelaxed Constriants (1,3,4) | 100% | 95% | 95% | 27% |
| Method 3, Relaxed Constriant set 3 | 82% | 81% | 72% | 36% |
| Method 3, Relaxed Constriants 4 | 100% | 100% | 100% | 100% |
| Method 3, relaxed constriants 3&4 | 82% | 81% | 82% | 36% |

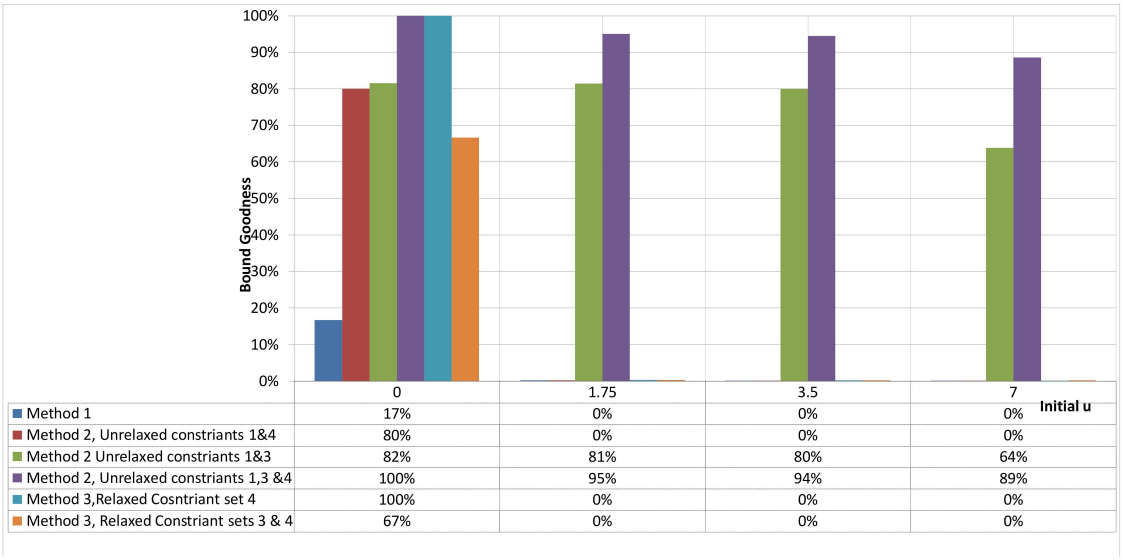
(a) Step size rule no.1.




| | 0 | 1.75 | 3.5 | 7 |
|---|---|---|---|---|
| Method 1 | 80% | 81% | 80% | 80% |
| Method 2, Unrelaxed constriants 1,4 | 100% | 100% | 96% | 97% |
| Method 2 , Unrelaxed Constriants 1,3 | 81% | 80% | 80% | 81% |
| method 2 , Unrelaxed Constriants 1,3 and 4 | 100% | 100% | 96% | 97% |
| Method 3 ,relaxed cosntraint set 3 | 82% | 81% | 81% | 81% |
| Method 3, relaxed constriants 4 | 100% | 100% | 100% | 100% |
| Method 3, Relaxed constraints 3 & 4 | 82% | 81% | 81% | 81% |



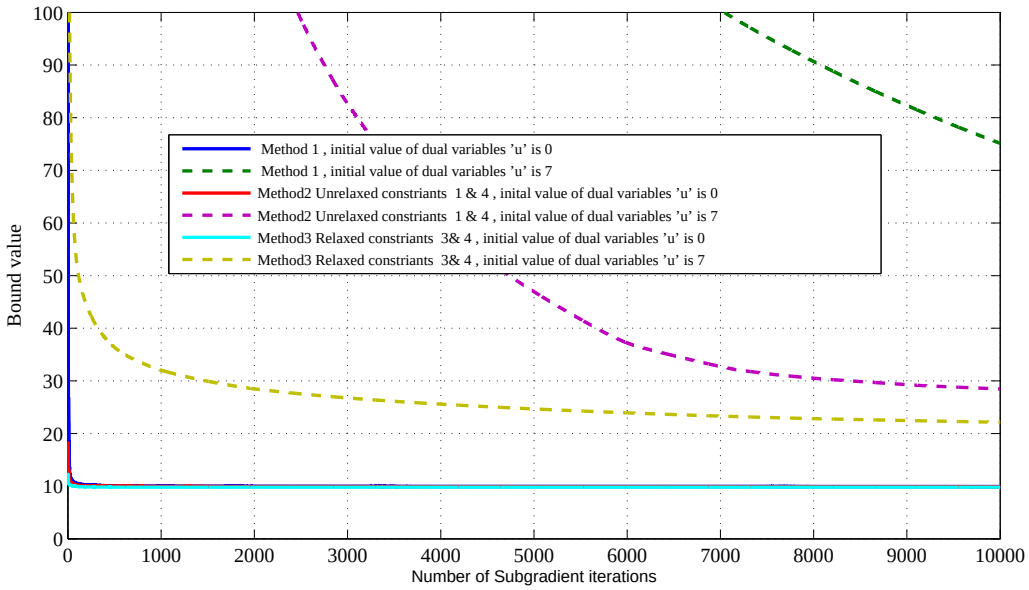
(b) Step size rule no. 2.



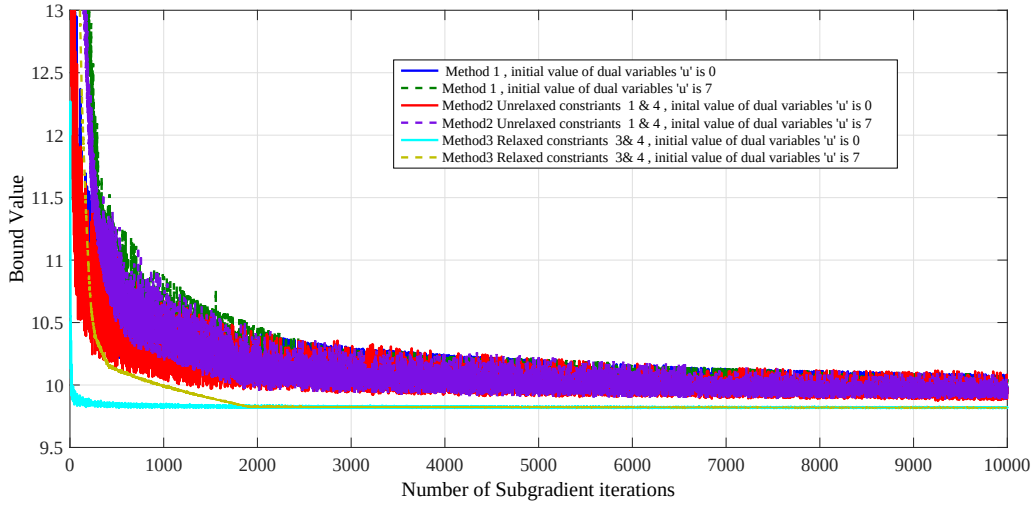

| | 0 | 1.75 | 3.5 | 7 |
|---|---|---|---|---|
| Method 1 | 17% | 0% | 0% | 0% |
| Method 2, Unrelaxed constriants 1&4 | 80% | 0% | 0% | 0% |
| Method 2 Unrelaxed constriants 1&3 | 82% | 81% | 80% | 64% |
| Method 2, Unrelaxed constriants 1,3 &4 | 100% | 95% | 94% | 89% |
| Method 3,Relaxed Cosntriant set 4 | 100% | 0% | 0% | 0% |
| Method 3, Relaxed Constriant sets 3 & 4 | 67% | 0% | 0% | 0% |

(c) Step size rule no. 3.

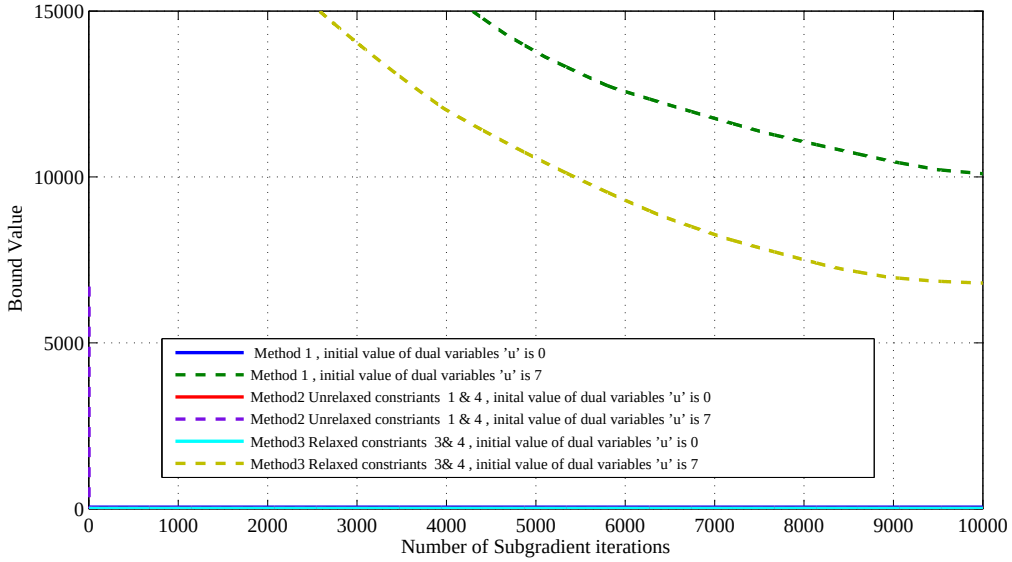
Figure 3.6: Goodness of bounds for the three proposed Lagrangian relaxation based solution methods at different initial dual variable values.




(a) Step size rule no.1.




(b) Step size rule no. 2.

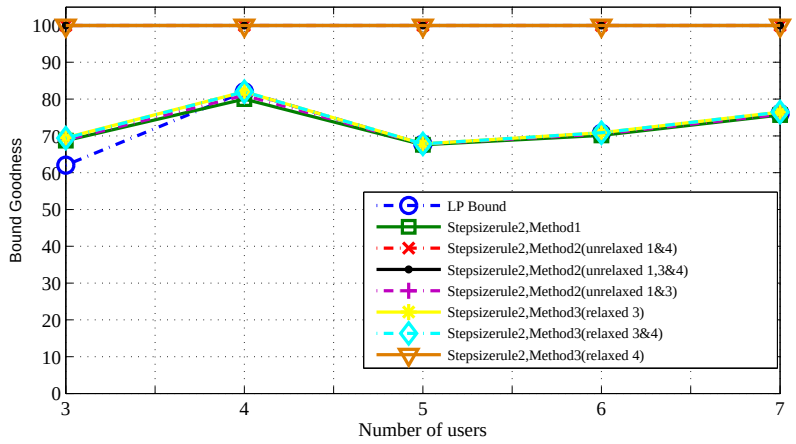



(c) Step size rule no. 3.

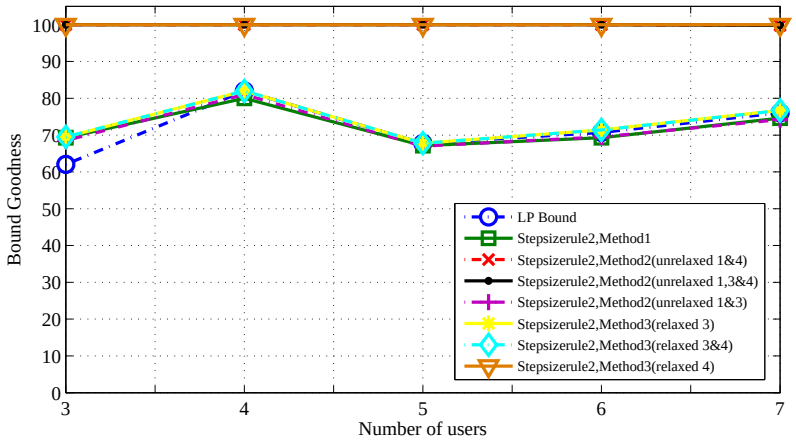
Figure 3.7: Bounds vs Subgradient algorithm iterations for the three proposed Lagrangian relaxation based solution methods.

neighbor cells of the HAP service area (i.e. 7 cells total). Each cell has a radius of 5 km and the users are uniformly distributed around its center. The HAP subplatform (SS) point coincides with the center of the middle cell. The following parameters and values are used in our simulation :

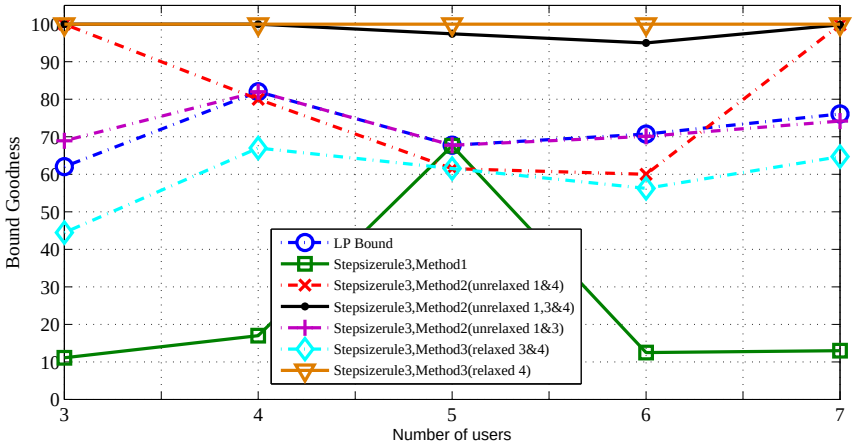
- HAP height is 20 Km.
- Total platform power $P_{PF}^{Total} = 7W$.
- Initial power distribution $p_{m,i,c,t} = P_{PF}^{Total}/M.S.C$ is uniform across all subchannels , cells and multicast sessions.
- Four subchannels.
- Two time slots.




(a) Step size rule no.1.




(b) Step size rule no. 2.




(c) Step size rule no. 3.

Figure 3.8: Bound Goodness for the Three Proposed Methods Versus the Number of Users (K).

- Two multicast sessions in the service area.
- $R_m^{min} = 256Kbps$ and $R_m^{max} = 10Mbps$.
- $y_m^{min} = 1$ for $m = 1, 2$, while $y_1^{max} = 6$ and $y_2^{max} = 4$.
- The values of noise power spectral density, OFDMA frame length, total bandwidth, target BER and carrier frequency are in Table 3.2.

For the channel model, we consider the following :

- Free space path loss was for large scale fading, (i.e. path-loss exponent of 2).
- a rain attenuation model over the first 3 km of height as the one used in [56], given as:

$$A\left(d_k\right) = 10^{\left(3d_k\chi/10h\right)} \tag{3.15}$$

  where $h$ is the HAP height , $\chi$ is the attenuation through the clouds and rain in $dB/km$.

- For the small scale fading we use Ricean fading with a Rice factor of 20 dB.

For the HAP antennas we use the aperture antenna model that was used in [57] in which the gain of the main lobe with respect to the boresight angle is defined as:

$$G_H\left(\varpi\right) = Ap_{eff} \cdot cos\left(\varpi\right)^n \frac{32log2}{2\left(2arccos\left(\sqrt[n]{0.5}\right)\right)^2} \tag{3.16}$$

where $Ap_{eff}$ is the aperture efficiency which we consider to be unity and $n$ is the roll-off factor of the antenna. The roll-off factor value is chosen to maximize the gain at the cell borders as in [57]. The side lobes are modeled by a flat level at -40 dB.

Each user is considered to have parabolic antenna, whose gain when receiving a transmission from the HAP antenna of the cell to which the user belongs to is given

by:

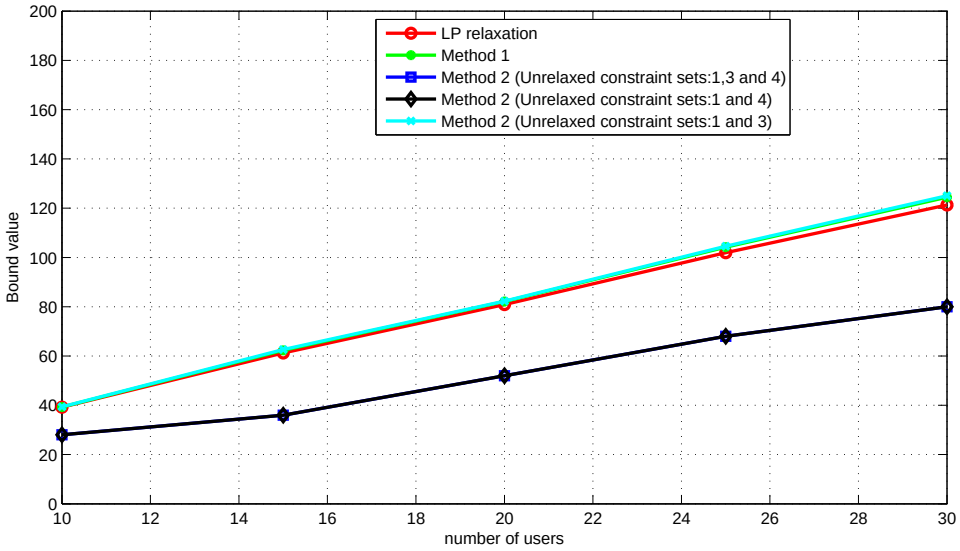
$$G_u = A_{eff}^u \left( \frac{\pi D f_c}{\check{C}_{light}} \right)^2 \tag{3.17}$$

where $A_{eff}^u$ is the aperture efficiency of the parabolic antenna and we take that to be unity in our simulation and $D$ is the diameter of the parabolic reflector. Any received transmission from any other HAP antenna of the other cells falls in the sidelobes of the user antenna which is modelled by a flat level of 0 dB.

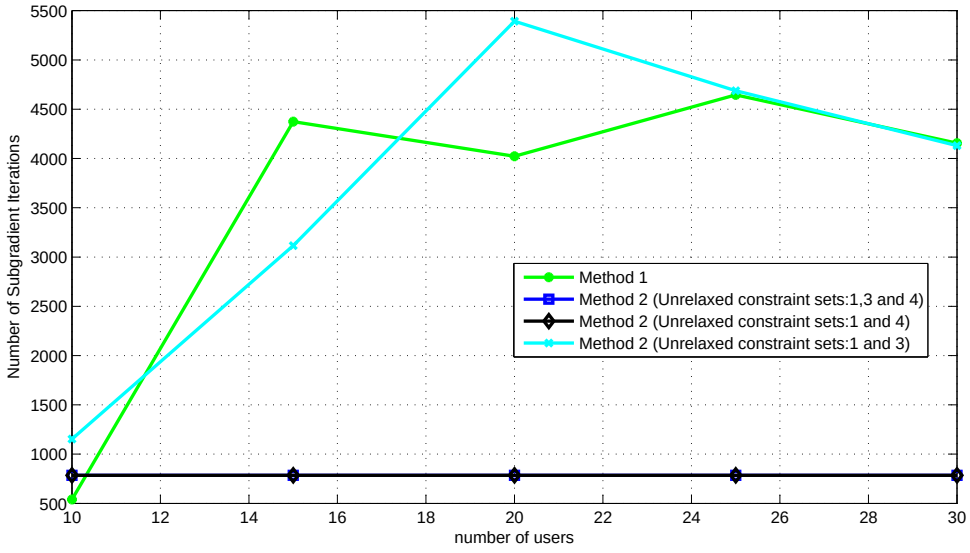
Using the observations from the previous results of smaller problem sizes, we use **step size rule 2** for its quick and correct convergence regardless of the initial dual variable vector values. Figures 3.9a and 3.9b, show the bound values and the required number of *subgradient* iterations respectively, for dual configuration 1 (Method 1) and dual configuration 2 (Method 2) when the number of users are varied in the range 10-30 at increments of five. Method 3 was not shown since it is BnB based and has a much higher (exponential) complexity compared to the other Method 1 and Method2. In figure 3.9a, a low bound is tighter than a higher one and hence is closer to the solution.

The results for the bounds obtained are consistent with the previous results that showed the bound goodness for each method. Figure 3.9b shows that the number of *subgradient* algorithm iterations required to converge to the bound final obtained value is lowest for dual configuration 2 (Method 2) when the undualized constraints are $D4$ *and/or* $D3$ *and* $D1$. However in a single iteration, the worst case complexity for dual configuration 1 (Method 1) is lower ($O\left(MSKCT \log\left(\max_m (y_m^{max})\right)\right)$ compared to $O\left(MS^2KCT\right)$) for Method 2.

In a second experiment that we conducted, we show the value of the objective function versus the number of iterations between phases 1 and 2 in the *primal solution algorithm*. Since this is a Lagrangian relaxation problem, we set initial values for the




(a) Bound value.




(b) Number of Subgradient Iterations.

Figure 3.9: Bound values and the number of required *subgradient* iterations to obtain the bounds

dual variable vector $\mathbf{u}$ and try $\mathbf{u} = \mathbf{1}, \mathbf{u} = \{2, 2...2\}, \mathbf{u} = \{3, 3...3\}$. The reason for comparing the results for different dual variable values is that these and the initial power configurations are the only parameters whose values can differ for each *subgradient* iteration. We therefore want to make sure that the behavior is the same for different values of those parameters. For power initialization, we use $p_{m,i,c,t} = \frac{P_{PF}^{Total}}{pm \times M \times C \times S}$ where $pm$ is an integer power multiplier that we vary in the range 1-4.

The results show that for a given dual variable vector $\mathbf{u}$:

1. the value of the Lagrangian relaxation objective function monotonically increases,

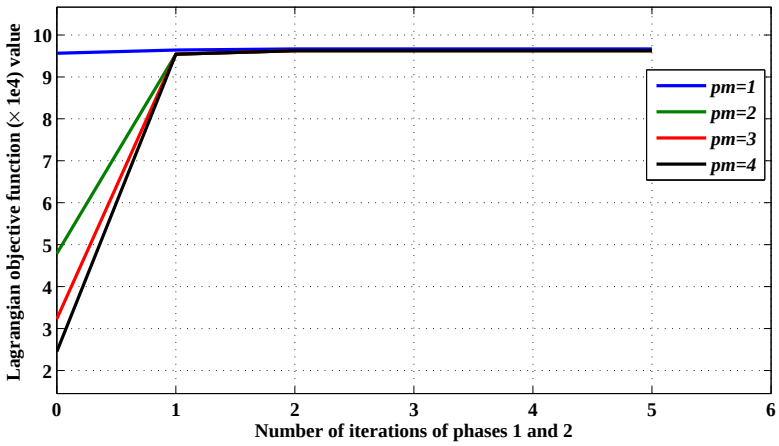
2. the solution is bounded since the value of the Lagrangian relaxation objective function stops increasing after a finite number of iterations,

3. no matter what the initial power multiplier value $pm$ is, the the procedure converges to the same solution and

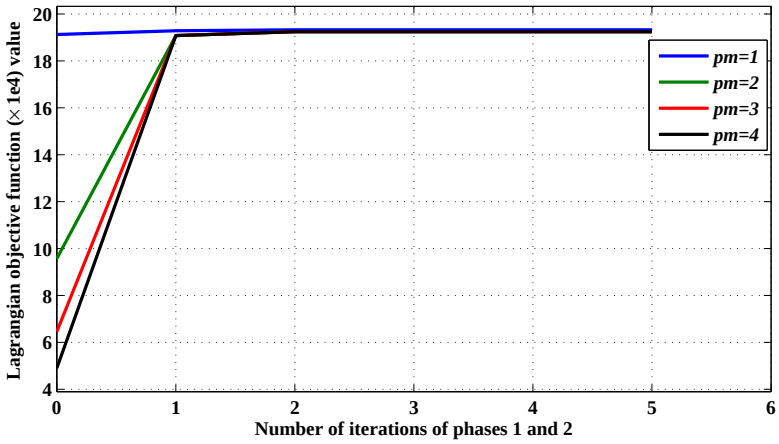
4. the number of required iterations to reach convergence in the experiments that we conducted was no more than three iterations only.

## 3.5 Chapter Conclusion

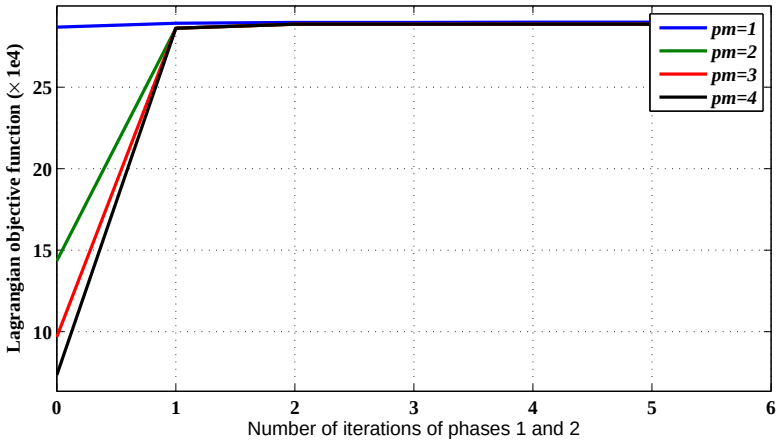
In this chapter, we presented our primary system model 'P-SysMod' for an OFDMA based HAP system for multicast transmissions in order to maximize the spectrum utilization. We formulated an optimization problem for our considered system. The optimization turned out to be an MINLP. We suggested a technique to obtain solution bounds, which could be used in BnB framework, that is based on Lagrangian relaxation, problem decompositions and the *subgradient* algorithm. We explained in details different algorithms and methods to solve for the problem's decision vectors $\mathbf{z}$




(a) $\mathbf{u} = \mathbf{1}$.




(b) $\mathbf{u} = \{2, 2, ..., 2\}$.




(c) $\mathbf{u} = \{3, 3, ..., 3\}$.

Figure 3.10: The objective function value versus iteration no. at different initial dual variable values

and **p** in the decomposed subproblems. Numerical and simulation results showed the goodnesses of the bounds. Dual configuration 2 (method 2) and dual configuration 3 (method 3) gave the best bounds while dual configuration 1 (method 1) gave acceptable bounds that are almost equal to those given by LP relaxation. Also, we used three different step size rules for the *subgradient* algorithm to find the dual decision vector **u**. Numerical experiments were conducted to compare the performance of the three step size rules and **Step size rule 2** was found to perform better in terms of convergence speed.

Although dual configuration 2 (Method 2) gave better bounds than Method 1, we showed that Method 1 can be implemented with lower complexity per *subgradient* iteration. Lower complexity is important to us due to quick radio link condition variations. Therefore, Method 1 seems a reasonable choice with acceptable bounds. Using, dual configuration 1 (Method1) for Phase 1, we were able to show how a greedy algorithm can solve for **p** in LR , a linear program, with a lower worst case complexity than simplex. We gave some results that showed that both phases 1 and 2 in the *primal solution algorithm* converge to a solution (a local solution at least) in three iterations.

In Chapter 4, the extended system model E-SysMod is provided for which we succeed to find a more efficient formulation than the one we obtained for P-SysMod.

# Chapter 4

# Multicasting in a Single HAP System: An Extended System Model and Problem Formulation

This chapter presents an extension to the system model that was proposed in Chapter 3 to a more generic and flexible situation where each user in the HAP service area can receive more than one multicast session if a user requests so. Also, each multicast group can be transmitted over more than one antenna on different frequency-time slots. Besides power, time and frequency being the resources to be allocated, antenna selection (space allocation) is also performed in the extended system model (E-SysMod). This is beneficial because it introduces more diversity in the system to support more multicast session admissions by exploiting possible spacial overlaps of the HAP antenna beam footprints. This increases the choices for multicasting ACRRA, hence giving higher possibility of more multicast-admissions. Subchannels can get assigned to antennas at different time slots to find the best possible allocation for different multicast sessions while satisfying the sessions' data capacity requirements as

well as the users' minimum SINR requirements over each subchannel for each session they receive.

The main difference between E-SysMod and the previous model in Section 3.1 (P-SysMod) is that we no longer adopt the concept of "cells" similar to terrestrial cellular systems. Instead, a multicast group could actually receive transmission on more than one antenna on different frequency-time slots simultaneously. P-SysMod did not allow that by adopting the concept of cells where a user can receive only from the antenna that illuminates the cell in which the user resides. In P-SysMod, a group of users that receive the same multicast session in different cells were considered separate groups while in E-SysMod, all users receiving the same multicast session are considered in the same group regardless of the antenna.

The second difference is that P-SysMod considered that a user can only receive multiple multicast sessions being transmitted only in the cell in which the user resides. While in E-SysMod, a user can receive multiple multicast sessions from any antenna for all the HAP service area according to the antenna-to-session assignment. Finally, E-SysMod considers different multicast session priorities for user admissions and aims to maximize the number of admitted users, with highest priorities, instead of the spectrum utilization as in P-SysMod. This means that users having higher priority to receive a particular session or equivalently, sessions of more importance to a user than others will be favored for admission.

The most important component of this chapter, from our point of view, is that a more efficient formulation for E-SysMod was derived that has a much smaller size compared to $\mathcal{HAP}_2^{Lagrange}$. We found that even with extending P-SysMod to E-SysMod, the obtained formulation in this chapter is much efficient than the formulation $\mathcal{HAP}_2^{Lagrange}$. This is an important achievement that we were able to make in

a later stage of our research. It is not just about the extension of the system model, since if we reduce the system model from E-SysMod to P-SysMod, the provided formulation in this chapter is still far more efficient than $\mathcal{HAP}_2^{Lagrange}$. Therefore, we can say that there are two contributions that we combine together in this chapter:

1. An efficient formulation and

2. an extension of P-SysMod to the more general and flexible E-SysMod.

Formulating the problem using much smaller number of variables and constraints is an important step to reduce the computational effort and memory requirements by the HAP computing hardware on-board. This is especially crucial when the radio channel gains are quick due to small scale fading, causing fast changes in the optimization problem coefficients $g_{i,k,c,t}$. Rapid changes in the problem instances require the ability of solving the problem before the instance changes.

The rest of this chapter is organized as follows. A detailed explanation of the extended system model, E-SysMod, is provided in Section 4.1. Section 4.2 gives the formulation $\mathcal{HAP}^{Eff}$ for E-SysMod as well as the interpretation for the constraints and the decision variables. Using reformulation techniques, the formulation $\mathcal{HAP}^{Eff}$ gets reduced to a *mixed binary polynomial constrained program* (MBPCP) in Section 4.3. In Section 4.4, the problem gets reduced to a mixed binary quadratically constrained program (MBQCP) $\mathcal{HAP}^{Eff}_{MBQCP}$. Section 4.5 provides the necessary analysis to prove that the quadratic constraints are non-convex. Finally, Section 4.6 provides a comparison between the sizes of $\mathcal{HAP}^{Eff}_{MBQCP}$ and $\mathcal{HAP}_2^{Lagrange}$.

## 4.1 Extended System Model

In this section, the extended system model, E-SysMod, for joint admission control-radio resource management (AC-RRA) for multicasting in a HAP service area is presented. The set of users that get admitted to receive a multicast session $m$ are considered a multicast group with the same index of the session, $m$. The HAP has multiple antennas over which the multicast streams are transmitted to the service area. A user can request to receive more than one session and hence may be admitted to (allowed to receive) one or more of the requested sessions. This means that after the admission is done, a user can belong to more than one multicast group. Any two multicast sessions may not be transmitted on the same resource trio combination $(i, c, t)$ to avoid inseparable signal interference, where $i$ is the antenna index, $c$ is the subchannel index and $t$ is the time slot index. For a frequency-time slot $(c, t)$ to be assigned to a particular user to receive session $m$ on antenna $i$, it has to satisfy a minimum SINR threshold $\gamma_{m,i}^{th}$ to guarantee an acceptable bit-error-rate performance. $\gamma_{m,i}^{th}$ could be different across the sessions and antennas depending on the possibly different modulation and channel coding schemes.

Figure 4.1 shows the power $p_{m,i,c,t}$ for session $m$ being assigned to the trio $(i, c, t)$. The antenna, frequency and time resources are represented graphically by three dimensions where the antenna dimension is not necessarily orthogonal to the frequency-time plane due to the possibility of antenna foot print overlaps. Orthogonality here means the absence of interference between any pair of trios $(i, c, t)$ represented by the small cubes in the figure. HAP power is allocated to each of the trio cubes for the different multicast sessions being transmitted to the service area. The "cubes" are assigned to the different multicast groups and the users in the HAP service area are assigned to these groups according to their priority value $\rho_{m,k}$ ,QoS requirements

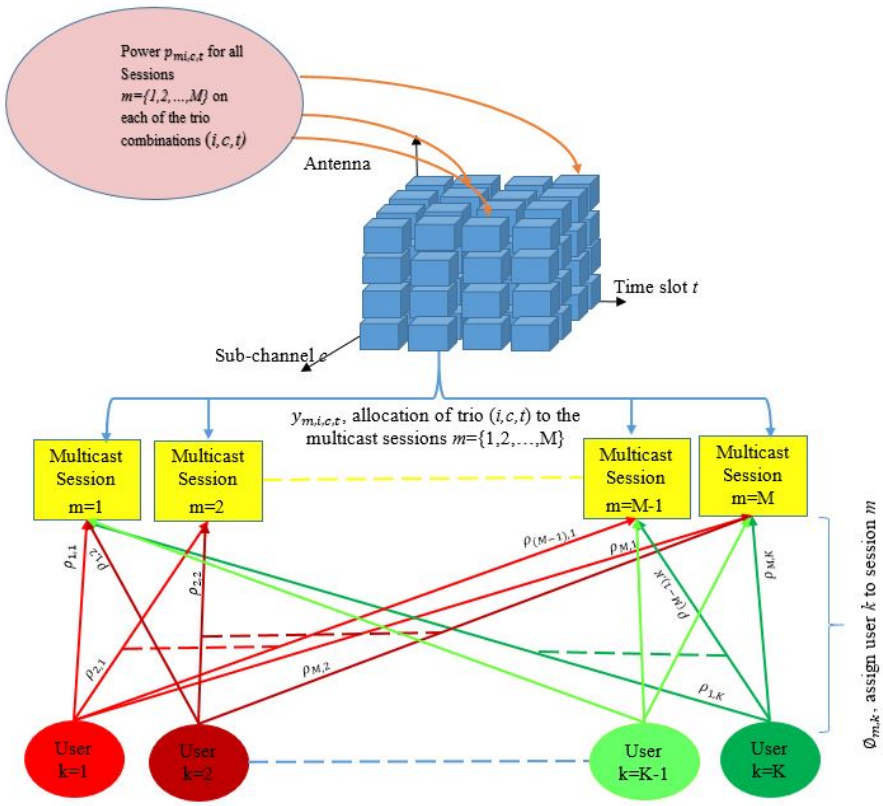



Figure 4.1: Illustration of the multicasting AC-RRA in E-SysMod

and availability of resources. The concept of "cells", similar to terrestrial cellular communications is not necessary here since a user can receive from multiple antennas as long as the SINR is above the required threshold. This is especially useful when many antenna spot beams overlap together in the service area like in Figure 4.2.

For E-SysMod, there are two definitions associated with a group's data capacity. The minimum capacity of the group is defined as :

$$R_m^{min} = \sum_{i=1}^{S} \sum_{c=1}^{C} \sum_{t=1}^{T} r_{m,i,c,t}^{min}, \tag{4.1}$$

where $r_{m,i,c,t}^{min}$ is the capacity of session $m$ over the trio $(i, c, t)$ for the user with the

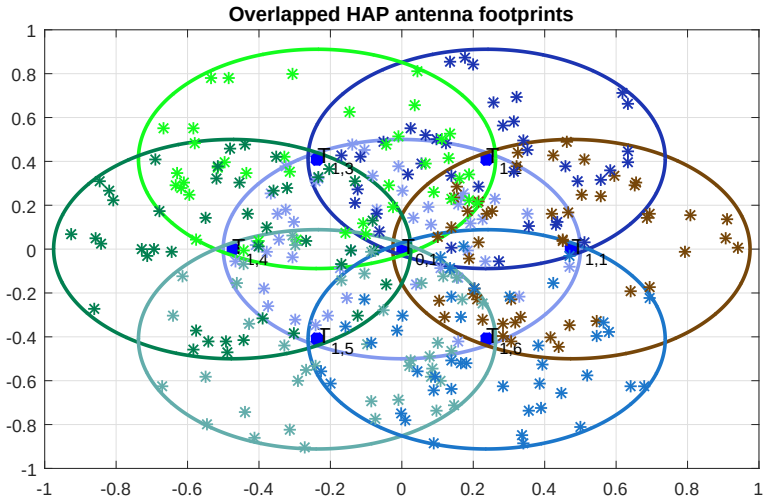



Figure 4.2: Illustration of the HAP antenna beam overlaps

minimum SINR on $(i, c, t)$ and is given as :

$$r^{min}_{m,i,c,t} = \frac{\Delta B \Delta T}{F} log\left(1 + \min_{k}\, x_{m,i,k,c,t}\right), \tag{4.2}$$

where $\Delta B$ is the subchannel bandwidth, $\Delta T$ is the time slot duration, $F$ is the frame length duration and $x_{m,i,k,c,t}$ either:

- takes the value of the SINR of the user $k$ on the trio combination $(i, c, t)$ if the user gets to receive session $m$,
- takes a very large number $\hat{M}$ (theoretically infinity) if user $k$ does not get to receive session $m$ but some other users do, or
- zero if no users in the service area are assigned to receive session $m$.

hence $x_{m,i,k,c,t}$ can be expressed as

$$x_{m,i,k,c,t} = \frac{p_{m,i,c,t}\left[g_{i,k,c,t} + (1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}. \tag{4.3}$$

For E-SysMod, we introduce the maximum capacity of a multicast group, which we define as:

$$R_m^{max} = \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} r_{m,i,c,t}^{max}, \tag{4.4}$$

where $r_{m,i,c,t}^{max}$ is the data capacity of session $m$ over the trio combination $(i,c,t)$ which is defined to be the data capacity of the user with maximum SINR on $(i,c,t)$ and is given as:

$$r_{m,i,c,t}^{max} = \frac{\Delta B \Delta T}{F} log\left(1 + \underset{k}{max}\ t_{m,i,k,c,t}\right), \tag{4.5}$$

where $t_{m,i,k,c,t}$ either:

- takes the value of the SINR of user $k$ on the trio combination $(i,c,t)$ if the user gets to receive session $m$, or
- is zero if user $k$ does not get to receive session $m$.

hence $t_{m,i,k,c,t}$ can be expressed as:

$$t_{m,i,k,c,t} = \frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M}\sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}. \tag{4.6}$$

## 4.2 Formulation of E-SysMod

This section illustrates a more efficient formulation that takes into account the extended system model E-SysMod discussed in Section 4.1. The number of variables and functional constraints in the new formulation are greatly reduced which we believe to be a good achievement, especially that this was achieved for an extended

Table 4.1: Notation Definitions for E-SysMod

| Notation | Definition |
|---|---|
| $M$ | is the number of multicast sessions in the HAP service area. |
| $S$ | is the number of HAP antennas onboard. |
| $K$ | number of users in the service area. |
| $C$ | is the number of available subchannels. |
| $T$ | total number of time slots available over OFDMA frame. |
| $\Delta B$ | is the subchannel bandwidth. |
| $\Delta T$ | is one slot duration. |
| $F$ | is the OFDMA frame duration. |
| $\sigma^2$ | is the additive white Gaussian noise power per subchannel. |
| $p_{m,i,c,t}$ | is the value of the power assigned for multicast session $m$ on antenna $i$ in the frequency-time slot $(c,t)$. |
| $\lambda_{m,k}$ | is a binary constant that indicates whether user $k$ requests to join session $m$. |
| $\phi_{m,k}$ | is a binary variable that indicates whether a user $k$ gets assigned to receive multicast session $m$. |
| $\rho_{m,k}$ | is a constant in $\mathcal{HAP}^{Eff}_{MBQCP}$ that represents priority for user $k$ on session $m$, and is a positive integer. |
| $\theta_m$ | is a binary variable that indicates whether session $m$ receives any resources, or equivalently, whether any user gets assigned to receive the session's transmission. |
| $y_{m,i,c,t}$ | is a binary variable that indicates whether the trio combination $(i,c,t)$ is assigned for session $m$. |
| $\hat{M}$ | is a very large arbitrary number. |
| $\gamma^{th}_{m,i}$ | is the SINR value that satisfies a desired target BER for session $m$ on antenna $i$ as different sessions transmitted on different antennas may be modulated and coded differently thus requiring different SINR thresholds. |

model. The notation for the new formulation is given in Table 4.1 which still has some similarities with the notation in Chapter 3. Those notations given in blue font color are either new or have slightly different interpretations. The notations given in black font in Table 4.1 have the same definitions as those in Chapter 3.

Using the newly defined variables $\phi_{m,k}$, $\theta_m$ and $y_{m,i,c,t}$, the E-SysMod problem's formulation takes into account:

- the same QoS, resource and multicast transmission requirements as in the P-SysMod,
- as well as the differences in the extended system model explained earlier in this chapter.

The key thing that enabled us to obtain a smaller formulation, is replacing the variable $z_{m,i,k,c,t}$ in formulation $\mathcal{HAP}_2^{Lagrange}$ with the two variables $\phi_{m,k}$ and $y_{m,i,c,t}$. The formulation is given below, and an interpretation for each constraint is provided right after:

$$\max_{\phi_{m,k},\theta_m,y_{m,i,c,t},p_{m,i,c,t}} \quad \sum_{m=1}^{M}\sum_{k=1}^{K}\rho_{m,k}\phi_{m,k} \qquad (\mathcal{HAP}^{Eff})$$

s.t.

$$C1: \phi_{m,k} \leq \lambda_{m,k}, \quad \forall\ m,k$$

$$C2: \sum_{m=1}^{M} y_{m,i,c,t} \leq 1, \quad \forall i,c,t$$

$$C3: \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \geq \phi_{m,k}, \quad \forall m,k$$

$$C4: y_{m,i,c,t} \leq \sum_{k=1}^{K}\phi_{m,k}, \quad \forall\ m,i,c,t$$

$$C5: P_{PF}^{Total} y_{m,i,c,t} \geq p_{m,i,c,t}, \quad \forall m,i,c,t$$

$$C6: \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{i=1}^{C} p_{m,i,c,t} \leq P_{PF}^{Total}, \quad \forall t$$

$$C7: p_{m,i,c,t} \geq 0, \ \forall\ m,i,c,t$$

$$C8: \frac{g_{i,k,c,t}p_{m,i,c,t} + (1-\phi_{m,k})\,\hat{M}}{\sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2} \geq\ y_{m,i,c,t}\gamma_{m,i}^{th}, \quad \forall m,i,k,c,t$$

$$C9: \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \leq SCT\theta_m, \quad \forall\ m$$

$$C10: \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \geq \theta_m, \quad \forall\ m$$

$$C11: \theta_m R_m^{min} \leq \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{s=1}^{S} \frac{\Delta B \Delta T}{F} log\left(1 + \min_k\ x_{m,i,k,c,t}\right), \quad \forall m$$

$$C12: \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{s=1}^{S} \frac{\Delta B \Delta T}{F} log\left(1 + \max_k\ t_{m,i,k,c,t}\right) \geq R_m^{max}, \quad \forall m$$

The interpretation of the objective function and constraints in $\mathcal{HAP}^{Eff}$ is as follows:

- The objective function represents a weighted sum of all admissions of different users over all sessions. The larger weights force the users of higher priorities across all multicast sessions to be admitted rather than low priority users. Also, a user would be admitted to the session that has the highest priority for the user, given the constraints can be satisfied. This is different from the objective function in $\mathcal{HAP}_2^{Lagrange}$ which sums all the users in all the frequency-time slots across all HAP cells.

- $C1$ ensures that if user $k$ does not request to receive session $m$ (i.e. $\lambda_{m,k} = 0$) then the user can never be assigned to receive it (i.e. $\phi_{m,k}$ is set to zero). This constraint set is somehow similar to $D1$ in $\mathcal{HAP}_2^{Lagrange}$, yet consists of $M.K$ constraints versus $M.S.K.C.T$ in $D1$ for $\mathcal{HAP}_2^{Lagrange}$. The functional difference is that $D1$ for $\mathcal{HAP}_2^{Lagrange}$ ensures that the user can be admitted to receive session $m$ when:

  - user $k$ is in cell $i$, and
  - session $m$ is being transmitted in cell $i$.

In E-SysMod, we do not have these two restrictions.

- $C2$ ensures that a given trio combination $(i, c, t)$ can at most be assigned to one multicast group (session). This is equivalent to $D5$ in $\mathcal{HAP}_2^{Lagrange}$, yet consists of a much smaller number of constraints as will be shown later in Section 4.6.

- $C3$ ensures that user $k$ can be assigned to multicast group $m$ only when the session gets assigned at least one resource trio combination $(i, c, t)$. This constraint set, besides $C4$, are both required in $\mathcal{HAP}^{Eff}$ to connect the two sets of variables $\phi_{m,k}$ and $y_{m,i,c,t}$. These were not required in $\mathcal{HAP}_2^{Lagrange}$ since $z_{m,i,k,c,t}$ captured them both in a single variable.

- $C4$ ensures that if no users are assigned to session $m$, then no resource trios $(i, c, t)$ should be allocated to the group.

- $C5$ ensures that if the trio combination $(i, c, t)$ is not assigned for session $m$ then the power level assigned for group $m$ on $(i, c, t)$ should be forced to zero. This is equivalent to constraint set $D10$ in $\mathcal{HAP}_2^{Lagrange}$. However, each constraint in $C5$ of $\mathcal{HAP}^{Eff}$ has only two variables compared to $K + 1$ variables in each constraint in $D10$ of $\mathcal{HAP}_2^{Lagrange}$.

- $C6$ ensures that the total power at a given time slot assigned for all multicast groups on any antenna-frequency $(i, c)$ pairs, must be limited to the total available HAP power. This is exactly the same constraint as $D9$ in $\mathcal{HAP}_2^{Lagrange}$.

- $C7$ ensures that the power values $p_{m,i,c,t}$ are all non-negative. Which is exactly the same as $D11$ in $\mathcal{HAP}_2^{Lagrange}$.

- $C8$ is a constraint set that enforces the SINR for user $k$ receiving session $m$ to be greater than a threshold value $\gamma_{m,i}^{th}$ to admit the user to group $m$. There are

three possibilities for each of the constraints in the set, which are explained as follows:

1. If the trio combination $(i, c, t)$ is not assigned to session $m$ (i.e. $y_{m,i,c,t} = 0$), constraint $C5$ forces the power variable $p_{m,i,c,t}$ to be zero. This makes the left hand side (L.H.S) in constraint ($C8$) either equal to the very large number $\hat{M}$, or equal to zero, depending on the value of $\phi_{m,k}$. Both cases satisfy the inequality rendering the constraint redundant.

2. If the trio $(i, c, t)$ is assigned to session $m$ (i.e. $y_{m,i,c,t} = 1$), but user $k$ is not assigned to receive $m$ (i.e. $\phi_{m,k} = 0$), the power variable $p_{m,i,c,t}$ could take any non-zero value. In this case, the term in the numerator of the R.H.S becomes greater than or equal to the very large number $\hat{M}$ making the constraint redundant.

3. For $y_{m,i,c,t} = 1$, if user $k$ is to get admitted for session $m$, then $\phi_{m,k} = 1$. In this case, the term on the L.H.S of the constraint equivalent to the SINR of session $m$ for user $k$ since the numerator becomes the product of the power variable $p_{m,i,c,t}$ times the gain of the user on the trio combination $(i, c, t)$ . The R.H.S. also becomes equal to the acceptable threshold value, $\gamma_{m,i}^{th}$, for session $m$ on antenna $i$. In this case the SINR constraint over the trio combination $(i, c, t)$ comes into effect for user $k$ and session $m$.

Constraint set $C8$ in $\mathcal{HAP}^{Eff}$ is functionally equivalent to $D3$ in $\mathcal{HAP}_2^{Lagrange}$.

- $C9$ and $C10$ together ensure that only if there are any resources being assigned for session $m$, then this must set the variable $\theta_m = 1$, otherwise $\theta_m = 0$ is enforced. This is needed for the minimum data capacity constraint $C12$. Constraint sets $C9$ and $C10$ have no equivalent constraint sets in $\mathcal{HAP}_2^{Lagrange}$.

- $C11$ ensures the minimum capacity $R_m^{min}$ of a multicast session is satisfied. We use the definition of the minimum capacity of a multicast group given in Equation (4.1). There are four possibilities for $x_{m,i,k,c,t}$ (defined by Equation 4.3) which are explained as follows:

  1. $y_{m,i,c,t} = 0$ *and* $\phi_{m,k} = 0$. In this case, constraint $C5$ will force the power variable $p_{m,i,c,t}$ to be zero which results in, $x_{m,i,k,c,t} = 0$ and $\underset{k}{min}\ x_{m,i,k,c,t} = 0$ giving a capacity of zero on the trio combination $(i, c, t)$.

  2. $y_{m,i,c,t} = 0$ *and* $\phi_{m,k} = 1$. This would have exactly the same result as the first case, a capacity of zero on that trio combination $(i, c, t)$ for the same reasons.

  3. $y_{m,i,c,t} = 1$ *and* $\phi_{m,k} = 0$. In this case $x_{m,i,k,c,t} = \infty$ theoretically, where infinity is practically considered any number greater than or equal to $\hat{M}$, which ensures that for that particular user, its SINR value is never returned by the term $\underset{k}{min}\ \ x_{m,i,k,c,t}$. There are definitely other users who have $\phi_{m,k} = 1$, according to constraint $C4$, from which the least SINR on $(i, c, t)$ is returned by $\underset{k}{min}\ \ x_{m,i,k,c,t}$.

  4. $y_{m,i,c,t} = 1$ *and* $\phi_{m,k} = 1$ in this case $x_{m,i,k,c,t} = \frac{p_{m,i,c,t} g_{i,k,c,t}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}$ which is the SINR of the user $k$ and session $m$ over the trio combination $(i, c, t)$. Therefore, $\underset{k}{min}\ x_{m,i,k,c,t}$ would return the minimum SINRs of all users in group $m$ over $(i, c, t)$.

  The variable $\theta_m$ ensures that the constraint is not in effect in the case that no resources are allocated at all for session $m$ , i.e. $\theta_m = 0$. This constraint set extends the lower bound constraint set for $C4$ in $\mathcal{HAP}^{Init}$ by summing the data capacity of session $m$ over all the HAP antennas. It is worth noting that

for P-SysMod, constraint set $D2$ in $\mathcal{HAP}_2^{Lagrange}$ enforced all users to receive multicast sessions from only one antenna, which is the antenna that covers the cell they reside in.

- $C12$ ensures that the maximum capacity of the group or session $m$, defined by Equation (4.4), is satisfied. The possibilities for $t_{m,i,k,c,t}$, defined by Equation 4.6, are explained as follows:

  1. For the case $y_{m,i,c,t} = 0$, no matter what the value of $\phi_{m,k}$ is, the power variable $p_{m,i,c,t}$ is forced to zero by constraint $C5$, therefore we get $t_{m,i,k,c,t} = 0 \; \forall \; k$, and $\underset{k}{max} \; t_{m,i,k,c,t} = 0$.

  2. For the case $y_{m,i,c,t} = 1$ and user $k$ is not assigned to group $m$, i.e. $\phi_{m,k} = 0$, $t_{m,i,k,c,t}$ returns zero but the term $\underset{k}{max} \; t_{m,i,k,c,t}$ returns the highest SINR, over $(i, c, t)$, amongst all users assigned to session/group $m$. We are sure that if $y_{m,i,c,t} = 1$ then there is at least one user who has $\phi_{m,k} = 1$ according to constraint set $C5$.

  3. For the case $y_{m,i,c,t} = 1$, and user $k$ assigned to the group $m$, i.e. $\phi_{m,k} = 1$, $t_{m,i,k,c,t} = \frac{p_{m,i,c,t} g_{i,k,c,t}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}$ and the term $\underset{k}{max} \; t_{m,i,k,c,t}$ returns the highest SINR over $(i, c, t)$ amongst all users assigned to session/group $m$.

Constraint set $C12$ in $\mathcal{HAP}^{Eff}$ is different from its corresponding upper bound data capacity constraint set in $C4$ $\mathcal{HAP}^{Init}$ in two aspects. The first aspect is that in $C12$ in $\mathcal{HAP}^{Eff}$ utilizes the newly introduced concept of maximum multicast group data capacity mentioned earlier in this chapter and given by Equations 4.4, 4.5 and 4.6. In this way, it is guaranteed that no user in any multicast group can have a data capacity greater than $R_m^{max}$. Constraint set $C4$ in $\mathcal{HAP}^{Init}$ on the other hand uses the data capacity of the user with

the poorest channel conditions to define the group's data capacity, and it is that data capacity that is enforced to be no more than $R_m^{max}$. This could lead to users with good channel and interference conditions in a group receiving a capacity greater than $R_m^{max}$, which constraint set $C12$ in $\mathcal{HAP}^{Eff}$ makes sure does not happen. The second difference is that since E-SysMod allows the users in a group $m$ to receive the multicast transmission on more than one antenna simultaneously, then the maximum data capacity of the group is obtained by summing all the group's data capacities over all the antennas. This was not considered in $\mathcal{HAP}^{Init}$ and its enhanced reformulated version $\mathcal{HAP}_2^{Lagrange}$.

It is worth mentioning that the SINR constraint set $C8$ ensures that for a given multicast session $m$, no more than one antenna can be used to transmit the session over the same frequency-time slot $(c,t)$. This is possible since in the L.H.S. of the constraint set, the interference terms in the denominator include received copies of the same desired session $m$ from the other antennas of the HAP from which the user is not meant to receive in the frequency-time slot $(c,t)$. The entire constraint set $C8$ guarantees if the SINR requirement is satisfied by receiving a session on one antenna in slot $(c,t)$, then this could not be possible simultaneously over any other antenna for slot $(c,t)$ given the assumption $\gamma_{m,i}^{th} \geq 1$.

As we can see, the problem formulation labeled $\mathcal{HAP}^{Eff}$ is a binary mixed nonlinear constrained problem. Constraint set $C8$ has a special structure of being a mixed binary quadratic constraint set that consists only of bilinear terms. Constraint sets $C11$ and $C12$ are non linear mixed binary constraints with *min* and *max* terms respectively that complicate them further. In Section 4.3 reformulation techniques are used to eliminate the *min-max* terms and replace those constraints with multivariate polynomial constraints. Then we show how the polynomial constraints are reduced

to multivariate quadratic constraints that consist only of bilinear terms in Section 4.4.

## 4.3 Reducing the Formulation $\mathcal{HAP}^{Eff}$ to a Mixed Binary Polynomial Constrained Problem

In this section we show how the constraint sets $C11$ and $C12$ in $\mathcal{HAP}^{Eff}$ are replaced by mixed binary polynomial constraints ($MBPCs$), some of which are quadratic.

For constraint set $C11$ in $\mathcal{HAP}^{Eff}$, the constraint can be rewritten in the form:

$$log\left[\prod_{i=1}^{S}\prod_{c=1}^{C}\prod_{t=1}^{T}\left(1+\min_{k}\frac{p_{m,i,c,t}\left[g_{i,k,c,t}+(1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i'\neq i}g_{i',k,c,t}p_{m,i',c,t}+\sigma^2}\right)\right]\geq\frac{\theta_m R_m^{min}F}{\Delta B\Delta T},\quad\forall\ m. \tag{4.7}$$

Taking exponential of 2 for both sides of the constraint, we get:

$$\prod_{i=1}^{S}\prod_{c=1}^{C}\prod_{t=1}^{T}\left(1+\min_{k}\frac{p_{m,i,c,t}\left[g_{i,k,c,t}+(1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i'\neq i}g_{i',k,c,t}p_{m,i',c,t}+\sigma^2}\right)\geq 2^{\frac{\theta_m R_m^{min}F}{\Delta B\Delta T}},\quad\forall\ m. \tag{4.8}$$

The right hand side of the constraint can be rewritten to give the constraint:

$$\prod_{i=1}^{S}\prod_{c=1}^{C}\prod_{t=1}^{T}\left(1+\min_{k}\frac{p_{m,i,c,t}\left[g_{i,k,c,t}+(1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i'\neq i}g_{i',k,c,t}p_{m,i',c,t}+\sigma^2}\right)\geq\hat{R}_m^{min}\theta_m+(1-\theta_m),\quad\forall\ m, \tag{4.9}$$

where $\hat{R}_m^{min}=2^{\frac{R_m^{min}F}{\Delta B\Delta T}}$ . Then we introduce the auxiliary variables $w_{m,i,c,t}$ for the terms

$$\left(1+\min_{k}\frac{p_{m,i,c,t}\left[g_{i,k,c,t}+(1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i'\neq i}g_{i',k,c,t}p_{m,i',c,t}+\sigma^2}\right),$$

which give the following set of equations:

$$w_{m,i,c,t} = \min_{k} \left( \frac{p_{m,i,c,t} \left[ g_{i,k,c,t} + (1 - \phi_{m,k}) \hat{M} \right]}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \right) + 1, \quad \forall\ m, i, c, t. \tag{4.10}$$

and the following inequality set becomes valid:

$$\frac{p_{m,i,c,t} \left[ g_{i,k,c,t} + (1 - \phi_{m,k}) \hat{M} \right]}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \geq w_{m,i,c,t} - 1, \quad \forall\ m, i, k, c, t. \tag{4.11}$$

Therefore constraint set $C11$ can be replaced by:

$$\prod_{i=1}^{S} \prod_{c=1}^{C} \prod_{t=1}^{T} w_{m,i,c,t} \geq \hat{R}_m^{min} \theta_m + (1 - \theta_m), \quad \forall\ m, \tag{4.12}$$

and

$$\frac{p_{m,i,c,t} \left[ g_{i,k,c,t} + (1 - \phi_{m,k}) \hat{M} \right]}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \geq w_{m,i,c,t} - 1, \quad \forall\ m, i, k, c, t. \tag{4.13}$$

where $w_{m,i,c,t} \geq 1$.

For $C12$, the constraint set can be rewritten in the form:

$$log \left[ \prod_{i=1}^{S} \prod_{c=1}^{C} \prod_{t=1}^{T} \left( 1 + \max_{k} \frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \right) \right] \leq \frac{R_m^{max} F}{\Delta B \Delta T}, \ \forall\ m, \tag{4.14}$$

taking the exponent of 2 for both sides we get:

$$\prod_{i=1}^{S} \prod_{c=1}^{C} \prod_{t=1}^{T} \left( 1 + \max_{k} \frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \right) \leq \hat{R}_m^{max}, \ \forall\ m, \tag{4.15}$$

where $\hat{R}_m^{max} = 2^{\frac{R_m^{max}F}{\Delta B \Delta T}}$. Then we introduce the auxiliary variables $u_{m,i,c,t}$ for the terms

$$\left(1 + \underset{k}{max} \frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}\right),$$

which gives the following set of inequalities:

$$u_{m,i,c,t} = 1 + \underset{k}{max} \left(\frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2}\right), \quad \forall\ m, i, c, t, \tag{4.16}$$

and the following inequality set becomes valid:

$$\frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \leq u_{m,i,c,t} - 1, \quad \forall\ m, i, k, c, t. \tag{4.17}$$

Therefore the constraint $C12$ can be replaced by:

$$\prod_{i=1}^{S} \prod_{c=1}^{C} \prod_{t=1}^{T} u_{m,i,c,t} \leq \hat{R}_m^{max}, \ \forall\ m, \tag{4.18}$$

and

$$\frac{g_{i,k,c,t} p_{m,i,c,t} \phi_{m,k}}{\sum_{m=1}^{M} \sum_{\forall i' \neq i} g_{i',k,c,t} p_{m,i',c,t} + \sigma^2} \leq u_{m,i,c,t} - 1, \quad \forall\ m, i, k, c, t. \tag{4.19}$$

where $u_{m,i,c,t} \geq 1$. The new constraints given by (4.12), (4.13), (4.18) and (4.19) are all polynomials where the ones given by (4.13) and (4.19) are second degree polynomial (quadratic). Therefore replacing constraint sets C11 and C12 in $\mathcal{HAP}^{Eff}$ with (4.12), (4.13), (4.18) and (4.19) gives a mixed binary polynomial constraint program (MBPCP). Section 4.4 shows how this is further reduced to a mixed binary quadratically constrained program (MBQCP).

## 4.4 Reduction of the Formulation to Mixed Binary Quadratic Constraints

Any MBPCP optimization problem maybe reduced to a MBQCP by the introduction of auxiliary variables and constraints to reduce all polynomial degrees to 2. For example a cubic polynomial term $x_1x_2x_3$ could be modeled as $x_1X_{23}$ with $X_{23} = x_2x_3$. Using this simple reformulation technique, the polynomial constraints obtained in the previous section, can be converted to mixed binary quadratic constraints by replacing (4.12) by the following:

$$w_{m,(1)}W_{m,1} \geq \hat{R}_m^{min}\theta_m + (1-\theta_m)\,, \quad \forall\ m, \tag{4.20a}$$

$$W_{m,j} = w_{m,(j+1)}W_{m,j+1},\ \forall j = 1,2,...,n-3, \quad \forall\ m, \tag{4.20b}$$

$$W_{m,(n-2)} = w_{m,(n-1)}w_{m,(n)}, \quad \forall\ m, \tag{4.20c}$$

where $n = S.C.T$ and $j = (i-1)\,.C.T + (c-1)\,.T + t$ for the set of variables $W_{m,j}$ while for $w_{m,(j)}$, $j \equiv (i,c,t)$. Equality constraints can be replaced by inequality constraints to give:

$$w_{m,(1)}W_{m,1} \geq \hat{R}_m^{min}\theta_m + (1-\theta_m)\,, \quad \forall\ m, \tag{4.21a}$$

$$W_{m,j} \leq w_{m,(j+1)}W_{m,j+1},\ \forall j = 1,2,...,n-3, \quad \forall\ m, \tag{4.21b}$$

$$W_{m,j} \geq w_{m,(j+1)}W_{m,j+1},\ \forall j = 1,2,...,n-3, \quad \forall\ m, \tag{4.21c}$$

$$W_{m,n-2} \leq w_{m,(n-1)}w_{m,(n)}, \quad \forall\ m, \tag{4.21d}$$

$$W_{m,n-2} \geq w_{m,(n-1)}w_{m,(n)}, \quad \forall\ m, \tag{4.21e}$$

These sets replace the set of $M$ constraints in (4.12) with $3M + 2M.\,(S.C.T - 3)$ quadratic constraints and adds $M \times (S.C.T - 2)$ new variables $W_{m,j}$. Similarly the

constraint set in (4.18) can be replaced by:

$$u_{m,(1)}U_{m,1} \leq \hat{R}_m^{max} \quad \forall\ m, \tag{4.22a}$$

$$U_{m,j} \leq u_{m,(j+1)}U_{m,j+1}\ \forall j = 1,2,...,n-3, \quad \forall\ m, \tag{4.22b}$$

$$U_{m,j} \geq u_{m,(j+1)}U_{m,j+1}\ \forall j = 1,2,...,n-3, \quad \forall\ m, \tag{4.22c}$$

$$U_{m,n-2} \leq u_{m,(n-1)}u_{m,(n)} \quad \forall\ m, \tag{4.22d}$$

$$U_{m,n-2} \geq u_{m,(n-1)}u_{m,(n)} \quad \forall\ m. \tag{4.22e}$$

Again, this replaces the $M$ constraints in (4.18) with $3M+2M.(S.C.T-3)$ quadratic constraints and adds $M \times (S.C.T-2)$ new variables $U_{m,j}$.

The optimization problem is now an MBQCP given by:

$$\underset{\phi_{m,k},\theta_m,y_{m,i,c,t},u_{m,i,c,t},U_{m,j},w_{m,i,c,t},W_{m,j},p_{m,i,c,t}}{max} \quad \sum_{m=1}^{M}\sum_{k=1}^{K}\rho_{m,k}\phi_{m,k} \qquad (\mathcal{HAP}_{MBQCP}^{Eff})$$

s.t.

$$\overline{C1}: \phi_{m,k} \leq \lambda_{m,k}, \quad \forall\ m,k$$

$$\overline{C2}: \sum_{m=1}^{M} y_{m,i,c,t} \leq 1, \quad \forall i,c,t$$

$$\overline{C3}: \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \geq \phi_{m,k}, \quad \forall m,k$$

$$\overline{C4}: y_{m,i,c,t} \leq \sum_{k=1}^{K}\phi_{m,k}, \quad \forall\ m,i,c,t$$

$$\overline{C5}: P_{PF}^{Total}y_{m,i,c,t} \geq p_{m,i,c,t}, \quad \forall m,i,c,t$$

$$\overline{C6}: \sum_{m=1}^{M}\sum_{i=1}^{S}\sum_{c=1}^{C} p_{m,i,c,t} \leq P_{PF}^{Total}, \quad \forall t$$

$$\overline{C7} : p_{m,i,c,t} \geq 0, \ \forall\ m,i,c,t$$

$$\overline{C8} : \frac{g_{i,k,c,t}p_{m,i,c,t} + (1-\phi_{m,k})\,\hat{M}}{\sum_{m=1}^{M}\sum_{\forall i' \neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2} \geq \ y_{m,i,c,t}\gamma_{m,i}^{th}, \quad \forall m,i,k,c,t$$

$$\overline{C9} : \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \leq SCT\theta_m, \ \ \forall\ m$$

$$\overline{C10} : \sum_{i=1}^{S}\sum_{c=1}^{C}\sum_{t=1}^{T} y_{m,i,c,t} \geq \theta_m, \ \ \forall\ m$$

$$\overline{Q1a} : w_{m,(1)}W_{m,1} \geq \hat{R}_m^{min}\theta_m + (1-\theta_m), \ \ \forall\ m$$

$$\overline{Q1b} : W_{m,j} \leq w_{m,(j+1)}W_{m,j+1}\ \forall j = 1,2,...,n-3, \ \ \forall\ m$$

$$\overline{Q1c} : W_{m,j} \geq w_{m,(j+1)}W_{m,j+1}\ \forall j = 1,2,...,n-3, \ \ \forall\ m$$

$$\overline{Q1d} : W_{m,n-2} \leq w_{m,(n-1)}w_{m,(n)}, \ \ \forall\ m$$

$$\overline{Q1e} : W_{m,n-2} \geq w_{m,(n-1)}w_{m,(n)}, \ \ \forall\ m$$

$$\overline{Q2} : \frac{p_{m,i,c,t}\left[g_{i,k,c,t} + (1-\phi_{m,k})\,\hat{M}\right]}{\sum_{m=1}^{M}\sum_{\forall i' \neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2} \geq w_{m,i,c,t} - 1, \ \ \forall\ m,i,k,c,t$$

$$\overline{Q3a} : u_{m,(1)}U_{m,1} \leq \hat{R}_m^{max}, \ \ \forall\ m$$

$$\overline{Q3b} : U_{m,j} \leq u_{m,(j+1)}U_{m,j+1}\ \forall j = 1,2,...,n-3, \ \ \forall\ m$$

$$\overline{Q3c} : U_{m,j} \geq u_{m,(j+1)}U_{m,j+1}\ \forall j = 1,2,...,n-3, \ \ \forall\ m$$

$$\overline{Q3d} : U_{m,n-2} \leq u_{m,(n-1)}u_{m,(n)} \ \ \forall\ m$$

$$\overline{Q3e} : U_{m,n-2} \geq u_{m,(n-1)}u_{m,(n)} \ \ \forall\ m$$

$$\overline{Q4} : \frac{g_{i,k,c,t}p_{m,i,c,t}\phi_{m,k}}{\sum_{m=1}^{M}\sum_{\forall i' \neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2} \leq u_{m,i,c,t} - 1 \ \ \forall\ m,i,k,c,t$$

$$\phi_{m,k}, \theta_m, y_{m,i,c,t} \ \in \{0,1\} \ \forall\ m,i,k,c,t$$

$$0 \leq p_{m,i,c,t} \leq P_{PF}^{Tot}, \ 1 \leq w_{m,i,c,t} \leq \hat{R}_m^{max}, 1 \leq W_{m,j} \leq \hat{R}_m^{max}\ 1 \leq u_{m,i,c,t} \leq \hat{R}_m^{max},$$

$$0 \leq U_{m,j} \leq \hat{R}_m^{max}, \ \forall\ m,i,c,t.$$

In the next section, we mathematically analyze the convexity of the quadratic constraint sets $\overline{C8}$, $\overline{Q1a}-\overline{Q1e}$, $\overline{Q2}$, $\overline{Q3a}-\overline{Q3e}$ and $\overline{Q4}$ and show that they are all non convex. Knowing whether those constraints sets are convex is crucial in determining the solution techniques.

## 4.5 Convexity Analysis of the Quadratic Constraint Sets in $\mathcal{HAP}^{Eff}_{MBQCP}$

In quadratically constrained programming, it is well known that a constraint in the vector form:

$$\tilde{\mathbf{x}}^T\mathbf{Q}\tilde{\mathbf{x}} + \mathbf{q}\tilde{\mathbf{x}} \leq b, \tag{4.23}$$

where

- $\tilde{\mathbf{x}}$ is the column vector of decision variables,
- $\mathbf{q}$ is a row vector of constants and
- $b$ is a scalar constant,

is convex if and only if the symmetric matrix $\mathbf{Q}$, which is the Hessian matrix of the constraint [61], is positive semi-definite, i.e. $\mathbf{Q} \succeq 0$ [62].

Starting with the quadratic constraint set $\overline{C8}$, and after relaxing the binary variables $\phi_{m,k}$ and $y_{m,i,c,t}$ to be continuous variables in the ranges $0 \leq \phi_{m,k} \leq 1$ and $0 \leq y_{m,i,c,t} \leq 1$, the quadratic part of the constraint, $y_{m,i,c,t}\gamma^{th}_{m,i}\left(\sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}p_{m,i',c,t} + \sigma^2\right)$ for four terms can be viewed with the simple notation:

$$f_{\overline{C8}}(\tilde{\mathbf{x}}) = a_1\tilde{x}_1\tilde{x}_2 + a_2\tilde{x}_1\tilde{x}_3 + a_3\tilde{x}_1\tilde{x}_4, \tag{4.24}$$

where $a_j > 0,\ \forall\ j$ since the product of the constants $\gamma_{m,i}^{th} g_{i',k,c,t}$ is known to be strictly positive. The small number of terms here is chosen arbitrarily for brevity purposes to illustrate the convexity analysis while still preserving generality for any number of terms.

The Hessian matrix $\nabla^2 f_{\overline{Q8}}(\tilde{\mathbf{x}})$ is a symmetric matrix which consists of the elements:

$$\left[\nabla^2 f_{\overline{C8}}(\tilde{\mathbf{x}})\right]_{ij} = \frac{\partial^2 f_{\overline{Q8}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_i \partial \tilde{x}_j}. \tag{4.25}$$

Evaluation of those elements given by Equation (4.25), gives the Hessian matrix for the quadratic component of $\overline{Q8}$ is:

$$\nabla^2 f_{\overline{C8}}(\tilde{\mathbf{x}}) = \frac{1}{2}\begin{bmatrix} 0 & a_1 & a_2 & a_3 \\ a_1 & 0 & 0 & 0 \\ a_2 & 0 & 0 & 0 \\ a_3 & 0 & 0 & 0 \end{bmatrix}. \tag{4.26}$$

Now if any of the principal minors is negative then the matrix is neither positive definite nor positive semi-definite which are necessary and sufficient conditions for strict convexity or non-strict convexity respectively [63]. We start checking the *leading* principal minors [63] , if any of those is negative we stop checking the rest and we conclude that the function is non convex and hence the entire constraint is non-convex. The $n^{th}$ leading principal minor is defined as the determinant of the matrix whose elements are the first $n \times n$ elements in the Hessian matrix given in (4.26) and we refer to it here as $\left|H_n^{\overline{C8}}\right|$. It can very easily be seen that $\left|H_1^{\overline{C8}}\right| = 0$ and $\left|H_1^{\overline{C8}}\right| = -\frac{1}{2}a_1^2$. We stop after obtaining the second leading principal minor which shows that constraint set $\overline{C8}$ is non-convex.

For constraint set $Q2$, the quadratic constraint component can be simply written in the form:

$$f_{\overline{Q2}} = a_1\tilde{x}_1\tilde{x}_2 + a_2\tilde{x}_1\tilde{x}_3 + a_3\tilde{x}_1\tilde{x}_4 + a_4\tilde{x}_5\tilde{x}_6, \tag{4.27}$$

which yields the Hessian matrix

$$\nabla^2 f_{\overline{Q2}}(\tilde{\mathbf{x}}) = \frac{1}{2}\begin{bmatrix} 0 & a_1 & a_2 & a_3 & 0 & 0 \\ a_1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & 0 & 0 & 0 & 0 & 0 \\ a_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 0 & a_4 & a_4 \end{bmatrix}, \tag{4.28}$$

where the $a_j > 0,\ \forall\ j$. Testing the leading principal minors, we find that $\left|H_1^{\overline{Q2}}\right| = 0$ and $\left|H_1^{\overline{Q2}}\right| = -\frac{1}{2}a_1^2$. We stop after obtaining the second leading principal minor which shows that constraint set $\overline{Q2}$ is non-convex.

For constraint set $Q4$, the quadratic constraint component can be simply written in the form:

$$f_{\overline{Q2}} = a_1\tilde{x}_1\tilde{x}_2 - a_2\tilde{x}_3\tilde{x}_4 - a_3\tilde{x}_3\tilde{x}_5 - a_4\tilde{x}_3\tilde{x}_6, \tag{4.29}$$

which yields the Hessian matrix

$$\nabla^2 f_{\overline{Q4}}(\tilde{\mathbf{x}}) = \frac{1}{2}\begin{bmatrix} 0 & a_1 & 0 & 0 & 0 & 0 \\ a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_2 & 0 & 0 \\ 0 & 0 & -a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.30}$$

where the $a_j > 0,\ \forall\ j$. Testing the leading principal minors, we find that $\left|H_1^{\overline{Q4}}\right| = 0$ and $\left|H_1^{\overline{Q4}}\right| = -\frac{1}{2}a_1^2$. We stop after obtaining the second leading principal minor which shows that constraint set $\overline{Q4}$ is non-convex.

For constraint set $\overline{Q1a}$, the constraint function can be written in the following simple notation,

$$f_{\overline{Q1a}} = -\tilde{x}\tilde{y} + a\tilde{z} - b, \tag{4.31}$$

which yields a Hessian matrix

$$\nabla^2 f_{\overline{Q1a}}(\tilde{\mathbf{x}}) = \frac{1}{2}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.32}$$

Testing the leading principal minors, we find that $\left|H_1^{\overline{Q1a}}\right| = 0$ and $\left|H_1^{\overline{Q1a}}\right| = -\frac{1}{2}$. We stop after obtaining the second leading principal minor which shows that constraint set $\overline{Q1a}$ is non-convex.

Constraint set $\overline{Q1b}$ takes the same structure as $\overline{Q1a}$ and hence its constraints are also non-convex constraints. Constraint set $\overline{Q1c}$ take the negated form of $\overline{Q1a}$

yielding the Hessian matrix:

$$\nabla^2 f_{\overline{Q1c}}(\tilde{\mathbf{x}}) = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.33}$$

Testing the leading principal minors, we find that $\left|H_1^{\overline{Q1c}}\right| = 0$ and $\left|H_1^{\overline{Q1c}}\right| = -\frac{1}{2}$. We stop after obtaining the second leading principal minor which shows that constraint set $\overline{Q1c}$ is non-convex. The structure of constraint set $\overline{Q1d}$ is exactly similar as that of constraint set $\overline{Q1b}$ and the structure of $\overline{Q1e}$ matches that of $\overline{Q1c}$. Therefore $\overline{Q1d}$ and $\overline{Q1e}$ are non-convex constraint sets.

For constraint set $\overline{Q3a}$, the Hessian Matrix can be simply found to be,

$$\nabla^2 f_{\overline{Q3a}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{4.34}$$

whose second leading principal is $-1$ yielding the constraint set non-convex. The rest of the constraint sets $\overline{Q3b}$ to $\overline{Q3e}$ have structures equivalent to $\overline{Q1b}$ to $\overline{Q1e}$ hence are non-convex.

All the quadratic constraints in the formulation $\mathcal{HAP}_{MBQCP}^{Eff}$ were shown to be non-convex in this section. In Section 4.6, we compare the sizes of formulations $\mathcal{HAP}_{MBQCP}^{Eff}$ and $\mathcal{HAP}_{2}^{Lagrange}$.

## 4.6 Comparison of Problem Sizes

In this section we illustrate the differences in the sizes of the formulations $\mathcal{HAP}_{MBQCP}^{Eff}$ and $\mathcal{HAP}_{2}^{Lagrange}$. Considering $\mathcal{HAP}_{2}^{Lagrange}$ first, we see that the number of variables

are as follows:

- The number of binary variables, $z_{m,i,k,c,t}$, is the product $MSKCT$ and
- the number of continuous variables, $p_{m,i,c,t}$, is $MSCT$,
- hence giving a total number of variables

$$VN_{\mathcal{HAP}_2^{Lagrange}} = MSKCT + MSCT. \tag{4.35}$$

The number of constraints (excluding bounds and binary constraints) in each constraint set for $\mathcal{HAP}_2^{Lagrange}$ are as follows:

- constraint set $D1$ comprises $MSKCT$ constraints,
- constraint set $D2$ comprises $MSKCT\,[CT-1]\,[K-1]$ constraints,
- constraint set $D3$ comprises $MSKCT$ constraints,
- constraint set $D4$ comprises $MSKCT$ constraints,
- constraint set $D5$ comprises $MSKCT\,[M-1]\,[K-1]$ constraints,
- constraint set $D6$ comprises $MSKCT$ constraints,
- constraint set $D7$ comprises $MSK$ constraints,
- constraint set $D9$ comprises $T$ constraints,
- constraint set $D10$ comprises $MSCT$ constraints,

which all add up to

$$CN_{\mathcal{HAP}_2^{Lagrange}} = MSKCT\,[CT-1]\,[K-1]+ \\ MSKCT\,[M-1]\,[K-1]+4MSKCT+MSK+MSCT+T. \quad (4.36)$$

For the formulation $\mathcal{HAP}^{Eff}_{MBQCP}$, we have the following numbers of variables:

- The numbers of binary variables $\phi_{m,k}, \theta_m, y_{m,i,c,t}$ are the $MK$,$M$ and $MSCT$ respectively giving a total number of binary variables $MK+M+MSCT$.
- The number of continuous variables:
  - $p_{m,i,c,t}$ are $MSCT$,
  - $u_{m,i,c,t}$ are $MSCT$,
  - $w_{m,i,c,t}$ are $MSCT$
  - $U_{m,j}$ are $M\,[SCT-2]$, and
  - $W_{m,j}$ are $M\,[SCT-2]$.

  all adding up to $3MSCT+2M\,[SCT-2]$ continuous variables.

The number of binary and continuous variables add up to:

$$VN_{\mathcal{HAP}^{Eff}_{MBQCP}} = 4MSCT+2M\,[SCT-2]+MK+M. \quad (4.37)$$

The number of constraints (excluding bounds and binary constraints) in each constraint set for $\mathcal{HAP}^{Eff}_{MBQCP}$ are as follows:

- Constraint set $\overline{C1}$ consists of $MK$ constraints,
- Constraint set $\overline{C2}$ consists of $SCT$ constraints,

- Constraint set $\overline{C3}$ consists of $MK$ constraints,
- Constraint set $\overline{C4}$ consists of $MSCT$ constraints,
- Constraint set $\overline{C5}$ consists of $MSCT$ constraints,
- Constraint set $\overline{C6}$ consists of $T$ constraints,
- Constraint set $\overline{C8}$ consists of $MSKCT$ constraints,
- Constraint set $\overline{C9}$ consists of $M$ constraints,
- Constraint set $\overline{C10}$ consists of $M$ constraints,
- Constraint set $\overline{Q1a}$ consists of $M$ constraints,
- Constraint set $\overline{Q1b}$ consists of $M\,[SCT-3]$ constraints,
- Constraint set $\overline{Q1c}$ consists of $M\,[SCT-3]$ constraints,
- Constraint set $\overline{Q1d}$ consists of $M$ constraints,
- Constraint set $\overline{Q1e}$ consists of $M$ constraints,
- Constraint set $\overline{Q2}$ consists of $MSKCT$ constraints,
- Constraint set $\overline{Q3a}$ consists of $M$ constraints,
- Constraint set $\overline{Q3b}$ consists of $M\,[SCT-3]$ constraints,
- Constraint set $\overline{Q3c}$ consists of $M\,[SCT-3]$ constraints,
- Constraint set $\overline{Q3d}$ consists of $M$ constraints,
- Constraint set $\overline{Q3e}$ consists of $M$ constraints,

- Constraint set $\overline{Q4}$ consists of $MSKCT$ constraints,

which all add up to

$$CN_{\mathcal{HAP}_{MBQCP}^{Eff}} = 2MK + SCT + 2MSCT + T + 8M + 4M\left[SCT - 3\right] + 3MSKCT. \tag{4.38}$$

Finally, both the formulations $\mathcal{HAP}_2^{Lagrange}$ and $\mathcal{HAP}_{MBQCP}^{Eff}$ consist of bilinear terms. By counting the bilinear terms in $\mathcal{HAP}_2^{Lagrange}$ obtained from constraints sets $D3$ and $D4$ we get:

$$N_{\mathcal{HAP}_2^{Lagrange}}^{BiL} = M^2S^2KCT + MSKCT. \tag{4.39}$$

Also, by counting the bilinear terms in constraint sets $\overline{C8}, \overline{Q1a}, \overline{Q2}, \overline{Q1b}, \overline{Q1c}, \overline{Q1d}, \overline{Q1e}, \overline{Q3a}, \overline{Q3b}, \overline{Q3c}, \overline{Q3d}, \overline{Q3e}$ and $\overline{Q4}$ we get:

$$N_{\mathcal{HAP}_{MBQCP}^{Eff}}^{BiL} = M^2S\left(S-1\right)KCT + 2MSKCT\left[1 + M\left(S-1\right)\right] + 4M\left[SCT - 3\right] + 6M. \tag{4.40}$$

We graphically illustrate a comparison of efficiency for the two formulations $\mathcal{HAP}_2^{Lagrange}$ and $\mathcal{HAP}_{MBQCP}^{Eff}$ in figures 4.3, 4.4, 4.5, 4.6 and 4.7. In these figures we compare the number of binary variables, continuous variables, total number of variables, number of constraints and number of bilinear terms for both formulations. We refer to the indices $m$, $i$, $k$, $c$ and $t$ as the problem "dimensions". Therefore there are five dimensions for the problem in both formulations which are the number of multicast sessions, the number of HAP antennas on-board, the number of users in the service area, the number of sub-channels and the number of time slots respectively. We vary the dimensions of the problem as follows:

- The number of multicast sessions $M$ is varied in the range $1 - 250$,

- the number of antennas on-board $S$ is varied in the range $1 - 20$,
- the number of users $K$ in the service area is varied in the range $1 - 500$,
- the number of available sub-channels $C$ is varied in the range $1 - 32$ and
- the number of available sub-channels $T$ is varied in the range $1 - 24$.

Figures 4.3, 4.4, 4.5, 4.6 and 4.7 are comprised of five plots each in which one dimension is varied within its ranges mentioned above and the others are kept fixed at values equal to their maximums in their respective ranges.

The results in Figure 4.3 show that the number of binary variables for $\mathcal{HAP}_{MBQCP}^{Eff}$ is way lower than those in $\mathcal{HAP}_{2}^{Lagrange}$. On the other hand in Figure 4.4, the number of continuous variables $\mathcal{HAP}_{MBQCP}^{Eff}$ are almost 4 times those in$\mathcal{HAP}_{2}^{Lagrange}$ in the worst case for the given example. However by looking at both figures 4.3 and 4.4 we can see that the number of continuous variables in both formulations are much lower than the binary variables which makes the total number of variables in Figure 4.5 almost equivalent to the total number of binary variables. Moreover, it is well known that when there are both binary variables and continuous variables in a problem, the binary variables are the main cause of algorithmic complexity involved in solving the problem. Therefore, comparing the numbers of continuous and binary variables in both formulations, we see that $\mathcal{HAP}_{MBQCP}^{Eff}$ has a much lower complexity compared to $\mathcal{HAP}_{2}^{Lagrange}$.

Taking a look at the number of total constraints in Figure 4.6, we see that the number of constraints in formulation $\mathcal{HAP}_{MBQCP}^{Eff}$ is far lower than$\mathcal{HAP}_{2}^{Lagrange}$. This comes at the cost of up to three times larger number of bilinear terms, in the worst case for this example, for $\mathcal{HAP}_{MBQCP}^{Eff}$ in all dimensions as Figure 4.7 shows. Notice the similar behaviors for both $\mathcal{HAP}_{MBQCP}^{Eff}$ and $\mathcal{HAP}_{2}^{Lagrange}$ in Figure 4.7 for each

dimension. For the dimensions of the number of multicast sessions, $m$, and the number of HAP antenna onboard, $i$, the number of bilinear for both formulations grow quadratically. For the other three dimensions, the growth is linear.

In Chapter 5, the solution methods for formulation $\mathcal{HAP}^{Eff}_{MBQCP}$ are provided.

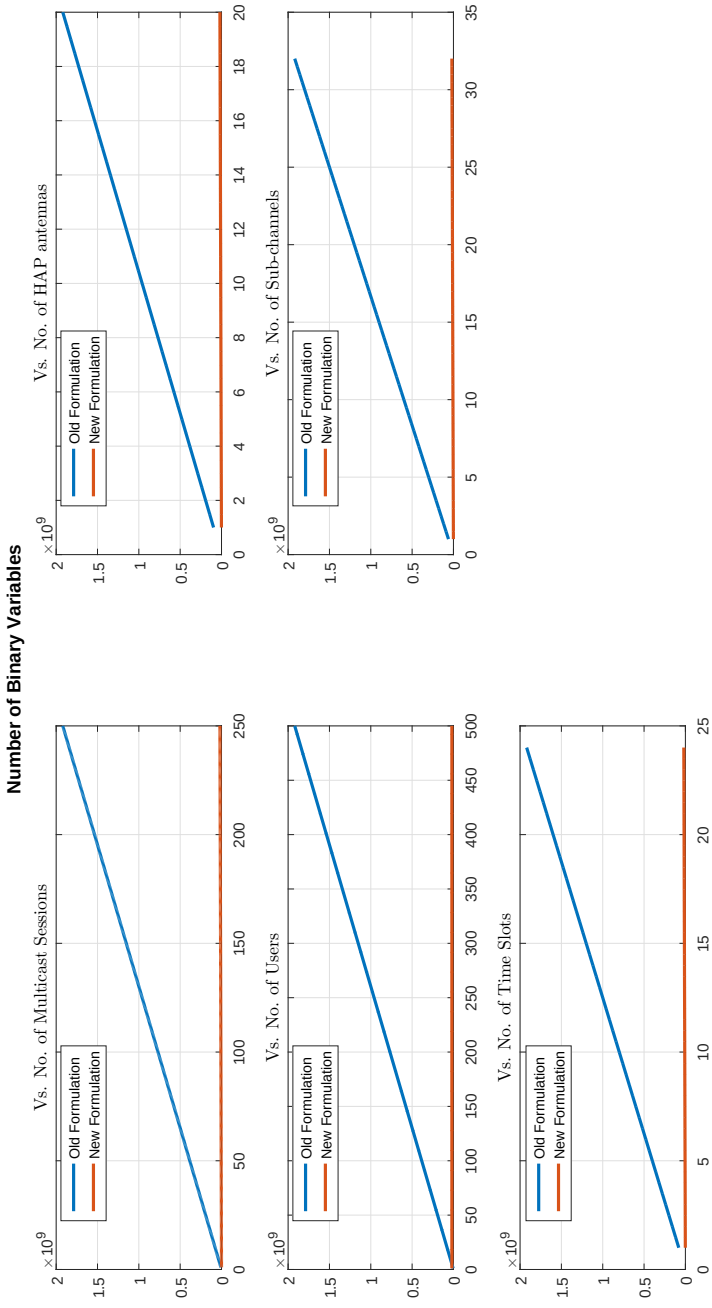



Figure 4.3: Illustration of the number of binary variables versus the different problem dimensions for $\mathcal{HAP}_2^{Lagrange}$ (old formulation) and $\mathcal{HAP}_{MBQCP}^{Eff}$ (new formulation)

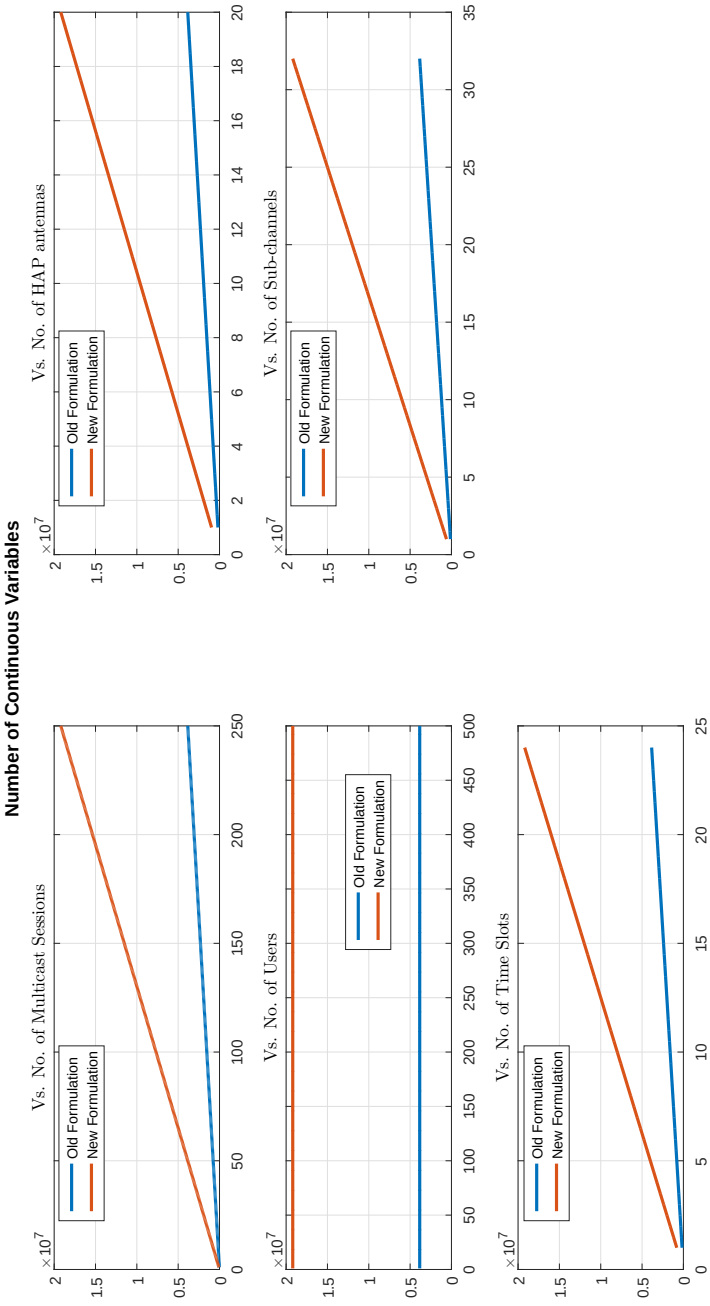



Figure 4.4: Illustration of the number of continuous variables versus the different problem dimensions for $\mathcal{HAP}_2^{Lagrange}$ (old formulation) and $\mathcal{HAP}_{MBQCP}^{Eff}$ (new formulation)

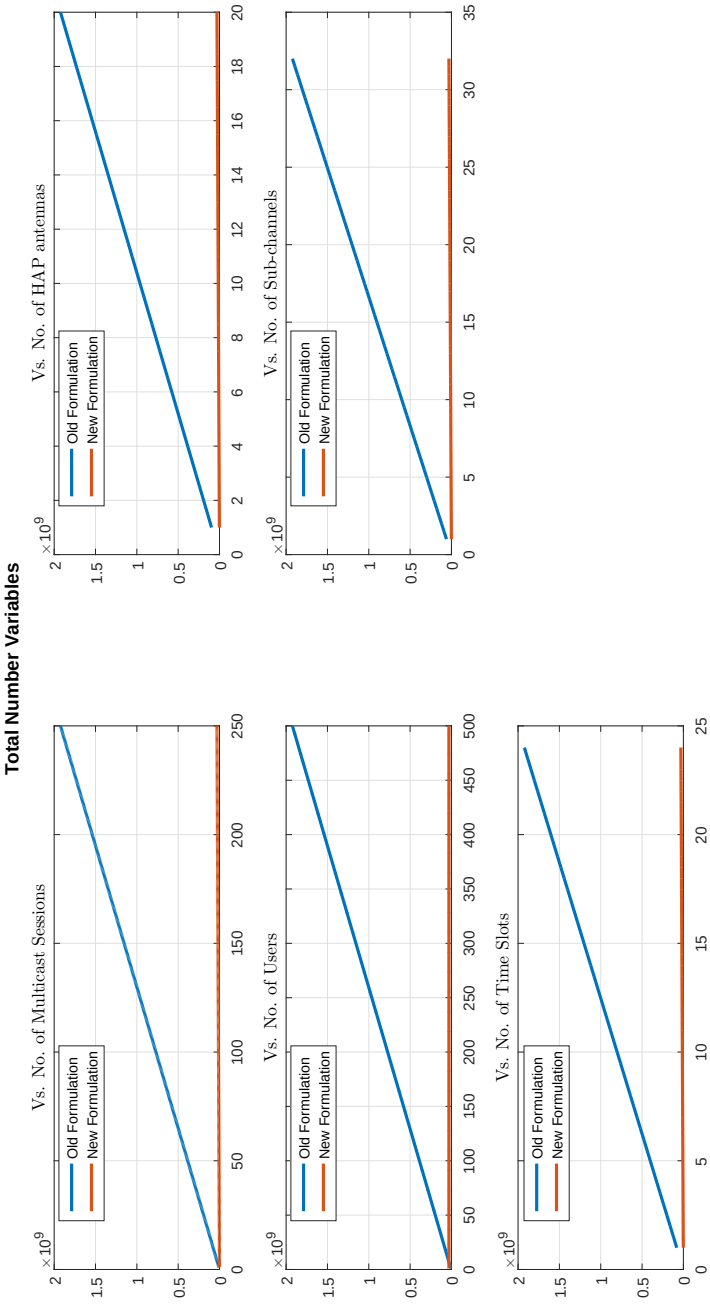



Figure 4.5: Illustration of the total number of variables versus the different problem dimensions for $\mathcal{HAP}_2^{Lagrange}$ (old formulation) and $\mathcal{HAP}_{MBQCP}^{Eff}$ (new formulation)

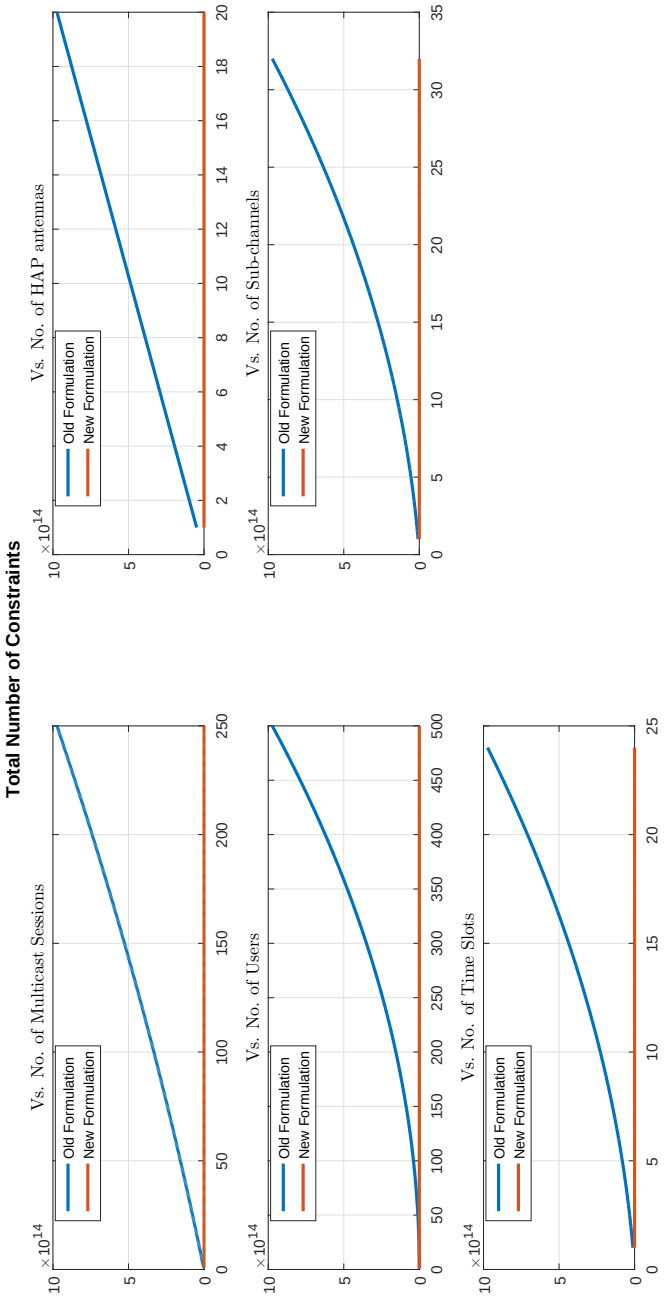



Figure 4.6: Illustration of the total number of constraints versus the different problem dimensions for $\mathcal{HAP}_2^{Lagrange}$ (old formulation) and $\mathcal{HAP}_{MBQCP}^{Eff}$ (new formulation)

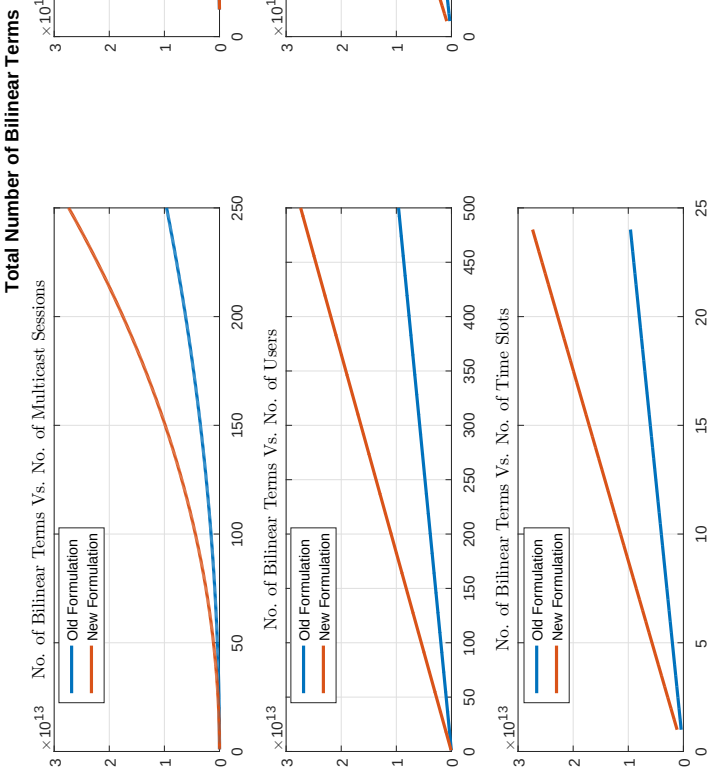



Figure 4.7: Illustration of the total number of bilinear terms versus the different problem dimensions for $\mathcal{HAP}_2^{Lagrange}$ (old formulation) and $\mathcal{HAP}_{MBQCP}^{Eff}$ (new formulation)

# Chapter 5

# Proposed Solution Method for E-SysMod Formulation ($\mathcal{HAP}^{Eff}_{MBQCP}$)

This chapter explains how formulation $\mathcal{HAP}^{Eff}_{MBQCP}$ is solved. An approach similar to [23] and [21] is used in which, an outer approximation is generated by linear underestimation of the non-convex quadratic constraints to relax the problem's feasible region. The problem becomes a *mixed binary linear program* (MBLP) and hence an LP solver can be used in a branch and cut algorithm to solve $\mathcal{HAP}^{Eff}_{MBQCP}$.The branch-and-bound (BnB) algorithm recursively splits the problem into smaller subproblems, thereby creating a branching tree and implicitly enumerating all potential solutions. At each subproblem, domain propagation is performed to exclude further values from the variables' domains, and a relaxation may be solved to achieve an upper (dual) bound. The relaxation is then strengthened by adding further valid constraints, which cut off the optimum of the relaxation. Primal heuristics are integrated in the BnB procedure to improve the lower (primal) bound. The solver used

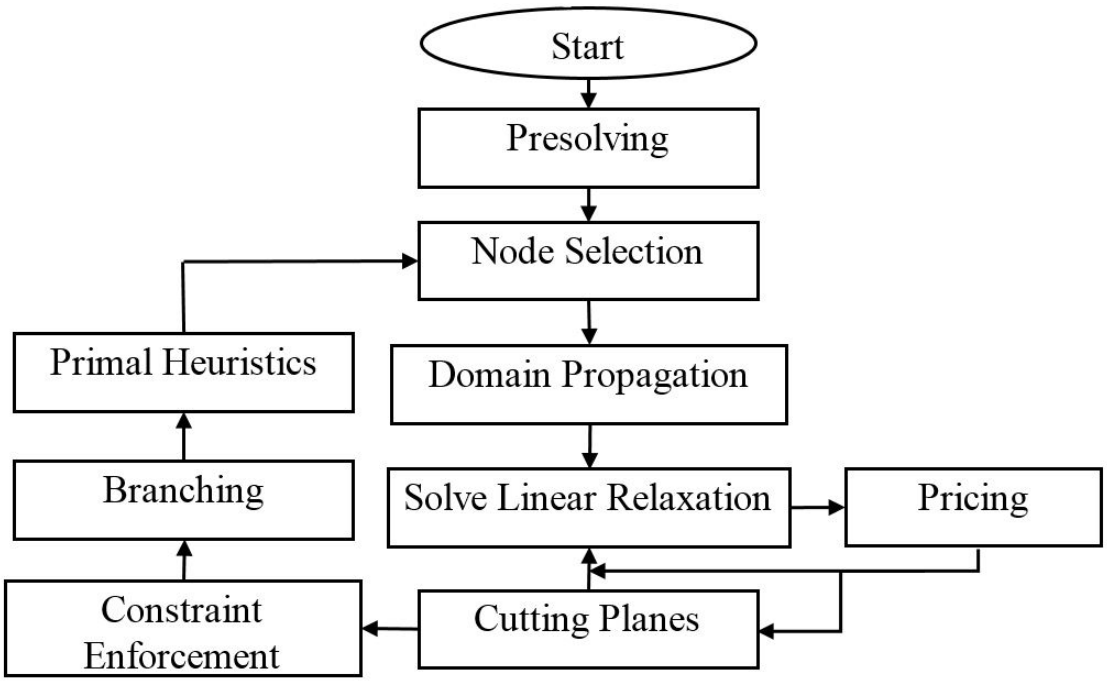



Figure 5.1: Flowchart illustrating the interrelation between the SCIP solver components used for solving $\mathcal{HAP}^{Eff}_{MBQCP}$

for the experiments is *Solving Constraint Integer Programs* (SCIP) which is capable of solving a non-convex *mixed integer quadratically constraint program* (MIQCP) to optimality in finite time [64]. The interdependencies between the algorithmic components of SCIP solver are shown in Figure 5.1. In this chapter, an explanation for the components used in the experiments (Chapter 6) done for $\mathcal{HAP}^{Eff}_{MBQCP}$ are provided. The components used are:

- Presolving
- Branching
- Separating Cuts
- Domain Propagation

- Primal Heuristics

The SCIP solver is rich with choices of different strategies for each of the components listed above that could be selected based on the problem structure or conducted experiments. The explanation provided for each of these components focuses on how their respective strategies and elements are used specifically for $\mathcal{HAP}^{Eff}_{MBQCP}$. Table 5.1 provides a summary of the specific methods used in each of those components for $\mathcal{HAP}^{Eff}_{MBQCP}$. For more detailed explanation on these components and other components in Figure 5.1, refer to [64], [23], [21] and [65].

This chapter is organized as follows. Section 5.1 briefly outlines the main purpose of the presolving phase and explains a linearization reformulation presolving technique used for a specific set of quadratic constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$. Section 5.2 explains the BnB algorithm which is invoked right after the presolving phase. Section 5.3 outlines the branching techniques considered. Section 5.4 explains the cutting planes procedure and the types of cuts used. Section 5.5 explains domain propagation and the utilized constraint specific propagation techniques as well as the dual propagation techniques that take into account the objective function value of the LP relaxation. Finally Section 5.6 lists the set of heuristics integrated in the BnB algorithm to solve $\mathcal{HAP}^{Eff}_{MBQCP}$.

## 5.1 Presolving

Presolving is a set of operations invoked before the branch-and-bound algorithm to transform the problem instance to an easier instance to solve. The three main tasks for presolving are:

1. Reducing the size of the instance by removing redundant constraints and fixed variables.

Table 5.1: Different techniques used in each of the solution components used for $\mathcal{HAP}^{Eff}_{MBQCP}$.

| **Presolving** | **Branching** | **Separating Cuts** | **Domain Propagation** | **Primal Heuristics** |
|---|---|---|---|---|
| A reformulation linearization technique for constraint set $\overline{C8}$ in $\mathcal{HAP}^{Eff}_{MBQCP}$ | Random Branching | McCormick Separators | Interval-arithmetic based quadratic constraints forward and backward propagators | Simple rounding |
| | Most Infeasible Branching | Implied Cuts | General linear constraint propagators | Rounding |
| | Pseudocost Branching | Clique Cuts | Set packing constraints propagator for constraint set $\overline{C2}$ | Integer shifting |
| | Strong Branching | Gomory Cuts | Set Covering Constraints propagator for constraint set $\overline{C3}$ | Pseudocost diving |
| | Hybrid Strong/Pseudocost Branching | | Variable bound constraints propagator for constraint set $\overline{C5}$ | Feasibility Pump |
| | Reliability Branching | | A problem structure based propagation scheme | Clique partition based LNS |
| | Inference Branching | | Knapsack constraint based constraint propagator for the objective function | RENS |
| | Cloud Branching | | Reduced costs propagation for the objective function | Undercover heuristic |
| | | | | RINS |
| | | | | Crossover |

2. Strengthening the LP relaxation by exploiting integrality to improve the constraint coefficients and tighten variable bounds.

3. Extracting implications and clique information from the instance that can be used for cutting plane generation (explained in Section 5.4) or branching.

Applying domain propagation (explained in Section 5.5) to the root node to tighten the decision and any added auxiliary variables is one of the simple forms of presolving. There are *primal* and *dual* presolving techniques. The *primal* presolving techniques do the reductions such that the feasible set of solutions of the instance are not altered. On the other hand *dual* presolving considers the direction of change in the objective function when reducing the bounds of variables or the coefficients of a linear constraint. Any reformulations resulting from fixed or aggregated variables in linear constraints are realized also in the quadratic constraints. If the quadratic terms vanish, the constraint is upgraded to a linear type and the linear constraint handler takes the further processing from there. The details related to the presolving techniques related to linear and quadratic constraints for a *mixed integer quadratically constrained program* (MIQCP) can be found in [64] and [23] respectively.

In the experiments performed for $\mathcal{HAP}^{Eff}_{MBQCP}$ for the presolving phase, we consider one of the reformulations in [23] which is a linear reformulation for bilinear terms that are a product of a binary variable $\ddot{x}$ with a linear term i.e. $\ddot{x}\sum_{j=1}^{k} a_i\ddot{y}$. This type of reformulation is applicable to the constraint set $\overline{C8}$ in $\mathcal{HAP}^{Eff}_{MBQCP}$ where the terms that consist of the product of binary variables and linear terms are $y_{m,i,c,t}\sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}p_{m,i',c,t}$. The product is replaced by the auxiliary variable $z$ and the linear constraints:

$$\tilde{p}^{L}y_{m,i,c,t} \leq z \leq \tilde{p}^{U}y_{m,i,c,t}, \tag{5.1a}$$

$$\sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}p_{m,i,c,t} - \tilde{p}^{L}\left(1-y_{m,i,c,t}\right) \leq z \leq \sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}p_{m,i,c,t} - \tilde{p}^{U}\left(1-y_{m,i,c,t}\right), \tag{5.1b}$$

where

$$\tilde{p}^{L} = \sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}\tilde{p}^{L}_{m,i,c,t}, \tag{5.2a}$$

$$\tilde{p}^{U} = \sum_{m=1}^{M}\sum_{\forall i'\neq i} g_{i',k,c,t}\tilde{p}^{U}_{m,i,c,t}, \tag{5.2b}$$

given the local bounds $\tilde{p}^{L}_{m,i,c,t}$ and $\tilde{p}^{U}_{m,i,c,t}$. This reformulation linearizes the constraint set $\overline{C8}$ at the expense of introducing one continuous variable for each constraint in the set, and four linear constraints for each quadratic constraint in $\overline{C8}$. In Chapter 6, algorithmic performance, with and without, the presolving linearization reformulation explained in this section, for different number of presolving rounds, are presented.

After the presolving phase, the BnB algorithm is invoked. Any reference for $\mathcal{HAP}^{Eff}_{MBQCP}$ in the rest of this chapter refers to the instance *after* going through the presolving phase. The BnB algorithm is provided and explained in Section 5.2.

## 5.2 Branch and Bound based Solution Framework

The branch and bound scheme is a general framework used in solving non-convex problems, which include MBQCPs and MBLPs, to divide it into smaller problems that can be solved (conquered) and hence is a divide and conquer algorithm [62]. The best local solution across all the subproblems, which are referred to as nodes, is the global solution of the entire problem. *Branching* is basically the splitting of a subproblem into two or more nodes. Since the discrete variables that we have in $\mathcal{HAP}^{Eff}_{MBQCP}$ are binary by nature, binary branching is the only choice, i.e. no more than two children nodes for any node in the tree. The *root* node is the whole problem

$\mathcal{HAP}^{Eff}_{MBQCP}$ before division while the the rest of the nodes are smaller subproblems that have either been solved or still need to be solved. A *leaf* node is a subproblem that either has not yet been processed or one that has been processed and *pruned*. Any pruned node has no descendants. A node gets pruned when it proves that none of its descendants can give a better primal feasible solution than the *incumbent*. An incumbent is the name given to the best feasible solution found so far during the course of the BnB algorithm. All the nodes (un-pruned leafs) that are yet to be processed are stored in a queue $\ddot{\mathcal{L}}$.

The *bounding* step avoids complete enumeration of potential solutions of the problem. The better the dual $\ddot{c}_{dual}$ and primal $\ddot{c}_{primal}$ bounds are, the more effective the bounding process in excluding subproblems from solving. The dual bound is found by solving the relaxation $\mathcal{Q}_{relax}$ of a node subproblem $\mathcal{Q}$. The relaxation $\mathcal{Q}_{relax}$ for $\mathcal{HAP}^{Eff}_{MBQCP}$ is obtained by replacing all the bilinear terms individually by McCormick linear under-estimators as explained in Section 5.4 [23], and by relaxing all the binary variables into the continuous domain [0,1]. Algorithm 1 illustrates the main procedures of a BnB framework. For abstract notation, all the decision variables in $\mathcal{HAP}^{Eff}_{MBQCP}$ are represented by the decision vector $\ddot{\mathbf{x}}$. Furthermore, an arbitrary decision variable is referred to as $\ddot{x}_j$, where $j \in \tilde{N}$ for any variable and if the variable is binary then additionally $j \in \mathcal{B}$, where $\tilde{N}$ is the set of all decision variables and $\mathcal{B}$ is the set of all binary decision variables in $\mathcal{HAP}^{Eff}_{MBQCP}$.

The input to the algorithm is a presolved instance of $\mathcal{HAP}^{Eff}_{MBQCP}$ which resembles the root node. If the instance is feasible then the output of the algorithm is the global optimal solution $\ddot{\mathbf{x}}^{\mathcal{HAP}^{Eff}_{MBQCP}}_{opt}$ and the corresponding objective function value $\ddot{c}^{\mathcal{HAP}^{Eff}_{MBQCP}}_{opt}$, otherwise the algorithm concludes that the instance is infeasible. The algorithm is initialized by assigning the root node $\mathcal{HAP}^{Eff}_{MBQCP}$ into the empty node

**Algorithm 1** Branch-and-Bound Framework for solving $\mathcal{HAP}^{Eff}_{MBQCP}$

**Input**: Maximization of an instance of $\mathcal{HAP}^{Eff}_{MBQCP}$
**output** : Optimal solution $\ddot{\mathbf{x}}_{opt}^{\mathcal{HAP}^{Eff}_{MBQCP}}$ with objective function value $\ddot{c}_{opt}^{\mathcal{HAP}^{Eff}_{MBQCP}}$ or conclusion that $\mathcal{HAP}^{Eff}_{MBQCP}$ has no solution by $\ddot{c}_{opt}^{\mathcal{HAP}^{Eff}_{MBQCP}} = -\infty$
**Initialize**:
$\mathcal{Q} \leftarrow \mathcal{HAP}^{Eff}_{MBQCP}$
$\ddot{\mathcal{L}} = \{\mathcal{Q}\}$
$\ddot{c}_{primal} = -\infty$
**Abort**:
**if** $\ddot{\mathcal{L}} = \emptyset$ **then**
  $\ddot{\mathbf{x}}_{opt}^{\mathcal{HAP}^{Eff}_{MBQCP}} \leftarrow \ddot{\mathbf{x}}_{BFS}$
  $\ddot{c}_{opt}^{\mathcal{HAP}^{Eff}_{MBQCP}} \leftarrow \ddot{c}_{primal}$
  **STOP**
**end if**
**Select**:
Choose $\mathcal{Q} \in \ddot{\mathcal{L}}$ and
$\ddot{\mathcal{L}} \leftarrow \ddot{\mathcal{L}} \setminus \{\mathcal{Q}\}$
**Solve**:
Solve the linear relaxation $\mathcal{Q}_{relax}$ after applying McCormick under-estimators to all bilinear terms of $\mathcal{HAP}^{Eff}_{MBQCP}$.
**if** $\mathcal{Q}_{relax} = \emptyset$ **then**
  $\ddot{c}_{dual} \leftarrow -\infty$
**else**
  let $\ddot{\mathbf{x}}_{relax}$ be the optimal solution of $\mathcal{Q}_{relax}$ and $\ddot{c}_{dual}$ its objective function value.
**end if**
**Bound**:
**if** $\ddot{c}_{dual} \leq \ddot{c}_{primal}$ **then**
  Prune node $\mathcal{Q}$
  **goto** step (6)
**end if**
**Feasibility Check**:
**if** $\ddot{\mathbf{x}}_{relax}$ is feasible for $\mathcal{HAP}^{Eff}_{MBQCP}$ **then**
  $\ddot{\mathbf{x}}_{BFS} \leftarrow \ddot{\mathbf{x}}_{relax}$
  $\ddot{c}_{primal} \leftarrow \ddot{c}_{dual}$
  **goto** step (6)
**end if**
**Branch**:
Divide $\mathcal{Q}$ into two subproblems $\mathcal{Q} = \mathcal{Q}_0 \cup \mathcal{Q}_1$
$\ddot{\mathcal{L}} \leftarrow \{\mathcal{Q}_0 \cup \mathcal{Q}_1\}$
**goto** step (6).

queue $\ddot{\mathcal{L}}$. The **Abort** procedure is invoked when the node queue is empty to return the best feasible solution found so far $\ddot{\mathbf{x}}_{BFS}$ and its corresponding objective function value $\ddot{c}_{primal}$. If the node queue still has further unprocessed nodes, the **Select** procedure is invoked to choose a node $\mathcal{Q}$ depending on a *node selection criterion* before it gets removed from the queue. The relaxation of the selected node $\mathcal{Q}_{relax}$ is solved using the simplex algorithm [66] after applying McCormick under-estimators to outer-approximate the non-convex quadratic constraints of $\mathcal{HAP}^{Eff}_{MBQCP}$. If $\mathcal{Q}_{relax}$ is found infeasible then $\ddot{c}_{dual}$ is assigned the smallest possible value (theoretically $-\infty$) to insure that the node gets pruned in the **Bound** step. Otherwise $\ddot{\mathbf{x}}_{relax}$ becomes the solution of $\mathcal{Q}_{relax}$ and $\ddot{c}_{dual}$ is its corresponding objective function value.

The **Bound** procedure is responsible for pruning branches from the search tree whose descendant nodes are guaranteed not to include any solutions better than the currently available best feasible solution (incumbent) $\ddot{\mathbf{x}}_{BFS}$. This is known using the simple comparison between the obtained $\ddot{c}_{dual}$ from the **Solve** procedure and the objective function value $\ddot{c}_{primal}$ for the incumbent. In a maximization problem, like $\mathcal{HAP}^{Eff}_{MBQCP}$, if the dual (upper) bound is lower than the primal (lower) bound value, this is an indication that any of the descendants of the node can never have any better feasible solutions. If the node gets pruned, the algorithm goes back to the **Abort** procedure to check if there are any nodes left in the queue $\ddot{\mathcal{L}}$. If no pruning occurs, the **Feasibility Check** procedure is invoked and sets the solution $\ddot{\mathbf{x}}_{relax}$ of the relaxed subproblem $\mathcal{Q}_{relax}$ as the solution of the $\mathcal{Q}$ itself only if the solution $\ddot{\mathbf{x}}_{relax}$ is feasible to $\mathcal{Q}$. If $\ddot{\mathbf{x}}_{relax}$ is not feasible to $\mathcal{Q}$, the **Branch** procedure then gets invoked to divide node $\mathcal{Q}$ into further nodes. This happens by selecting an appropriate variable to branch on. Since all the discrete variables in $\mathcal{HAP}^{Eff}_{MBQCP}$ are binary, then the branching is also binary. After branching takes place, the **Abort**

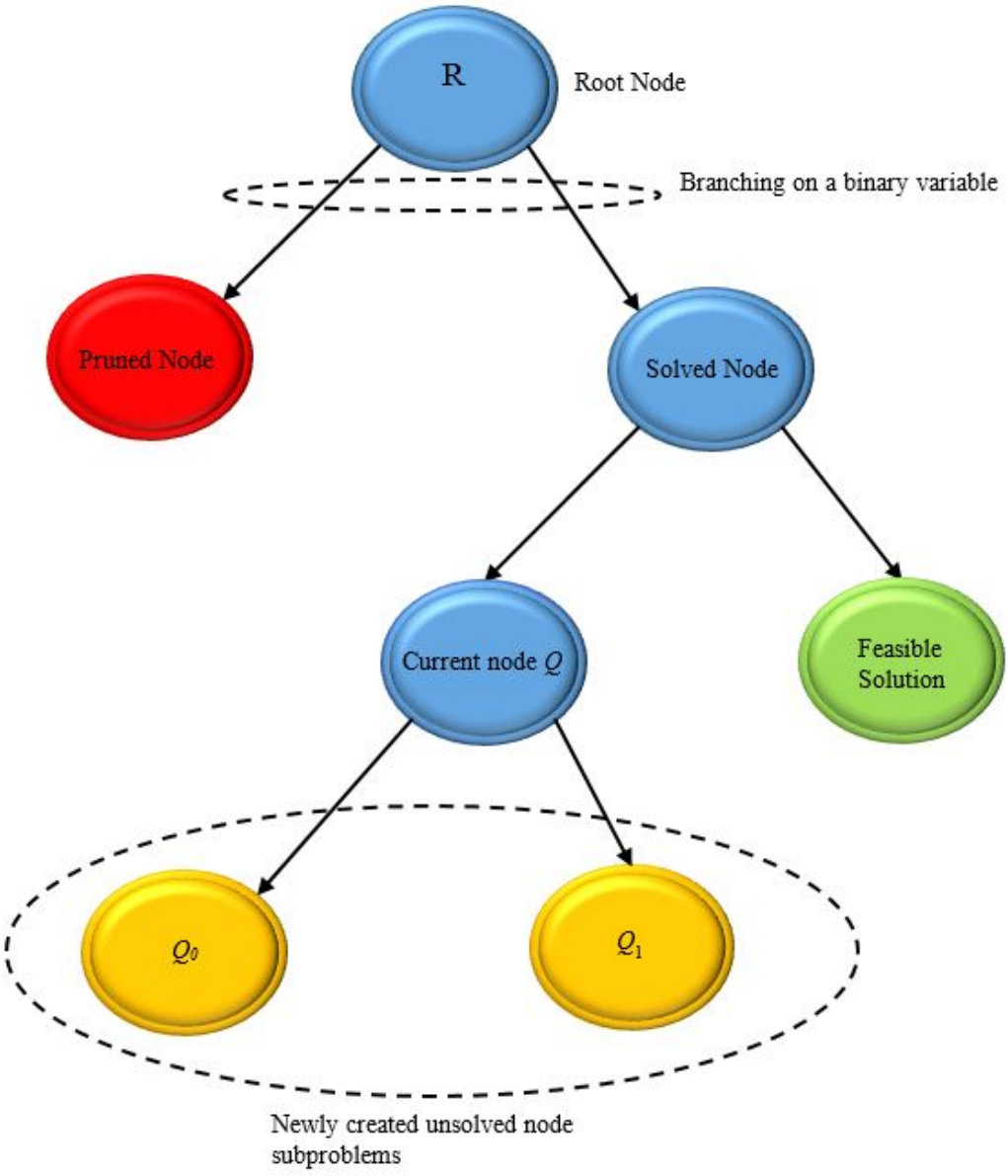



Figure 5.2: Illustration of BnB Tree

procedure gets invoked to check whether there are any unprocessed nodes left in $\ddot{\mathcal{L}}$. Figure 5.2 illustrates how the BnB tree looks like.

The node selection indicated by the **Select** procedure, the branching rules indicated by **Branch** and the relaxation whose solution is used in the **Bound** procedure

all have a major impact on how early good feasible solutions can be found and how fast the dual bounds decreases. They all influence the **Bound** procedure which is expected to prune large parts of the BnB tree. An explanation for different branching rules used in the experiments conducted on $\mathcal{HAP}^{Eff}_{MBQCP}$ is provided in Section 5.3.

## 5.3 Branching

*Branching* is the splitting of a node into two or more nodes by adding new upper and lower bounds on one of the variables which is called the *branching variable*. By reducing a variables domain, the created children nodes have smaller feasible regions each, which helps reduce the work required to find feasible solutions better than the currently available best feasible solution $\ddot{\mathbf{x}}_{BFS}$. Another type of branching which is not considered in this thesis is branching on hyperplanes [67]. In this thesis we use splitting with hyperplanes to cut off infeasible solutions rather than divide the problem into further smaller subproblems.

One advantage of using LP relaxation within BnB is in the branching process. Branching changes the created children subproblems from a parent node $\mathcal{Q}$ by introducing new upper or lower bound to one variable which preserves the dual feasibility of the solution obtained for $\mathcal{Q}_{relax}$. This enables the use of dual simplex using the parent node solution as a warm up start and hence the work done in solving $\mathcal{Q}_{relax}$ counts towards solving the relaxation of its children, which saves a lot of further computational work.

For $\mathcal{HAP}^{Eff}_{MBQCP}$, the only discrete variables are binary and hence two nodes only are created by branching on binary variables. A score $s_j^{branch}$ is calculated for each

variable using the equation [65]:

$$s_j^{branch} = max\left\{\ddot{q}_j^0, \bar{\epsilon}\right\} max\left\{\ddot{q}_j^1, \bar{\epsilon}\right\}, \tag{5.3}$$

which measures the improvement in the dual bound by branching on the variable $\ddot{x}_j$ for $j \in \mathcal{B}$ where:

- $\mathcal{B}$ is the set of binary variables in $\mathcal{HAP}^{Eff}_{MBQCP}$,
- $\ddot{q}_j^0$ is a function that is directly dependent on and proportional to the dual bound improvement $\ddot{\Delta}_j^0$ over the parent subproblem's relaxation $\mathcal{Q}_{relax}$ by setting $\ddot{x}_j = 0$,
- $\ddot{q}_j^1$ is a function that is directly dependent on and proportional to the dual bound improvement $\ddot{\Delta}_j^1$ over the parent subproblem's relaxation $\mathcal{Q}_{relax}$ by setting $\ddot{x}_j = 1$,
- $\bar{\epsilon}$ which is a very small positive constant which is necessary to compare $\left(\ddot{\Delta}_j^0, \ddot{\Delta}_j^1\right)$ and $\left(\ddot{\Delta}_k^0, \ddot{\Delta}_k^1\right)$ and is set by default to $\bar{\epsilon} = 10^{-6}$ in SCIP.

There many different ways by which a branching variable can be selected, and of course, they can have different performances in bound improvement which are illustrated in the results provided in Chapter 6. The following branching schemes are considered for comparison in the experiments conducted for $\mathcal{HAP}^{Eff}_{MBQCP}$.

### *5.3.1 Random Branching*

As the name indicates, there is nothing done in this technique except arbitrarily selecting any unfixed binary variable that violates the binary condition.

### *5.3.2 Most Infeasible Branching*

This rule chooses the variable with the smallest tendency to be rounded either downwards or upwards. Hence for binary variables with fractional values in the solution of $\mathcal{Q}_{relax}$, the one that is closest to 0.5 receives the highest score. The score function for a fractional binary variable is given as:

$$s_j^{branch} = min\left\{\ddot{x}_j^{relax}, 1 - \ddot{x}_j^{relax}\right\}, \quad j \in \mathcal{B}. \tag{5.4}$$

### *5.3.3 Pseudocost Branching*

This type of branching keeps a history for the average performance of each variable that has been branched on so far. This is measured as the average improvement in the bound for all the times the variable has been branched on. To obtain the variable scores, first the unit bound change for $\ddot{x}_j$ is found using

$$\varsigma_j^0 = \frac{\ddot{\Delta}^0}{\ddot{x}_j^{relax}} \qquad and \qquad \varsigma_j^1 = \frac{\ddot{\Delta}^1}{\left(1 - \ddot{x}_j^{relax}\right)}. \tag{5.5}$$

Let the aggregate unit bound changes be $\ddot{\sigma}_j^0$ and $\ddot{\sigma}_j^1$ over all nodes for which $\ddot{x}_j$ was selected for branching and the numbers of these nodes be $\eta_j^0$ and $\eta_j^1$ then the pseudocosts of $\ddot{x}^j$ are the averages:

$$\Psi_j^0 = \frac{\ddot{\sigma}_j^0}{\eta_j^0} \qquad and \qquad \Psi_j^1 = \frac{\ddot{\sigma}_j^1}{\eta_j^1}. \tag{5.6}$$

. The score is then given as :

$$s_j^{branch} = max\left\{\ddot{x}_{relax}^j \Psi_j^0, \bar{\epsilon}\right\}.max\left\{\left(1 - \ddot{x}_{relax}^j\right)\Psi_j^1, \bar{\epsilon}\right\}. \tag{5.7}$$

During the course of Algorithm 1, a variable that has not yet been selected for branching is termed to be *uninitialized*, a term that will be used in subsequent subsections.

### *5.3.4 Strong Branching*

*Strong branching* can be summarized as solving the linear relaxations that result from branching on each binary branching candidate and choosing the variable that gives the best bound improvement to branch on. It is hence expected that to obtain the optimal solution for a problem instance, the number of nodes required to be explored is going to be low but the number of simplex LP iterations is going to be too high which consumes a lot of processing time. The full set of branching candidates $F_{cand}$ are all binary variables with fractional values. The entire set could be used in *strong branching* or only a subset $\overline{F} \subset F_{cand}$. If the entire set is used, the branching scheme is emphasized upon by the term *full strong branching*.

### *5.3.5 Hybrid Strong/ Pseudocost Branching*

Strong branching and *pseudocost branching* both have their advantages and draw backs. For strong branching, as mentioned in Subsection 5.3.4, the number of nodes to be explored before the optimal solution is reached is expected to be low. However, the time and LP iterations expended could be too high. On the other hand *pseudocost* branching is not expected to expend too many LP iterations (and hence time) to obtain the optimal solution, but would require more branching operations to do so and hence more number of nodes. This is because at the very beginning of the BnB tree, the *pseudocost branching* scheme has no history to use for guiding its choice on branching. Since the branching decisions near the top of the BnB tree are the most crucial, the absence of early history information would lead to more node explorations.

If the optimal solution is not the main objective and obtaining solutions with low duality gap is desired in a given time, it is expected that strong branching would return a large duality gap since the rate of improvement is expected to be slow for the given solution time due to the high number of LP iterations required. A hybrid branching technique, that combines both schemes, aims to get the best of each and reduce as much as possible the cons each has. It is achieved by implementing strong branching in the upper part of BnB tree up to a certain depth $d$. For nodes that are deeper than $d$ in the tree, pseudocost branching is applied.

### *5.3.6 Reliability Branching*

The branching decision based on pseudocosts either in pure *pseudocost branching* or in *hybrid strong/pseudocost branching* are based on uninitialized values which negatively affect the selection of branching variables. *Reliability branching* uses *strong branching* for variables with uninitialized pseudocosts, and hence is more dynamic than *hybrid strong/pseudocost branching* which uses *strong branching* for a fixed depth in the BnB tree. Furthermore, to use the pseudocosts for branching, *reliability branching* requires that the history for the branching variable be collected for at least $\eta_{rel}$ problems, where $\eta_{rel}$ is the reliability parameter. Hence if $min\left\{\eta_j^0, \eta_j^1\right\} \leq \eta_{rel}$ the variable $\ddot{x}_j$ is called unreliable. Moreover, the work expended in *strong branching* can be reduced by using a small subset of branching variable candidates $\overline{F} \subset F_{cand}$ as well as performing only a few simplex iterations for each candidate in $\overline{F}$ to estimate the changes in the dual bound. The dual bound is the value of the objective function of $\mathcal{Q}_{relax}$. Since the change in the objective function value is greatest in the first few simplex iterations compared to later iterations, the estimate for the dual bound is expected to be close to the actual value.

The $\eta_{rel}$ dynamically changes to restrict the number of strong branching simplex iterations for a given node $\mathcal{Q}$ to [64] :

$$\hat{\gamma}^{max}_{SB} = c_{sbiterquot}\hat{\gamma}_{LP} + \hat{\gamma}^{root}_{SB} + \hat{\gamma}_{fixed}, \tag{5.8}$$

where

- $\hat{\gamma}^{max}_{SB}$ is the number of simplex iterations for for the strong branching done in $\mathcal{Q}$
- $\hat{\gamma}_{LP}$ is the number of regular simplex iterations
- $c_{sbiterquot}$ is maximal fraction of strong branching LP iterations compared to node relaxation LP iterations,
- $\hat{\gamma}_{fixed}$ is a fixed number that can be pre-set.

If the number of strong branching LP iterations $\hat{\gamma}_{SB}$ exceeds $\hat{\gamma}^{max}_{SB}$, then $\eta_{rel}$ is set to zero and *pseudocost branching* is used. If $\hat{\gamma}_{SB} \in [c_{sbiterquot}\hat{\gamma}^{max}_{SB}, \hat{\gamma}^{max}_{SB}]$, $\eta_{rel}$ decreases from $\eta^{max}_{rel}$ to $\eta^{min}_{rel}$ linearly. If $\hat{\gamma}_{SB} < c_{sbiterquot}\gamma_{LP}$, then $\eta_{rel}$ increases in proportion to the quotient $\frac{\hat{\gamma}_{LP}}{\hat{\gamma}^{max}_{SB}}$.

### *5.3.7 Inference Branching*

This technique exploits domain propagation of branching variables. Its main idea is that it selects the variable whose domain tightening (variable fixation in case of binary variables) produces the most domain reductions in other variables. The impact of a variable on domain deductions is obtained from history information, like *pseudocost branching*, that measures the average inferred domain deductions $\ddot{\Phi}^1_j$ and $\ddot{\Phi}^0_j$ given by [65]:

$$\ddot{\Phi}^1_j = \frac{\ddot{\phi}^1_j}{\ddot{\nu}^1_j} \qquad and \qquad \ddot{\Phi}^0_j = \frac{\ddot{\phi}^0_j}{\ddot{\nu}^0_j}, \tag{5.9}$$

where

- $\ddot{\phi}^1_j$ and $\ddot{\phi}^0_j$ are the total deductions by setting the binary variable $\ddot{x}_j$ to 1 or 0 respectively,
- $\ddot{\nu}^1_j$ and $\ddot{\nu}^0_j$ are the numbers of corresponding subproblems for which domain propagation has been applied.

For uninitialized binary variables, clique and implication tables are used to calculate the inference values [65].

### *5.3.8 Cloud Branching*

All branching strategies described above deal with only one optimal fractional solution for $\mathcal{Q}_{relax}$. Whereas LP relaxations are known to be largely degenerate, multiple equivalent optimal solutions are the rule rather than the exception. Therefore considering only one optimal solution yields high possibilities of taking arbitrary, or inefficient branching decisions. Cloud branching exploits the knowledge of a cloud of multiple alternative optimal solutions of the given LP relaxation using dual degeneracy in a mixed integer program [68]. For a given cloud $\mathfrak{C} = \left\{\ddot{\mathbf{x}}^1, ...\ddot{\mathbf{x}}^k\right\}$ of optimal solutions of the LP relaxation, the initial set of branching variable candidates $F(\mathfrak{C})$ contains all the variables that are fractional in at least one solution of the cloud $\mathfrak{C}$. The cloud of solutions is generated in the context of strong branching which solves the LPs that would result from branching on all candidates.

The first step in the cloud branching strategy is to generate a cloud of alternative optimal solutions for the LP relaxation $\mathcal{Q}_{relax}$ of a node $\mathcal{Q}$. This is done by restricting search for the basic feasible variables to the optimal hyperplane of the polyhedron. To implement this type of search, the variables of a given optimal solution, whose reduced

costs are non-zero need to be fixed in the search procedure. To move from one basis to another on the optimal hyperplane, an auxiliary objective function is needed. The one used in our numerical experiments is a feasibility like pump objective function that is implemented in the SCIP solver and was proposed in [68] whose coefficients for the binary variables $j \in \mathcal{B}$ are given as:

$$c_j = \begin{cases} -1 \;\; if \;\; 0 < \ddot{x}^*_j < 0.5 \\ 1 \;\; if \;\; 0.5 \leq \ddot{x}^*_j < 1. \end{cases} \tag{5.10}$$

Using iterations of the primal simplex algorithm on the resulting auxiliary LP $\mathcal{Q}_{Aux}$, an alternative optimum basis to the LP relaxation of the BnB node can be obtained that has the closest hamming distance to the nearest integral point.

After obtaining a cloud $\mathfrak{C}$, the cloud interval for a variable $\ddot{x}_j \in F(\mathfrak{C})$ is given by $\left[l^{\mathfrak{C}}_j, u^{\mathfrak{C}}_j\right]$, where:

$$l^{\mathfrak{C}}_j = min\left\{\ddot{x}^i_j | \ddot{\mathbf{x}}^i \in C\right\}, \tag{5.11a}$$

$$u^{\mathfrak{C}}_j = max\left\{\ddot{x}^i_j | \ddot{\mathbf{x}}^i \in C\right\}. \tag{5.11b}$$

Accordingly, the set $F(\mathfrak{C})$ is partitioned into three which are :

$$F_2 = \left\{j \in F(\mathfrak{C}) \,|0 < l^{\mathfrak{C}}_j \wedge u^{\mathfrak{C}}_j < 1\right\}, \tag{5.12a}$$

$$F_0 = \left\{j \in F(\mathfrak{C}) \,| l^{\mathfrak{C}}_j = 0 \wedge u^{\mathfrak{C}}_j = 1\right\}, \tag{5.12b}$$

$$F_1 = F(\mathfrak{C}) \setminus (F_2 \cup F_0), \tag{5.12c}$$

which shows that for binary variables, the only type of discrete variables in $\mathcal{HAP}^{Eff}_{MBQCP}$, $F_2$ contains the fractional variables of all the solutions in the cloud

$\mathbb{C}$.

Branching on the variables in $F_0$ guarantees that the dual bound in both branching directions will not improve. Those in $F_1$ are guaranteed not to improve the bound in only one direction but hopefully will improve in the other direction. The candidates in $F_2$ are expected to to improve the dual bound in both directions. The cloud purpose is to filter out as many LPs so that strong branching only needs to solve a small subset of those. As long as there are any candidates existing in the set $F_2$, the other two sets are ignored and only the LPs for the candidates in $F_2$ are solved.

## 5.4 Cutting Planes

As mentioned at the end of Section 5.2, relaxation is one of the procedures that can greatly affect the algorithm's computational effort, speed and BnB tree size. For the quadratic non-convex constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$, linear relaxation is achieved using linear outer approximation with McCormick underestimators for all the bilinear terms of the quadratic constraints. The relaxation can be strengthened by using cutting planes that separate sets of solutions from a node relaxation $\mathcal{Q}_{relax}$ including its optimal solution $\ddot{\mathbf{x}}^{opt}_{relax}$ without removing any of the feasible solutions for node $\mathcal{Q}$. Adding cutting planes to the LP relaxation of a subproblem simply changes the simplex tableau by adding new rows that represent the cuts added.

Since the addition of new rows to the simplex tableau does not affect the LP dual feasibility of the solution $\ddot{\mathbf{x}}^{opt}_{relax}$, it could be used as a warm up start for the dual simplex algorithm to solve the strengthened relaxation $\mathcal{Q}^{Strengthened}_{relax}$ of the node $\mathcal{Q}$. This could be repeated several times until a maximum number of LP iterations is achieved, a maximum LP time limit is hit, a maximum number rounds of cut separation or a maximum number cuts is achieved. This procedure utilizes a very

important advantage of LP solvers, which is warm up starts. Warm up starts mean that instead of starting all over again after adding cutting planes, an optimal solution of a node relaxation can be used to find the new optimal solution of $\mathcal{Q}_{relax}$. This could save a lot unnecessary iterations and helps improves the dual bound $\ddot{c}_{dual}$ in the BnB algorithm leading to the pruning of large sections of the BnB tree.

Figure 5.3 illustrates what the insertion of a cutting plane into an LP relaxation does. This illustration assumes two binary variables whose axes are horizontal and vertical besides one continuous variable whose axis is perpendicular to the page. Hence, what we see in the figure is a cross-section of the feasible region where the black dots are actually parallel lines emanating from the page. Note that in the figure, the bilinear terms of the non-convex quadratic constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$ are considered to be linearly relaxed by McCormick under-estimators. The blue region represents the convex hull of the discrete feasible solutions to a small problem instance. A convex hull $clcnv\,(.)$ is the smallest convex set that includes the set of all feasible solutions. The union of the green area and the blue represents the relaxation of the subproblem $\mathcal{Q}_{relax}$ and its optimal solution $\ddot{\mathbf{x}}^{opt}_{relax}$. The purpose of the cutting plane is to tighten the relaxation by cutting into the feasible region of $\ddot{\mathbf{x}}^{opt}_{relax}$ as deep as possible without passing into the $clcnv\,(\mathcal{Q})$ which contains the feasible solutions of $\mathcal{Q}$.

Cutting plane separation is performed in rounds, where in each round, many different cutting planes are generated to cut off $\ddot{\mathbf{x}}^{opt}_{relax}$. All the cuts are stored in a *separating storage* from which only a subset of those are selected for cutoff. Just as in [65], the cuts are selected based upon the following:

- the *efficacy* $\overline{eff}_c$ of the cut, which is the Euclidean distance between the cut and $\ddot{\mathbf{x}}^{opt}_{relax}$,

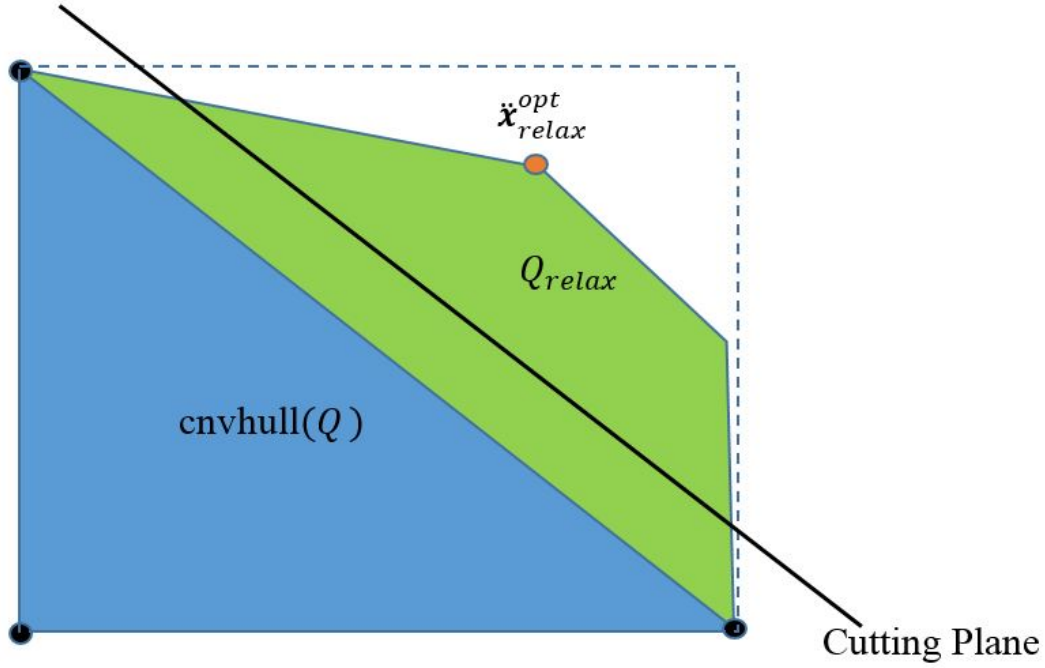



Figure 5.3: Illustration of a cutting plane that separates the optimal solution for $\mathcal{Q}_{relax}$ (represented by the red dot) from the convex hull of $\mathcal{Q}$ (represented by the blue triangle)

- the *orthogonality* $\overline{orth}_c$ of the cuts with respect to each other and
- the *parallelism* $\overline{par}_c$ of the cuts with respect to the objective function.

A score $s_{cut}\left(\overline{eff}_c, \overline{orth}_c, \overline{par}_c\right)$ based upon the weighted sum of these criteria is given by [65]:

$$s_{cut}\left(\overline{eff}_c, \overline{orth}_c, \overline{par}_c\right) = w_{eff}\overline{eff}_c + w_{orth}\overline{orth}_c + w_{par}\overline{par}_c. \tag{5.13}$$

Algorithm 2 explains in detailed steps how cutting planes selection is performed in the SCIP solver, which we use for $\mathcal{HAP}^{Eff}_{MBQCP}$.

If the solution of a relaxed subproblem $\ddot{\mathbf{x}}^{opt}_{relax}$ violates any of the quadratic constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$, which were all proved to be non-convex in Chapter 4, each bilinear term gets underestimated separately. For a term with a positive coefficient

**Algorithm 2** Cutting Plane Selection [65]

**Input**:
The LP solution $\ddot{\mathbf{x}}^{opt}_{relax}$ of the relaxation $\mathcal{Q}_{relax}$,
Set of generated cutting planes $\overline{\mathcal{C}}$,
the score function given by equation 5.13,
minimal orthogonality **minotho** $\in [0, 1]$,
maximal number of cuts **maxcutnum** per round.
**Output**:
: Updated LP relaxation.
Initialize the set of cuts that are to be added to the LP relaxation:
: $\tilde{\mathcal{C}}_{LP} \leftarrow \emptyset$
**Calculate Initial Score**:
For the cuts satisfying $\mathbf{a}_c^T \ddot{\mathbf{x}}^{opt}_{relax} \leq \overline{b}_c \ \forall \ c \in \overline{\mathcal{C}}$ calculate:

i Calculate and assign the efficacy value of cut $c$

$$\overline{eff}_c \leftarrow \left(\mathbf{a}_c^T \ddot{\mathbf{x}}^{opt}_{relax}\right) / \left\|\mathbf{a}_c^T\right\|$$

ii Calculate and assign the parallelism value of cut $c$

$$\overline{par}_c \leftarrow \left|\mathbf{a}_c^T \overline{\mathbf{c}}\right| / \left(\left\|\mathbf{a}_c^T\right\| \left\|\mathbf{a}_c^T . \overline{\mathbf{c}}\right\|\right)$$

where $\overline{\mathbf{c}}$ is the vector objective function coefficients

iii Initialize the orthogonality of cut $c$ as

$$orth_c \leftarrow 1$$

iv The initial score:

$$s_c \leftarrow s_{cut}\left(\overline{eff}_c, \overline{orth}_c, \overline{par}_c\right)$$

**while** $\overline{\mathcal{C}} \neq \emptyset$ **and** $\left|\tilde{\mathcal{C}}_{LP}\right| <$ **maxcutnum do**
  $c^* \leftarrow arg\left(\underset{c}{max}\ s_c\right)$
  $\tilde{\mathcal{C}}_{LP} \leftarrow c^*$
  $\mathcal{Q}_{relax} \leftarrow c^*$
  $\overline{\mathcal{C}} \leftarrow \overline{\mathcal{C}} \setminus \{c^*\}$
**end while**
**for** $c$ **to** $\left|\overline{\mathcal{C}}\right|$ **do**

$$\overline{orth}_c \leftarrow min\left\{\overline{orth}_c, 1 - \left|\mathbf{a}_{c^*}^T \mathbf{a}_c\right| / \left(\left\|\mathbf{a}_{c^*}^T\right\| \left\|\mathbf{a}_c\right\|\right)\right\}$$

  **if** $\overline{orth}_c <$ **minotho then**
    $\overline{\mathcal{C}} \leftarrow \overline{\mathcal{C}} \setminus \{c\}$
  **else**
    $s_c \leftarrow s_{cut}\left(\overline{eff}_c, \overline{orth}_c, \overline{par}_c\right)$
  **end if**
**end for**

$a$, the McCormick underestimators are given as [23]:

$$a\ddot{x}_j\ddot{x}_k \geq a\ddot{x}_j^L\ddot{x}_k + a\ddot{x}_k^L\ddot{x}_j - a\ddot{x}_j^L\ddot{x}_k^L, \tag{5.14a}$$

$$a\ddot{x}_j\ddot{x}_k \geq a\ddot{x}_j^U\ddot{x}_k + a\ddot{x}_k^U\ddot{x}_j - a\ddot{x}_j^U\ddot{x}_k^U, \tag{5.14b}$$

where $\ddot{x}_j^L$ and $\ddot{x}_j^U$ are the lower and upper bounds of $\ddot{x}_j$ respectively. If $\left(\ddot{x}_j^U - \ddot{x}_j^L\right)\ddot{x}_{relax_k}^{opt} + \left(\ddot{x}_k^U - \ddot{x}_k^L\right)\ddot{x}_{relax_j}^{opt} \leq \ddot{x}_j^U\ddot{x}_k^U - \ddot{x}_j^L\ddot{x}_k^L$ inequality 5.14a is used, otherwise inequality 5.14b is used. If the bilinear term coefficient $a$ is negative, the McCormick underestimators are:

$$a\ddot{x}_j\ddot{x}_k \geq a\ddot{x}_j^U\ddot{x}_k + a\ddot{x}_k^L\ddot{x}_j - a\ddot{x}_j^U\ddot{x}_k^L, \tag{5.15a}$$

$$a\ddot{x}_j\ddot{x}_k \geq a\ddot{x}_j^L\ddot{x}_k + a\ddot{x}_k^U\ddot{x}_j - a\ddot{x}_j^L\ddot{x}_k^U, \tag{5.15b}$$

If $\left(\ddot{x}_j^U - \ddot{x}_j^L\right)\ddot{x}_{relax_k}^{opt} - \left(\ddot{x}_k^U - \ddot{x}_k^L\right)\ddot{x}_{relax_j}^{opt} \leq \ddot{x}_j^U\ddot{x}_k^L - \ddot{x}_j^L\ddot{x}_k^U$ inequality 5.15a is used, otherwise inequality 5.15b is used.

Besides the McCormick separators, we use implied cuts, clique cuts [65] as well as the generic mixed integer Gomory cuts [69] for $\mathcal{HAP}^{Eff}_{MBQCP}$. For implied bound cuts, the separator inspects the implication graph to extract cuts that are violated by $\ddot{\mathbf{x}}^{opt}_{relax}$. These implications can only be violated if a binary variable has a fractional value in $\ddot{\mathbf{x}}^{opt}_{relax}$. The implication graph is a directed graph whose purpose is to store the strongest implications between variables in $\mathcal{HAP}^{Eff}_{MBQCP}$. An implication is simply a derivation of the values of other variables if a variable $\ddot{x}_j$ is set to a specific value (or range of values). For example, if we consider constraint set $\overline{C5}$ in formulation $\mathcal{HAP}^{Eff}_{MBQCP}$, we can see that if $y_{m,i,c,t} = 0$ this implies $p_{m,i,c,t} = 0$. A second example is $\overline{C9}$ which implies $\theta_m = 0$ when all the corresponding $y_{m,i,c,t}$ in the constraint are

equal to zero. Many similar implications are deduced in the presolving stage from which the implication graph is constructed.

For clique cuts, a clique graph that stores sets of binary variables where for each set, only one (complemented) binary variable can be set to one (zero) while the rest of the (complemented) variables in the set should be set to zero (one). A clique inequality has the form

$$\sum_{Q_{binary}} \ddot{x}_j \leq 1, \tag{5.16}$$

where $Q_{binary} \subset \mathcal{B} \cup \overline{\mathcal{B}}$ is a subset of binary variables and their complements. In $\mathcal{HAP}^{Eff}_{MBQCP}$, some cliques could be obtained directly from the set packing constraint set $\overline{C2}$ which are cliques by themselves. Presolving could further deduce more cliques while simplifying other constraints to the point that they could be upgraded to set packing constraints or using probing [70]. The LP values of binary variables are used as weights for the nodes of the clique graph for the separation of violated cliques using *TClique* [71] algorithm in SCIP solver.

## 5.5 Domain Propagation

After a node in the BnB tree gets selected for processing and primal heuristic methods have been called, domain propagation methods are called for different constraint types to tighten the variable's local domains, or if possible fix their values. This is performed iteratively in rounds until no more domain reductions are possible or until a maximum number of propagation rounds is hit. The inferred domain deductions can yield stronger linear underestimators in the separation process (for quadratic constraints), they can cutoff nodes due to infeasibility of a constraint and can result in further domain deductions on other constraints. There are different propagation methods in

SCIP that could be tailored for the different types of constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$ as explained in the following subsections.

### 5.5.1 Domain Propagation Schemes for Quadratic Constraints

For the quadratic constraints, SCIP uses an interval-arithmetic based method [72]. Forward and backward propagations are invoked for a quadratic constraint which could be rewritten as [23], :

$$\sum_{j\in\tilde{J}} d_j\ddot{x}_j + \sum k \in \tilde{K} \left( e_k + p_{k,k}x_k + \sum_{r\in\tilde{K}} p_{k,r}x_k \in [l,u] \right), \tag{5.17}$$

where

- $l, u \in \overline{\mathbb{Q}}$ where $\overline{\mathbb{Q}}$ is the set of rational numbers and $\pm\infty$
- $\tilde{J} \cup \tilde{K} \subseteq N$ where the set $N$ is the set of all variables in the problem,
- $\tilde{J} \cap \tilde{K} = \emptyset$
- $p_{k,r} = 0$ for $k > r$.

Forward propagation is a step that is done to reduce the constraint interval $[l_c, u_c]$ which, using simple interval arithmetic, yields the new reduced interval for the constraint $[l_c^{new}, u_c^{new}]$ with respect to the current variable domains. If the intersection between the new constraint interval and the old constraint is empty, i.e. $[l_c, u_c] \cap [l_c^{new}, u_c^{new}] = \emptyset$ , the BnB node $\mathcal{Q}$ could be pruned, otherwise the constraint domain could be reduced to $[l_c, u_c] \cap [l_c^{new}, u_c^{new}]$.

Backward constraint propagation infers domain deductions on the variables in a quadratic constraint given the constraint interval $[l_c, u_c]$. This is achieved by solving

the quadratic interval equation for the quadratic constraint and its interval whose solutions for all its bilinear (or quadratic) variables are the intervals $\left[l^{\ddot{x}_j}, u^{\ddot{x}_j}\right]$. If the the intersection between the old variable bounds and the new inferred variable intervals is empty, i.e. $\left[l^{\ddot{x}_j}, u^{\ddot{x}_j}\right] \cap \left[\ddot{x}_j^L, \ddot{x}_j^U\right] = \emptyset$, then the BnB node $\mathcal{Q}$ can be pruned, otherwise the new variable bounds inferred are $\left[l^{\ddot{x}_j}, u^{\ddot{x}_j}\right] \cap \left[\ddot{x}_j^L, \ddot{x}_j^U\right]$.

### *5.5.2 Domain Propagation Schemes for Linear Constraints*

In the formulation $\mathcal{HAP}^{Eff}_{MBQCP}$ constraint sets $\overline{C1} - \overline{C7}$ and $\overline{C9} - \overline{C10}$ are all linear sets. Constraint sets $\overline{C2}$ and $\overline{C5}$ are explicitly set packing and variable bound constraints respectively. Furthermore, during the presolving phase, many linear constraints in a given instance of $\mathcal{HAP}^{Eff}_{MBQCP}$ get reduced to set packing, variable bound and set covering constraints. For example, one of the instances passed to SCIP, in the experiments conducted had 256 linear constraints and 2844 quadratic constraints. After the presolving phase, they were reduced to 2004 quadratic constraints at the expense of 5882 general linear constraints, 146 set packing and covering constraints and 2604 variable bound constraints.

For general linear constraints taking the form $\mathbf{a}^T\ddot{\mathbf{x}} \leq \overline{\beta}$, variable bound propagation uses the concept of *activity bounds* defined as:

$$\underline{\alpha} = min\left\{\mathbf{a}^T\ddot{\mathbf{x}} | \ddot{\mathbf{x}}^{\tilde{L}} \leq \ddot{\mathbf{x}} \leq \ddot{\mathbf{x}}^{\tilde{U}}\right\} \qquad and \qquad \overline{\alpha} = max\left\{\mathbf{a}^T\ddot{\mathbf{x}} | \ddot{\mathbf{x}}^{\tilde{L}} \leq \ddot{\mathbf{x}} \leq \ddot{\mathbf{x}}^{\tilde{U}}\right\}. \quad (5.18)$$

where

- $\underline{\alpha}$ and $\overline{\alpha}$ are the minimal and maximal activities of the constraint.
- $\ddot{\mathbf{x}}^{\tilde{L}}$ and $\ddot{\mathbf{x}}^{\tilde{U}}$ are the local (to the node) lower and upper variable bounds for the variable vector $\ddot{\mathbf{x}}$.

The lower and upper *activity bound residuals* $\underline{\alpha}_j$ and $\overline{\alpha}_j$ respectively for each variable are:

$$\underline{\alpha}_j = min\left\{\mathbf{a}^T\ddot{\mathbf{x}} - a_j\ddot{x}_j | \ddot{\mathbf{x}}^{\tilde{L}} \leq \ddot{\mathbf{x}} \leq \ddot{\mathbf{x}}^{\tilde{U}}\right\} \qquad and \qquad \overline{\alpha}_j = max\left\{\mathbf{a}^T\ddot{\mathbf{x}} - a_j\ddot{x}_j | \ddot{x}^{\tilde{L}} \leq \ddot{\mathbf{x}}^{\tilde{L}} \leq \ddot{\mathbf{x}}^{\tilde{U}}\right\}. \tag{5.19}$$

The propagations are then based upon the following:

1. If $\overline{\alpha} \leq \overline{\beta}$ then the constraint is redundant and can be removed.

2. If $\underline{\alpha} \geq \overline{\beta}$ then the constraint cannot be satisfied within the node's local bounds and the node's subproblem becomes infeasible.

3. For all the binary and continuous variables in the vector $\ddot{\mathbf{x}}$ of $\mathcal{HAP}^{Eff}_{MBQCP}$ and any variables or constraints added to it in the presolving phase, the local variable bounds are calculated as [65]:

$$\ddot{x}_j \leq \frac{\overline{\beta} - \underline{\alpha}_j}{a_j},\ if\ a_j > 0 \qquad or \qquad \ddot{x}_j \geq \frac{\overline{\beta} - \underline{\alpha}_j}{a_j},\ if\ a_j < 0 \tag{5.20}$$

   If the variable is binary then a lower bound is rounded up and an upper bound is rounded down which could fix the binary variable to one of its two possible values.

## Set Packing Constraints

Any set packing constraint (including $\overline{C2}$) taking the form of Equation (5.16) and has only one possibility of domain propagation, that is:

$$\ddot{x}_k = 1 \qquad \rightarrow \forall j \in Q_{binary} \setminus \{k\} : \ddot{x}_j = 0 \tag{5.21}$$

For this type of constraint, if two or more variables in the set $Q_{binary}$ are fixed to one, the subproblem $\mathcal{Q}$ is then infeasible and can be pruned. If exactly one variable in the set $Q_{binary}$ is fixed to one, then the rest of the variables in the set must be zero. In this case the constraint gets deleted from the subproblem. This is achieved by keeping a track and updating a counter of the current number of variables fixed to 1 in the constraints in $\overline{C2}$.

### Set Covering Constraints

During the course of the BnB algorithm, in each time a pair of subproblems are created, a binary variable is either fixed to one or to zero. If any of the binary variables $\theta_m$ in $\mathcal{HAP}^{Eff}_{MBQCP}$ is fixed to 1 in any node in the tree, the corresponding constraint in the constraint set $\overline{C3}$ becomes a set covering constraint if none of the variables $y_{m,i,c,t}$ in that constraint was fixed to one. If any of the variables $y_{m,i,c,t}$ in $\overline{C3}$ were also fixed to one, then the constraint becomes redundant and is just dropped. If the constraint becomes a set covering constraint in a BnB node, domain propagation techniques that are specific to set covering constraints are applied.

A set covering constraint is one that takes the form

$$\sum_{Q_{binary}} \ddot{x}_j \geq 1. \tag{5.22}$$

This has the same propagation rule as the set packing constraints' propagation rule. SCIP handles propagation for set covering constraints differently though, with a tailored more efficient way that can handle large numbers of set covering constraints using the fact that a set covering constraint is equivalent to a clause $l_1 \vee l_2 \vee ... \vee \left|l_{Q_{binary}}\right|$. To propagate the clause, SCIP uses the method of *two watched literals* [73]. It uses

the main observation that implications are derived from clauses only if all but one literal in a clause is fixed to zero. Therefore, a clause is considered for propagation only if the number of literals fixed to zero increases from $\left|l_{Q_{binary}} - 2\right|$ to $\left|l_{Q_{binary}} - 1\right|$. The remaining unfixed literal in this case is implied and fixed to 1.

It is sufficient to watch the state of only two arbitrary literals of the constraint, as long as both remain unfixed, the constraint does not need to be processed as no propagation can be applied. If one literal is zero, the other literals of the clause are inspected. If at least one of them is fixed to 1, the constraint is then deduced to be redundant and is removed from the node's subproblem. If at least one literal in the clause is unfixed, then watching the fixed literal stops and the unfixed one gets watched instead. Finally, if all the literals except the other watched literal are fixed to zero, the other watched literal becomes implied and fixed to one.

## Variable Bound Constraints

Variable bound constraints are generally defined as:

$$\underline{\beta} \leq \ddot{x}_i + a_j \ddot{x}_j \leq \overline{\beta} \quad i, j : i \neq j, \tag{5.23}$$

with $\ddot{x}_j \in \mathbb{Z}$, $a_j \in \mathbb{R}$ and $\underline{\beta}, \overline{\beta} \in \mathbb{R} \cup \pm\infty$. A special case of it is the variable upper bound constraint that takes the form

$$\ddot{x}_i \leq u_i^{'} \ddot{x}_j, \quad i, j : i \neq j, \tag{5.24}$$

where $u_i^{'}$ is the local upper bound of the variable $\ddot{x}_i$ and $\ddot{x}_j$ is binary. Constraint set $\overline{C5}$ in formulation $\mathcal{HAP}^{Eff}_{MBQCP}$ belongs to the category of variable upper bound constraints and the domain propagation of the variables in that constraint set, namely

$y_{m,i,c,t}$ and $p_{m,i,c,t}$ in any of BnB nodes follow the simple rules:

1. fix $p_{m,i,c,t}$ to zero if $y_{m,i,c,t} = 0$, and
2. fix $y_{m,i,c,t}$ to 1 if $p_{m,i,c,t} \neq 0$.

## Problem Specific Constraint Propagation for $\mathcal{HAP}^{Eff}_{MBQCP}$

The simple constraint set $\overline{C1}$ enables the propagation of many reductions if the binary constant $\lambda_{m,k}$ is zero for some of the constraints in the set. For these constraints the corresponding variables $\phi_{m,k}$ are immediately deduced to be zeros and their corresponding constraints dropped. The $\overline{C1}$ constraints for the remaining variables $\phi_{m,k}$ for which the corresponding binary constant is $\lambda_{m,k} = 1$, also get dropped because they become redundant implying that the binary variables $\phi_{m,k} \leq 1$. This procedure propagates further reductions in the root node and possibly other nodes. The propagations would lead to the following:

1. All the constraints in $\overline{C3}, \overline{C8}, \overline{Q2}, \overline{Q4}$ in which $\phi_{m,k} = 0$ become redundant and can be removed.
2. If for any $m$, there are fixed variables $\phi_{m,k} = 0\ \forall k$ in a BnB node subproblem, all the variables $y_{m,i,c,t}$ for that given $m$ then consequently get fixed to zero due to constraint set $\overline{C4}$. This then leads to,
3. fixing $\theta_m$ to zero according to constraint set $\overline{C10}$.

## Objective Function Propagation

This type of propagation infers bounds on decision variables that are valid since taking a value for the variable outside its tightened bounds will not lead to a better incumbent

(primal) solution. Taking the objective function of formulation $\mathcal{HAP}^{Eff}_{MBQCP}$, which is always guaranteed to be integer, and does not include any continuous variables, then inferior solutions are ruled out using:

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\rho_{m,k}\phi_{m,k} \geq \ddot{c}_{primal} - \left(1-\hat{\delta}\right), \tag{5.25}$$

where $\hat{\delta} \in \mathbb{R}^{+}$ is the optimality tolerance. This objective constraint can be easily transformed to a binary knapsack constraint by:

- multiplying the constraint by -1,
- complementing the variables with negative coefficients by $\overline{\phi}_{m,k} = 1 - \phi_{m,k}$, and
- rounding down the resulting right hand side constant.

A computationally efficient knapsack propagator available in SCIP (explained in details in [65]), is used for our problem to reduce the variable bounds and propagate further reductions for variables that belong to the objective function in $\mathcal{HAP}^{Eff}_{MBQCP}$, in this case $\phi_{m,k},\ \forall\ m,k$ only.

## Reduced Costs Propagation

This type of propagator tightens the bounds of a decision variable in the problem by comparing its reduced cost $r^{j}_{relax}$ values in the current node's LP relaxation solution (i.e. $\ddot{\mathbf{x}}^{opt}_{relax}$) with the objective function value $\ddot{c}_{primal}$ of the incumbent solution $\ddot{\mathbf{x}}_{BFS}$ and the objective value $\ddot{c}_{dual}$ of the LP relaxation. The new bounds on the decision variables are then obtained by:

$$\ddot{x}_j \geq \ddot{x}^{\tilde{L}}_{j} + \frac{\ddot{c}_{dual} - \ddot{c}_{primal}}{r^{j}_{relax}},\ if\ r^{j}_{relax} > 0, \tag{5.26a}$$

$$\ddot{x}_j \geq \ddot{x}_j^{\tilde{U}} + \frac{\ddot{c}_{dual} - \ddot{c}_{primal}}{r^j_{relax}},\ if\ r^j_{relax} < 0, \tag{5.26b}$$

where $\ddot{x}_j^{\tilde{L}}$ and $\ddot{x}_j^{\tilde{U}}$ are the local lower and upper bounds of the variable $\ddot{x}_j$ respectively.

### Dual Fixing Propagator

This propagator fixes variables, that have no restrictions in direction of their objective coefficient, to the best possible value. If the objective coefficient of a variable is 0 and it may be rounded both up and down, then this variable will be fixed to the closest feasible value to 1 (in a maximization problem).

## 5.6 Heuristics

Heuristics are known to be quick in terms of obtaining solutions for an optimization problem, however they are not guaranteed to obtain a solution , least of all an optimal solution. The branch-and-bound algorithm for solving an MBQCP requires too much computational effort that grows exponentially with the size of the problem, however it is a guaranteed method in obtaining good feasible solutions, including the optimal solution, in a finite amount of time. Therefore combining heuristics with the BnB algorithm is expected to improve the overall performance in solving the problem formulation $\mathcal{HAP}^{Eff}_{MBQCP}$. The advantages are mainly:

1. Early feasible solutions found by heuristics help prune the search tree early which reduces the search time in the BnB algorithm.

2. In case of moderate-to-quick changes in the radio environment, the parameters $g_{i,k,c,t}$ in $\mathcal{HAP}^{Eff}_{MBQCP}$ could change their values within a time interval that is shorter than the time needed to obtain the optimal solution. In such a situation, even the first feasible solution may be acceptable.

There are a number of heuristics that were tried in the experiments conducted for $\mathcal{HAP}^{Eff}_{MBQCP}$. The heuristics were called in the orders of their simplicity, i.e. the most computationally simple ones were called first. The heuristics phase is terminated if a feasible solution is found. Otherwise, if a heuristic algorithm could not find a feasible solution, a more computationally involved heuristic with better chances of finding feasible solutions is invoked. Furthermore, the simple heuristics are called in all nodes while the more complex ones are called at lower frequencies as the BnB grows deeper. There are two classification ways to categorize the heuristics used. One way to classify them is to divide them into the following two categories:

- *independent heuristics*: which do not require the presence of any previously found feasible solutions in the BnB tree,
- *improvement heuristics*: which require one or more feasible solutions that were found earlier in the BnB tree or by an *independent heuristic*. It uses the feasible solutions found earlier to deduce better feasible solutions.

Another way to classify the heuristics used is by dividing them into *mixed integer linear program* (MILP) heuristics and *mixed integer non-linear program* (MINLP) heuristics. The MIP heuristics could be used for the MBQCP problem $\mathcal{HAP}^{Eff}_{MBQCP}$ since MBQCP is relaxed to a MBLP using linear outer approximations. In the following subsections, the types of heuristics used in the solving procedure for $\mathcal{HAP}^{Eff}_{MBQCP}$ are mentioned, and a brief explanation for each is provided.

### *5.6.1 Simple Rounding*

This heuristic works simply by rounding a binary variable with a fractional value to either zero or one such that no linear constraints, including the bilinear terms'

underestimators, are violated. This uses the concept of *variable locks* in [65] $\zeta_j^0$ and $\zeta_j^1$ for every binary variable $j \in \mathcal{B}$ which are the numbers of constraints that block a variable's up-rounding or down-rounding respectively. This heuristic can have a chance of finding a feasible solution only when the violated constraints are the binary constraints on the binary variables while the quadratic constraints are satisfied and is hence an MIP heuristic. It is also an *independent heuristic* since it does not need an input of any earlier feasible solutions. It is a very simple, and hence very fast, heuristic and is therefore the first heuristic invoked in the solution procedure to try to find a feasible solution.

### *5.6.2 Rounding*

This type of heuristic is an extension to the *simple rounding* heuristic. It applies rounding to all fractional variables of binary nature even if this leads to infeasibility to the linear constraints. The infeasibility is then resolved by trying to select the rounding of the next variable such that the infeasibility is reduced or eliminated. The heuristic iterates over the set of fractional variables and if $\ddot{\mathbf{x}}^{opt}_{relax}$ is feasible to all linear constraints, including bilinear underestimators to $\mathcal{HAP}^{Eff}_{MBQCP}$, the fractional variable of binary nature having the highest number of variable locks ($max\left\{\zeta_j^0, \zeta_j^1\right\}$) is rounded to the most feasible direction. If $\ddot{\mathbf{x}}^{opt}_{relax}$ violates one or more constraints, one of these is selected and the algorithm searches for a fractional variable of binary nature to be rounded to a direction that decreases the infeasibility of the constraint.

The *rounding* heuristic is an MIP heuristic which could succeed as long as all quadratic constraints are satisfied in $\mathcal{Q}_{relax}$. Also, it is an *independent heuristic* since it does not need as input any feasible solutions found earlier in the BnB tree. The set of possible feasible solutions found by the *rounding* heuristic is a superset of those

for *simple rounding* heuristic. Therefore if no feasible solutions are found by *simple rounding* the more general *rounding* heuristic is invoked.

### 5.6.3 Integer Shifting

This heuristic relaxes all linear constraints by removing the continuous variables. This is done in all linear constraints, $\mathbf{a}^T \ddot{\mathbf{x}}^{opt}_{relax} \leq \overline{\beta}$, by subtracting the variable's contribution $a_j \ddot{x}^{opt}_{relax_j}$ from both sides. After that, the *integer shifting* heuristic technique continues in a similar way as the *rounding* heuristic but tries to continue in the case no rounding can decrease the violation of an infeasible linear constraint. The value of continuous variables is shifted, or inverted for binary variables, to decrease the violation of the constraint. This type of heuristic is an *independent* and MIP heuristic.

### 5.6.4 Pseudocost Diving

*Pseudocost diving* belongs to a group of heuristics known as *diving heuristics* [65]. Diving heuristics, have a generic framework whose main elements perform the following:

1. Choose a variable in the set $\mathcal{FRAC}$ of fractional variables of binary type and a rounding direction.

2. Fix the binary variable to the value corresponding to its rounding direction.

3. Call domain propagation to propagate the fixed variable.

4. Resolve the relaxation $\mathcal{Q}_{relax}$ after making the necessary changes that result from fixing the chosen variable.

5. If $\mathcal{Q}_{relax}$ has no feasible solution then stop with failure. Otherwise, using the obtained solution, repeat the entire procedure all over. These repetitions can continue until $\mathcal{FRAC} = \emptyset$, or an iteration limit is reached.

Different diving heuristics differ in the way the first step is performed. In a *pseudocost diving* heuristic, the pseudocost values $\Psi_j^0$ and $\Psi_j^1$ are used for the variable selection and deciding the rounding direction. First, a decision on the rounding direction for all the variables in $\mathcal{FRAC}$ for the solution $\ddot{\mathbf{x}}^{opt}_{relax}$ of $\mathcal{Q}_{relax}$ is obtained by performing the following steps:

1. The values of each of the variables $\ddot{x}^{opt}_{relax_j} : j \in \mathcal{FRAC}$ are compared with their counterpart $\ddot{x}^{\mathcal{R}}_{relax_j}$ in the root node $\mathcal{R}$. If the difference between $\ddot{x}^{opt}_{relax_j}$ and $\ddot{x}^{\mathcal{R}}_{relax_j}$ indicates the variable is being pushed to a certain direction, then that direction is selected for rounding.

2. Otherwise, if the difference does not yield a rounding decision, if the fractional part of the variable is less than 0.3 then it is rounded downwards, while if it is greater than 0.7 it gets rounded upwards.

3. If still a rounding decision is not made, then a variable in $\mathcal{FRAC}$ is rounded to 0 if $\Psi_j^0 < \Psi_j^1$ and 1 otherwise.

Finally, the variable in $\mathcal{FRAC}$ that maximizes:

$$\sqrt{1 - \ddot{x}^{opt}_{relax_j}}.\frac{1+\Psi_j^1}{1+\Psi_j^0}\ (downwards) \qquad or \qquad \sqrt{\ddot{x}^{opt}_{relax_j}}.\frac{1+\Psi_j^0}{1+\Psi_j^1}\ (upwards). \quad (5.27)$$

is chosen for rounding [65].

This type of heuristic is classified as a *independent heuristic* since it does not need, as an input, any feasible solutions obtained earlier and is under the category of MIP heuristics.

### 5.6.5 Feasibility Pump

This scheme was originally created by Fischetti, Glover and Lodi [74] for integer programs, generalized to MIPs by, Bertacco, Glover and Lodi [75] and slightly improved by Achterberg and Berthold [76]. It starts with the optimal solution of the relaxed subproblem $\mathcal{Q}_{relax}$, the variables that are of binary nature and have fractional solutions are rounded as:

$$\ddot{x}^{round_j}_{relax} = \left\lfloor \ddot{x}^{opt}_{relax_j} + 0.5 \right\rfloor \ \forall\ j \in \mathcal{FRAC}. \tag{5.28}$$

If the rounded solution is not feasible, another LP is solved to find a new point on the same LP polyhedron of subproblem $\mathcal{Q}_{relax}$. The objective function for the feasibility is to minimize the hamming distance:

$$\mathcal{H}_{distance} = \sum_{j \in \mathcal{FRAC}} \left| \ddot{x}^{opt}_{relax_j} - \ddot{x}^{round_j}_{relax} \right|. \tag{5.29}$$

This is repeated iteratively until a feasible solution is found or an iteration limit is reached. The *feasibility pump* heuristic is an *independent* MIP heuristic. It tries to resolve the infeasibilities that result from fractional values of variables that are of binary type.

### 5.6.6 Clique Partition based Large Neighborhood Search Heuristic

*Large neighborhood search* (LNS) heuristics [77] restrict the search for good feasible solutions to a neighborhood of a certain reference point. This restriction makes a node $\mathcal{Q}$ subproblem easier to solve with a better chance of finding a high quality feasible solution. The restricted subproblems do not need to be solved to optimality, but to a solution that is better than the current incumbent. The neighborhood is

defined by introducing additional constraints to the node's subproblem which are usually variable fixings. The success of an LNS is strongly tied to defining a good neighborhood. The main characteristics of a good neighborhood are:

1. it should have high quality solutions,
2. these solutions should be easy to discover,
3. the neighborhood should be easy to process.

The main elements of the *clique partition based LNS* are

1. First, the clique table constructed during the presolving phase is used to fix a subset of the binary variables in $\mathcal{HAP}^{Eff}_{MBQCP}$ [78]. The only type of binary variables in the objective function is the set of variables $\phi_{m,k}\ \forall\ m,k$. If any of those within the clique was not already fixed to 1, then the one with the largest coefficient is chosen to be fixed to 1. If the binary variable is not one of the $\phi_{m,k}$ variables, the selection of which variable in the clique to fix to 1 is random since their coefficients in the objective function is zero. The rest of the variables in a clique are fixed to zero using two rounds of domain propagation. The processes iterates until all binary variables are fixed.
2. If the domain propagation in any iteration detects infeasibility, one level of backtracking to undo the previous fixing and its corresponding domain propagation fixings is performed.
3. After a sufficient number of fixations (which is determined by a preset threshold), the resulting reduced (hopefully easier) LP is solved and an attempt is made to round the resulting solution to a feasible solution using the *simple rounding* heuristic.

4. The LNS is invoked again with a neighborhood defined by the fixations obtained from the last phase and a sub-MBQCP for that neighborhood is created and solved (not necessarily to optimality). If a feasible solution is found it gets returned and the heuristic terminates.

### *5.6.7 Relaxation Enforced Neighborhood Search (RENS)*

The basic idea for the RENS heuristic when used in a BnB algorithm to obtain feasible solutions for an MBQCP is very simple. Any binary variables in the solution $\ddot{\mathbf{x}}^{opt}_{relax}$ of the node relaxation $\mathcal{Q}_{relax}$ that have an integer value get fixed to these values [79]. The resulting problem is hence smaller sub-MBQCP whose linear outer approximation is obtained using McCormick underestimators and is solved as an MBLP. RENS is only called when the resulting sub-MBQCP in a node due to fixations is much easier than the original one. For $\mathcal{HAP}^{Eff}_{MBQCP}$, this means that at least a specific ratio of all binary variables in the node has to be fixed. Moreover, the sub-MBQCP does not have to be solved to optimality, hence the stopping criteria can be an acceptably good feasible solution, maximum number of nodes or maximum number of LP iterations. Finally, since the heuristic's purpose is to find a good primal solution quickly, then the dual bound improvement schemes like separating cuts and strong branching are disabled. RENS does not need as an input any earlier feasible solutions and is hence an *independent heuristic*.

### *5.6.8 Undercover Heuristic*

The *undercover* heuristic is suitable for any general non-convex MINLPs and obtains feasible solutions by solving sub-MIPs [80]. Therefore, it is an MINLP heuristic that is suitable for the more specific MBQCP class and solves sub-MBP for obtaining feasible

solutions. The previously mentioned heuristics deal with the integrality as the source of complexity to the problem while the *Undercover* heuristic considers the nonconvex constraints, which are quadratic in $\mathcal{HAP}^{Eff}_{MBQCP}$, as the source of complexity. By solving a vertex covering problem, it identifies a minimal set of variables to fix in order to linearize a quadratic constraint in $\mathcal{HAP}^{Eff}_{MBQCP}$ at the BnB node in which it is called. This is based on an observation that fixing certain variables can reduce the node's sub-problem to a sub-MIP, which in the case of $\mathcal{HAP}^{Eff}_{MBQCP}$ is a sub-MBLP.

A cover for a non-linear constraint is defined as the set of variables which, if fixed, linearize the constraint. A cover to the entire problem linearizes all the non-linear constraints. This may also require the fixing of continuous variables which could introduce significant errors that render the problem infeasible. Therefore the heuristic tries to minimize the number of fixed variables to obtain a large sub-MIP through obtaining minimum covers.

As in [80], the co-occurance graph consists of nodes that represent the problem's variables and edges $(i, j)$ which are present only if the Hessian matrix of some quadratic constraint has a nonzero entry for $(i, j)$. The minimum cover is obtained by solving a pure binary linear program for the co-occurance graph.

### *5.6.9 Relaxation Induced Neighborhood Search (RINS)*

*Relaxation Induced Neighborhood Search* RINS was first invented by Danna, Rothberg and LePape [81]. The fixations that yield the sub-MIP are obtained by fixing binary variables if both the incumbent and the LP solution of the node relaxation agree on a common value for a given binary variable. Since RINS needs the incumbent solution as an input, it is an *improvement heuristic*.

### *5.6.10 Crossover*

The *crossover* heuristic uses two or more feasible solutions to fix all the binary variables on which the solutions agree to their values and leaves the other variables' bounds in their global bounds [82], [83]. The larger the number of solutions that participate in the crossover the fewer the fixations are and the lager the resulting sub-MIP is. SCIP uses three solutions to define the sub-MIPs. Since the *crossover* heuristic needs at least two feasible solutions as inputs, it is an *improvement heuristic*.

In the Chapter 6, we provide the details of the experiments conducted on $\mathcal{HAP}^{Eff}_{MBQCP}$ using the different techniques explained in this chapter and present the obtained results that indicate the relative algorithmic performances.

# Chapter 6

# Numerical Experiments and Results for Solving $\mathcal{HAP}^{Eff}_{MBQCP}$

This chapter discusses the experiments conducted for $\mathcal{HAP}^{Eff}_{MBQCP}$ and presents the numerical results obtained for the algorithmic procedures given in Chapter 5 to evaluate their performances. Three different experiment sets are provided in this chapter. The first experiment set (Section 6.1) compares the performance of activating-versus-deactivating the reformulation linearizion technique, at the presolving phase, for the quadratic constraint set $\overline{C8}$ in $\mathcal{HAP}^{Eff}_{MBQCP}$ which was explained in Section 5.1. The second experiment set (Section 6.2) compares the performance of the different branching techniques explained in Section 5.3. The third experiment set (Section 6.3) compares the performances of different combinations of domain propagation, cutting planes and heuristics components in BnB. Mainly the performance for each set of experiments is measured using a subset of the following criteria:

1. the duality gap,

2. number of LP iterations expended,

3. number of nodes in the search tree,

4. average number of LP iterations per node,

5. number of instances a feasible solution was found,

6. the average time required to find the first feasible solution and

7. the objective function value.

The experiments were performed in Matlab, for which the open source optimization toolbox OPTI (version 2.16) [84] provided the interface with the SCIP 3.2.0 solver [64]. SCIP 3.2.0 is the solver used in all the experiments conducted for $\mathcal{HAP}^{Eff}_{MBQCP}$. The experiments were performed on a machine with a 6 core 3.5 GHz Intel Xeon processor. Using the parallel processing toolbox in Matlab, we were able to conduct different experiments in parallel. For example, to conduct experiments on different branching strategies, each CPU core performed the experiment of a specific branching strategy for the same set of problem instances in parallel. The generic SCIP solver settings used in all the experiment sets performed are given in Table 6.1.

Table 6.1: Generic SCIP solver settings for all experiment sets conducted.

| **Parameter** | **Value** |
|---|---|
| Solving time limit | 10 Minutes |
| LP iteration limit per node | $10^5$ iterations |
| BnB node limit | $10^7$ nodes |
| Feasibility Tolerance | $1^{-12}$ |
| Integrality Tolerance | $1^{-7}$ |

One hundred instances were solved for each experiment. Each instance has a size of 527 variables and 4261 constraints out of which 107 variables are binary and 2844 constraints are quadratic. To obtain the channel gain $g_{i,k,c,t}$ values, a simulation was conducted using equations (3.1), (3.15), (3.16) and(3.17) and the parameters

Table 6.2: Simulation parameters for HAP multicasting environment

| **Parameter** | **Value** |
|---|---|
| Number of multicasting sessions ($M$) | 2 |
| Number of antennas on board ($S$) | 7 |
| Number of users in the service area ($K$) | 10 |
| Number of available subchannels ($C$) | 3 |
| Number of available time slots ($T$) | 2 |
| HAP height | 20 Km |
| Degree of antenna beam footprint overlap | 105 % |
| HAP antenna footprint radius ($r_{footprint}$) | 500 meters |
| HAP antenna side lobe level | -40 dB |
| SINR threshold ($\gamma^{th}_{m,i}$) | 35 |
| Noise power spectral density ($N_o$) | -173 dBm/Hz |
| Maximum capacity requirements ($R^{max}_m$) | 20 Mbps |
| Minimum capacity requirements ($R^{min}_m$) | 10 Mbps |
| Carrier Frequency | 2.1 GHz |
| Total HAP Power ($P^{Total}_{PF}$ ) | 1 Watt |
| OFDMA frame length | 20 ms |
| Total Bandwidth | 15 MHz |
| Rice Factor (dB) | 20 dB |
| Rain Attenuation Factor ($\chi$) | 3 dB/Km |
| Set of values for the user-over-session priority levels ($\rho_{m,k}$) | $\rho_{m,k} \in \{1, 2, 3, 4, 5\}$ |
| The binary constants indicating the admission request of user $k$ for session $m$ ($\lambda_{m,k}$) | $\lambda_{m,k} = 1 \forall\ m, k$ |
| User antenna diameter ($D^{Ant}_{user}$) | $0.75m$ |

in Table 6.2. In the simulation, the user positions change during every iteration according to a uniform probability distribution about the HAP footprint centers. The degree of overlap between antenna footprints is defined as the ratio between the overlap distance $d_{overlap}$, illustrated in Figure 6.1, and the HAP antenna footprint radius $r_{footprint}$. Figure 4.2 illustrates the overlapped HAP antenna footprints in our experiments. To evaluate the average performance of all the instances for each experiment, arithmetic, geometric and shifted geometric means were used. For the shifted geometric mean, the shifting parameters values used are:

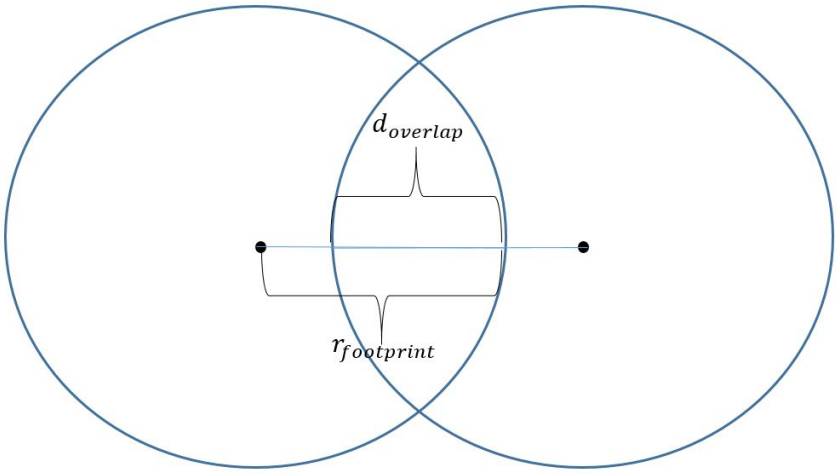



Figure 6.1: Two overlapping antenna beam footprints.

1. 15, for the time required to obtain the first feasible solution,
2. 50, for the dual gap,
3. 100, for the objective function value,
4. 100, for the number of BnB nodes,
5. 1000, for the number of LP iterations.

The shifted geometric mean of a sample $\omega_1, \omega_2, ..., \omega_k$ is given by [65]:

$$\psi_s = \left( \prod_{j=1}^{k} max\left\{\omega_j + s, 1\right\} \right)^{1/k} - s, \tag{6.1}$$

where $s$ is the shifting parameter. For geometric mean, $s = 0$. In the comments made on the results in the following sections, we use the shifted geometric means for comparison except for the average number of LP iterations per node which uses only arithmetic means.

The duality gap is calculated in all the experiments, in percentage, using the formula:

$$\varrho = \frac{|\ddot{c}_{dual} - \ddot{c}_{primal}|}{min\left(|\ddot{c}_{dual}|, |\ddot{c}_{primal}|\right)}. \tag{6.2}$$

## 6.1 Experiments and Results: Reformulation Linearization at Presolving Phase

In this section, the experimental procedures and results for the reformulation linearization technique explained in Section 5.1 are provided. The reformulation technique is invoked at the presolving phase and hence the experiments illustrate the performance of activating-versus-deactivating the linearization for the number of presolving rounds 1, 5, 25, 50 and 100. The performance evaluation criteria for the set of experiments provided in this section are:

1. the dual bound gap,
2. the total number of BnB nodes,
3. the total number of LP simplex iterations and
4. and the average number of LP iterations expended per node.

The following settings were considered for the reformulation linearization experiments:

1. node selection scheme is *best first search* with a maximum plunging depth in the BnB tree of 2 [64],
2. *most infeasible branching* scheme was used and
3. the only heuristic used was the *Undercover* heuristic.

In Figures 6.2, 6.3, 6.4 and 6.5 the duality gap, the number of LP iterations, the number of BnB nodes and the average number of LP iterations per node are illustrated for the reformulation linearization experiments. In those figures, RndNum$i$:'.', indicates the number of presolving rounds is $i$ for either: 'A' , activated reformulation linearization or 'D', deactivated reformulation linearization.

We can see that for a single presolving round, the dual gap is almost the same in both 'A' and 'D'. The number of BnB nodes is slightly lower by around 11% for 'A' compared to 'D'. The number of LP iterations and the average number of LP iterations per node are almost equal for 'A' and 'D'.

A small increase in the number of presolving rounds to 5 yields an increase in the dual gap for both cases 'A' and 'D' as shown in Figure 6.2. However, the dual gap for 'D' is lower than that of 'A' by around 30% at the expense of a much larger number of nodes in comparison to 'A' (more than 2700 %). The number of explored nodes is lower for both 'A' and 'D' for five presolving rounds as compared to one presolving round as Figure 6.4 shows. For 'A', it is lower by around 99 % and 'D' is lower by 50 % . The number of LP iterations decreases by around 85% for 'A' and increases by 15% for 'D' making it higher than 'A' by almost 800%. According to Figure 6.5, the number of average LP iterations expended per node for five presolving rounds is around 2500 % higher for 'A' compared to 'D'. Comparing five presolving rounds versus one, the average number of LP iterations per node increased by 5100 % for 'A' but only by 100% for 'D'.

Increasing the number of presolving rounds from 5 to 25 shows that the duality gap reduces for both 'A' and 'D' by 29% and 31% respectively (almost same reduction), while 'D' still has a lower dual gap by 30% compared to 'A'. The reduction in dual gap achieved by increasing the number of presolving rounds from 5 to 25 is accompanied

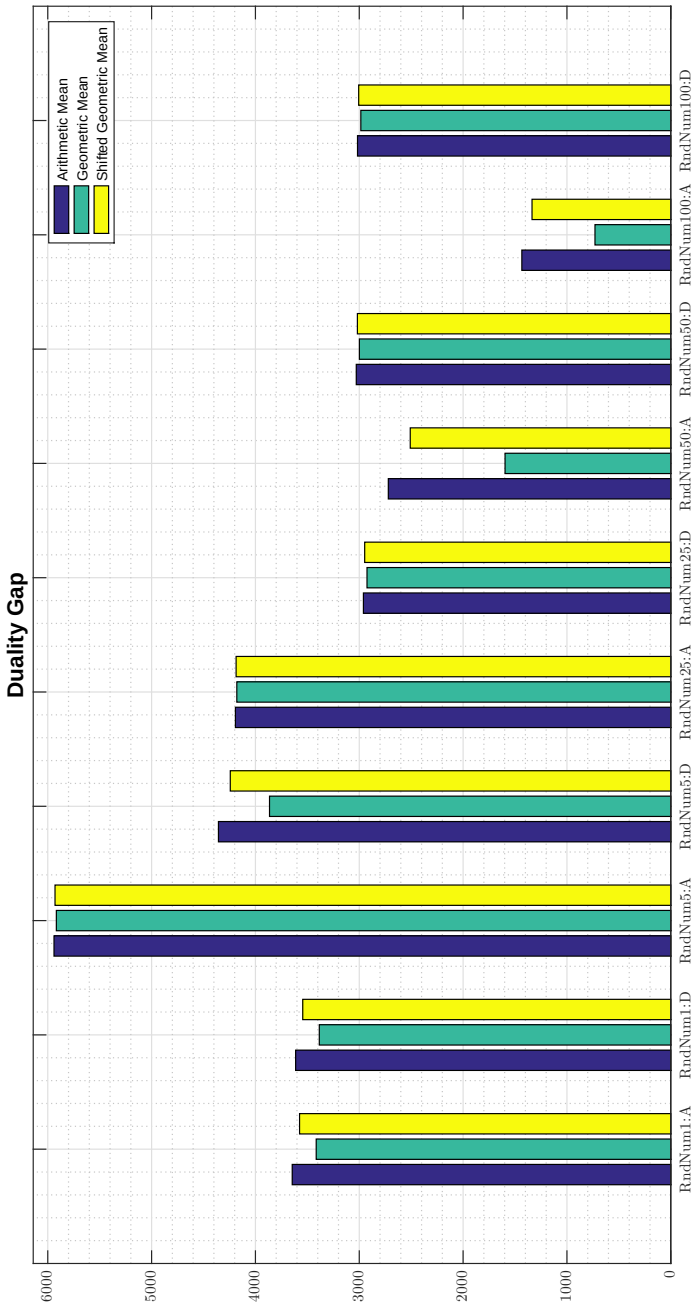



Figure 6.2: Reformulation Linearization Results: Duality Gap.

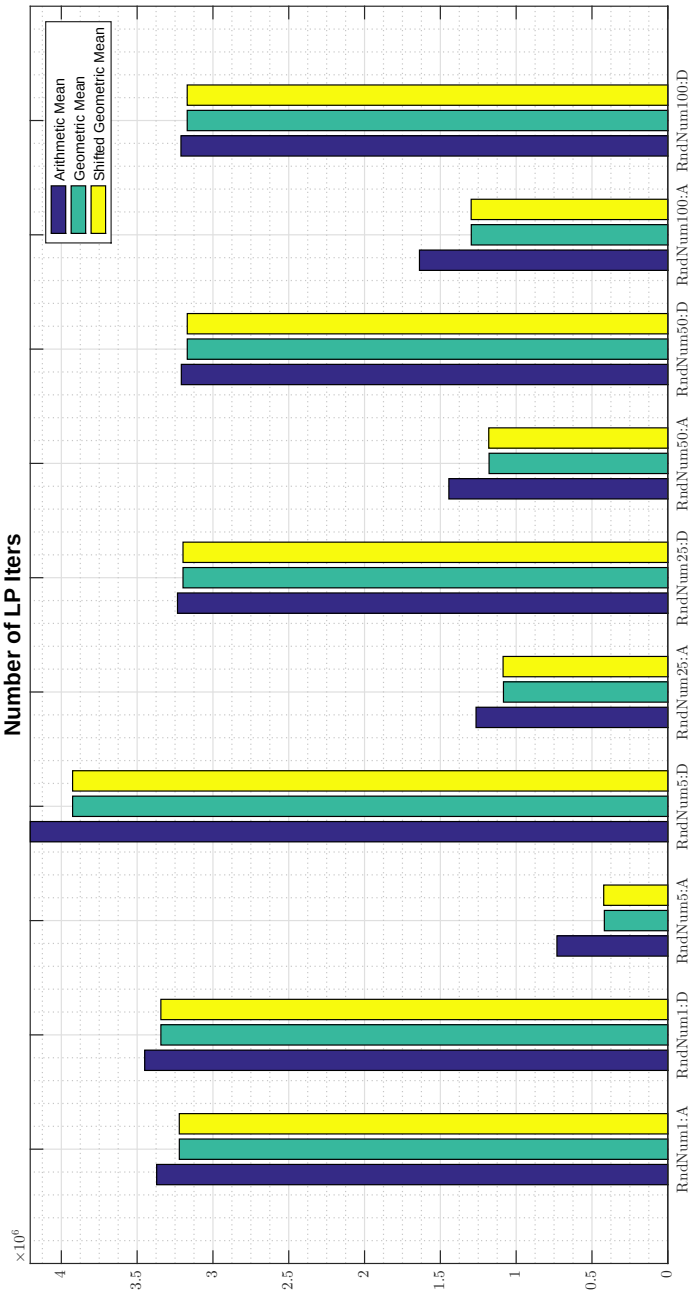



Figure 6.3: Reformulation Linearization Results: Number of LP Iterations.

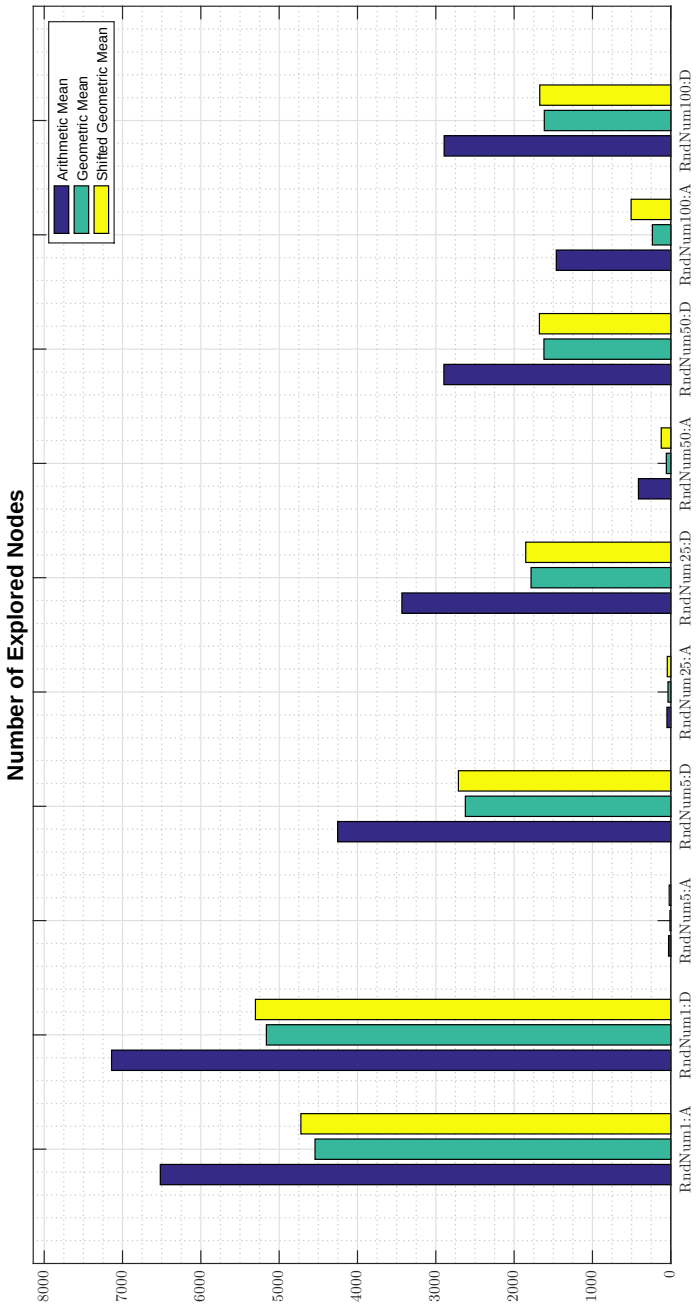



Figure 6.4: Reformulation Linearization Results: Number of BnB Nodes.

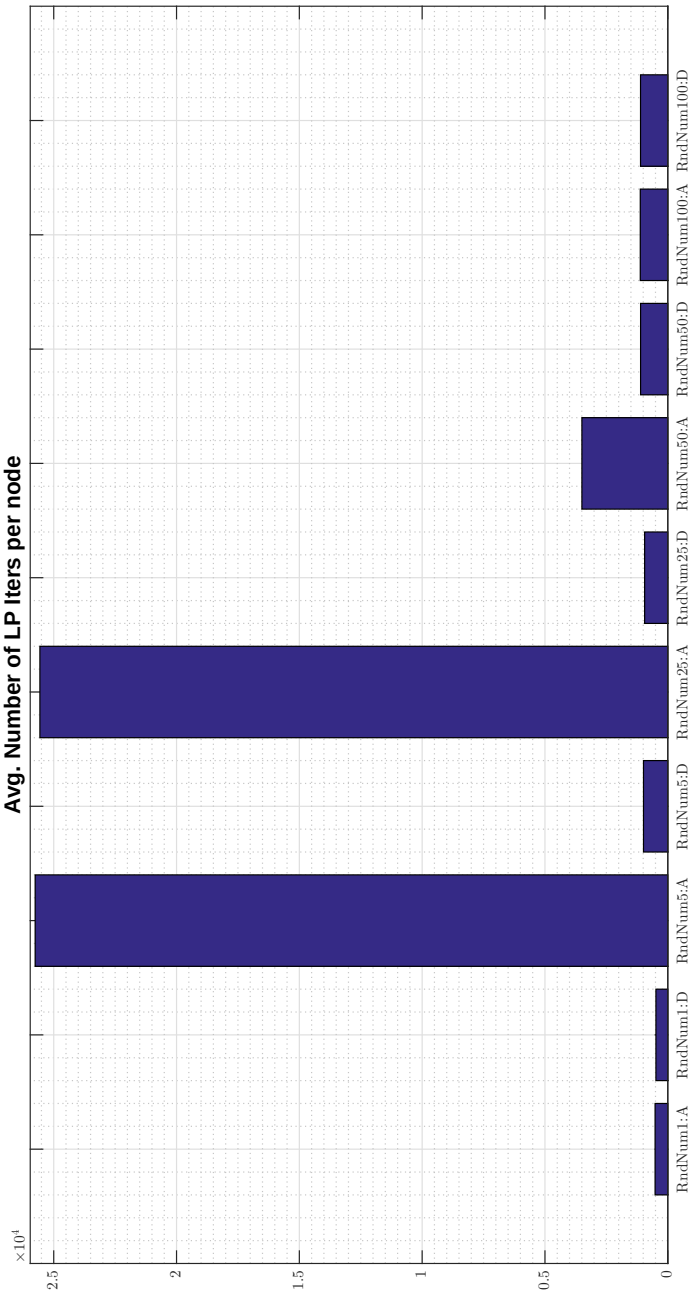



Figure 6.5: Reformulation Linearization Results: Average number of LP Iterations per Node.

by a reduction in the number of nodes in 'D' by about 31 % and a very small increase in the number of nodes in 'A'. For 25 presolving rounds, the number of LP iterations for 'A' is lower than 'D' by about 65 % but the number of average LP iterations per node is much higher by about 2450 %.

For presolving rounds 25, 50 and 100, it can be seen in Figure 6.2 that 'D' maintains the same duality gap while that of 'A' keeps decreasing. For 100 presolving rounds, we can see that 'A' has a duality gap lower than 'D' by 60 %. The number of BnB nodes gradually decreases slightly for 'D' when increasing the presolving rounds in the range 25, 50 and 100 while that for 'A' keeps increasing such that the number of nodes for 100 rounds increases by about 900 %. However at 100 presolving rounds, the number of nodes for 'A' is lower than 'D' by about 70 %. Figure 6.3 shows that the number of LP iterations for presolving rounds 25, 50 and 100 remains approximately the same for 'D' but increases slightly for 'A'. For presolving rounds 5, 25, 50 and 100 it can be seen from Figure 6.5 that the average number of LP iterations per node decreases enormously versus the number of rounds for 'A' and becomes equivalent to 'D' whose average LP iterations per node remains the same for rounds 5, 25, 50 and 100.

From the results, we can hence conclude that it is beneficial to use the reformulation linearizion technique for constraint set $\overline{C8}$ with a high number of presolving rounds (around 100).

## 6.2 Experiments and Results: Branching Schemes

In this section, the experimental procedures and results for the branching techniques given in Section 5.3 are provided. *Strong Branching* is not considered by itself in the experiments due to its expected high computational effort and time. However as

explained in Section 5.3, it is a component of *hybrid strong/psedocost*, *reliability* and *cloud* branching where its effect will be seen in those branching schemes. The main objective of the experiments is to evaluate the performance of the different branching schemes for $\mathcal{HAP}^{Eff}_{MBQCP}$ using:

1. the dual bound gap,
2. the total number of BnB nodes,
3. the total number of LP simplex iterations and
4. and the average number of LP iterations expended per node.

The following settings are considered for the experiments conducted for the branching schemes:

1. Separating cuts are deactivated,
2. node selection scheme is *best first search* with a maximum plunging depth in the BnB tree of 2 [64],
3. up to one round of presolving before starting the BnB algorithm,
4. for *hybrid strong/pseudocost branching*, maximum strong branching depths of $\tilde{d}_{strong} = 1$ and $\tilde{d}_{strong} = 2$ are tried in two different experiments.
5. For reliability branching the following settings are considered:
   - The maximum value for the reliability threshold is $\eta^{max}_{rel} = 5$,
   - the minimum value for the reliability threshold is $\eta^{min}_{rel} = 1$,
   - $\hat{\gamma}_{fixed} = 0$ in Equation (5.8),
   - maximum size of the set of strong branching candidates, $\overline{F} = 15$,

- maximum number of strong branching simplex iterations per branching variable is $\hat{\gamma}_{sbiterbrancand} = 100$,
- the ratio $c_{sbiterquot}$ in Equation (5.8) is set to $c_{sbiterquot} = 0.05$ and $c_{sbiterquot} = 0.2$ for two different experiments.

Figures 6.6, 6.7, 6.8 and 6.9 show the duality gap, number of LP iterations, number of BnB nodes and the average number of LP iterations per node for the different branching schemes. In those figures, HybDepth1 and HybDepth2 are the *hybrid strong/pseudocost branching* with strong branching invoked up to maximum depths of 1 and 2 respectively. Furthermore, $Relratio = 0.05$ and $Relratio = 0.2$ refer to *reliability branching* with $c_{sbiterquot} = 0.05$ and $c_{sbiterquot} = 0.2$. It can be seen that random branching has the highest duality gap, which is expected since the selection of branching candidates does not take into account the direction of change of the dual bound. The lowest duality gap was achieved (almost equally) by *inference branching*, *pseudocost branching*, *hybrid strong/pseudocost branching* ($\tilde{d}_{strong} = 1$ ) and surprisingly *most infeasible branching*. The second lowest are the *cloud branching* and *reliability branching* with $c_{sbiterquot} = 0.05$ equally both having a higher duality gap than the lowest four by about 18 %. Finally the second highest duality gap is obtained by *reliability branching* with $c_{sbiterquot} = 0.2$ with a duality gap higher than the lowest four by 50 %.

Comparing *pseudocost branching* versus *hybrid strong/pseudocost branching* with $\tilde{d}_{strong} = 1$, it can be seen that they almost perform equally in terms of duality gap, number of expended LP iterations, number of nodes and the average LP iterations per node. Increasing the depth for strong branching to $\tilde{d}_{strong} = 2$ in *hybrid strong/pseudocost branching* leads to an increase in the duality gap by 32 %. This is because when more strong branching is involved, a slightly greater number of LP

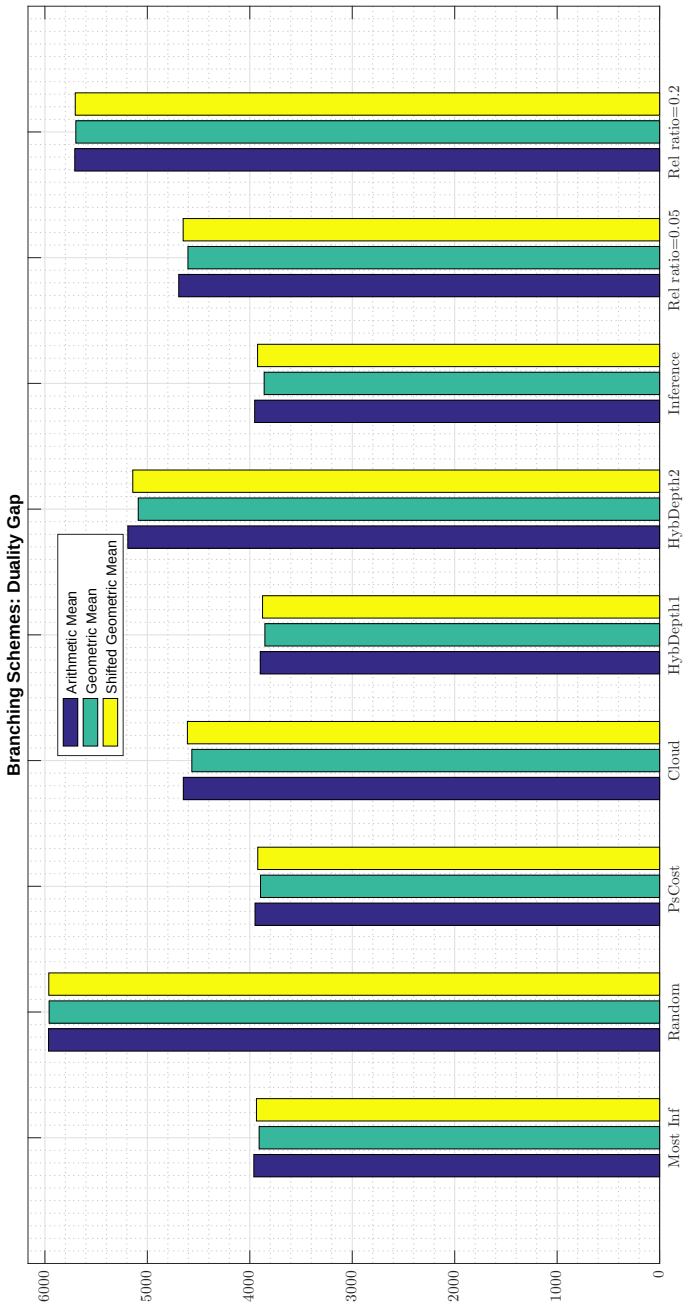



Figure 6.6: Branching Results: Duality Gap.

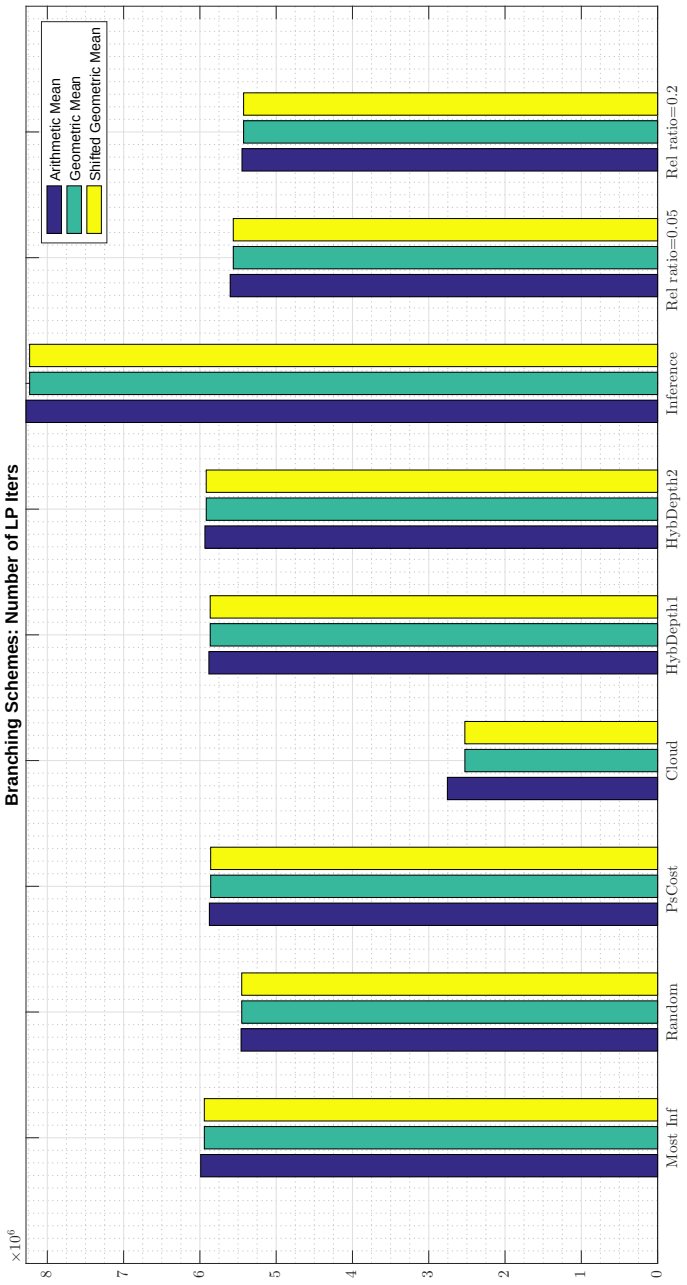



Figure 6.7: Branching Results: Number of LP Iterations.

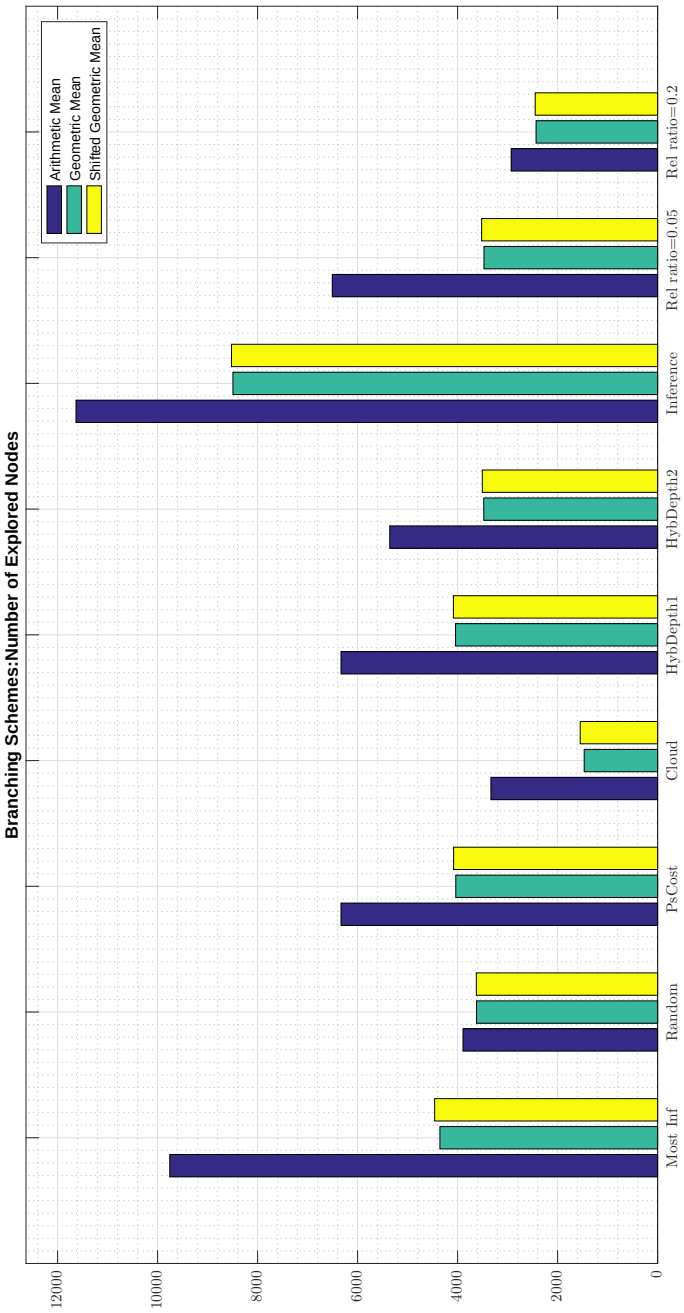



Figure 6.8: Branching Results: Number of BnB Nodes.

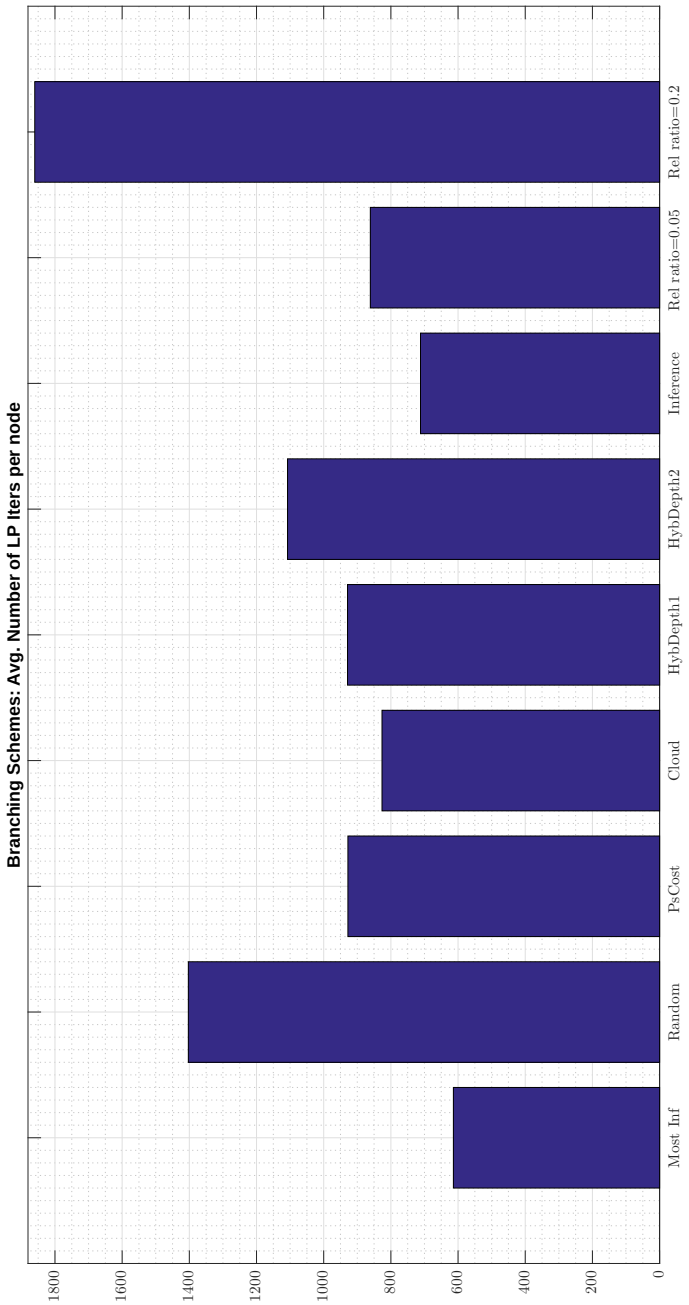



Figure 6.9: Branching Results: Average number of LP Iterations per Node.

iterations per node are expended as shown in Figure 6.9, meaning that in a given time limit, fewer nodes are explored as shown in Figure 6.8. When fewer nodes are explored for a given time limit, the overall dual bound improvement could be lower, even though the improvement per node can be higher for strong branching. The same reasoning applies for *reliability branching* in the two experiments in which $c_{sbiterquot} = 0.05$ and $c_{sbiterquot} = 0.2$.

Among the four branching schemes that give the lowest duality gaps, inference branching needs the largest number of nodes and LP iterations as Figures 6.8 and 6.7 show. *Cloud branching* expends the lowest number of nodes and LP iterations among all the branching schemes but has the second lowest duality gap. It requires 64 % less number of nodes and 58 % less number of LP iterations compared to *most infeasible branching*. Although *cloud branching* is based on strong branching, the cloud reduces out many LPs so that strong branching solves a small subset of those. It hence gives a better duality gap than HybDepth2 and *reliability* branching at $c_{sbiterquot} = 0.2$ and requires lower number of BnB nodes and LP iterations. It also gives an equally good duality gap for lower number of BnB nodes and LP iterations compared to *reliability* branching with $c_{sbiterquot} = 0.05$.

According to the observations and analysis based on the results in Figures 6.6, 6.7, 6.8 and 6.9, *cloud branching* seems to have a good trade-off balance of all the criteria of interest.

## 6.3 Experiments and Results: Separating Cuts, Domain Propagations, Heuristics

This section considers the impact of combinations of separating cuts, domain propagators and heuristics on the algorithmic computational performance in solving

$\mathcal{HAP}^{Eff}_{MBQCP}$. For each component, the selected types for the experiments are the ones described in Sections 5.4, 5.5 and 5.6. The relative impact on $\mathcal{HAP}^{Eff}_{MBQCP}$ of each component is measured by all the performance parameters mentioned at the beginning of this chapter. The following settings are considered for the experiments conducted:

1. node selection scheme is *best first search* with a maximum plunging depth in the BnB tree of 2 [64],
2. up to one round of presolving before starting the BnB algorithm,
3. *most infeasible branching* scheme was used,
4. a maximum number of domain propagation rounds of 1000,
5. unlimited separating cut rounds at the root node,
6. a maximum number of five separating cut rounds in the rest of the nodes,
7. a maximum number of cuts per round of 2000 at the root node,
8. a maximum number of cuts per round of 100 for the rest of the nodes,
9. a minimum orthogonality of 0.5 for a cut to enter the LP.

For the heuristics used in the conducted experiments, the priorities of calling the heuristics and how often in the BnB tree they are called are given in Table 6.3. The highest priority is given the numerical value 1 while the least is given the numerical value 10. The heuristics are called in decreasing order of their priority. The computationally simple heuristics are given the highest priority while those that are either more complex or are *improvement heuristics* are given lower priorities. The frequency parameter defines the level depths at which the heuristic is called. For

Table 6.3: Heuristics settings for the conducted experiments.

| **Heuristic** | **Priority level** | **Frequency** | **Frequency offset** |
|---|---|---|---|
| Simple Rounding | 1 | 1 | 0 |
| Rounding | 2 | 1 | 0 |
| Integer Shifting | 3 | 1 | 0 |
| Pseudocost Diving | 4 | 1 | 2 |
| RENS | 5 | 1 | 1 |
| Undercover | 6 | 2 | 0 |
| Clique | 7 | 2 | 1 |
| Feasibility Pump | 8 | 3 | 0 |
| RINS | 9 | 5 | 0 |
| Crossover | 10 | 3 | 3 |

example a frequency of 2 means that the the heuristic is called for nodes that are at depths 0, 2, 4, 6,.... . The frequency offset parameter defines the depth in the branching tree at which the heuristic is executed for the first time. For example frequency of 3 and frequency offset of 2 means that the heuristic is called at the depths 2, 5, 8, ... .

The results illustrated in the Figures 6.10, 6.11, 6.12, 6.13, 6.14, 6.15 and 6.16 compare the performance of the following for $\mathcal{HAP}^{Eff}_{MBQCP}$:

1. using branching only,
2. using branching and separating cuts (branch-cut),
3. using branching and domain propagation (branch-propagate),
4. using branching, separating cuts and propagation (branch-cut-propagate),
5. using branch-cut-propagate plus heuristics.

Figure 6.10 shows that using branch and cut lowers the duality gap only by approximately 5 % compared to using branching only for almost equal number of nodes and

LP iterations. Using branch and propagate lowers the dual gap by 33 % compared to branching only at the expense of an increased number of nodes by 700 % and increased number of LP iterations of 52 % as Figures 6.12 and 6.11 show. Moreover, Figure 6.13 shows that the average LP iterations per node for branch and propagate is much smaller than both, branching only and branch-and-cut. Furthermore, the branch-and-propagate manages to get a nontrivial feasible solution in 5 % of the instances within the time limit where branch only and branch-and-cut did not find any according to Figure 6.14.

Branch-cut-propagate does not lower the dual gap any further compared to branch-and-propagate. It even needs 150 % more BnB nodes and around 14% higher number of LP iterations as Figures 6.12 and 6.11 show. Moreover, the percentage of feasible solutions found goes down to 3 %.

Integrating heuristics to branch-cut-propagate reduces the duality gap by about 56 % due to the increased number of feasible solutions that are found (as given in Figure 6.14) which enhance the primal bound. The percentage of instances in which at least one feasible solution was found is about 70 % with heuristics versus 5% and 3% for branch-propagate and branch-cut-propagate without heuristics. Also, the time needed to obtain the first feasible solution decreases by around 64 % as Figure 6.15 shows. Moreover, the objective function value increases by nearly six times. This comes at the expense of an increase of 11 % in the number of LP iterations compared to branch-cut-prop as Figure 6.11 shows.

We can therefore conclude that using propagators and heuristics are recommended for $\mathcal{HAP}^{Eff}_{MBQCP}$ due to the improvements in performance they yield. Separating cuts however are more computationally expensive compared to the improvements they yield for $\mathcal{HAP}^{Eff}_{MBQCP}$.

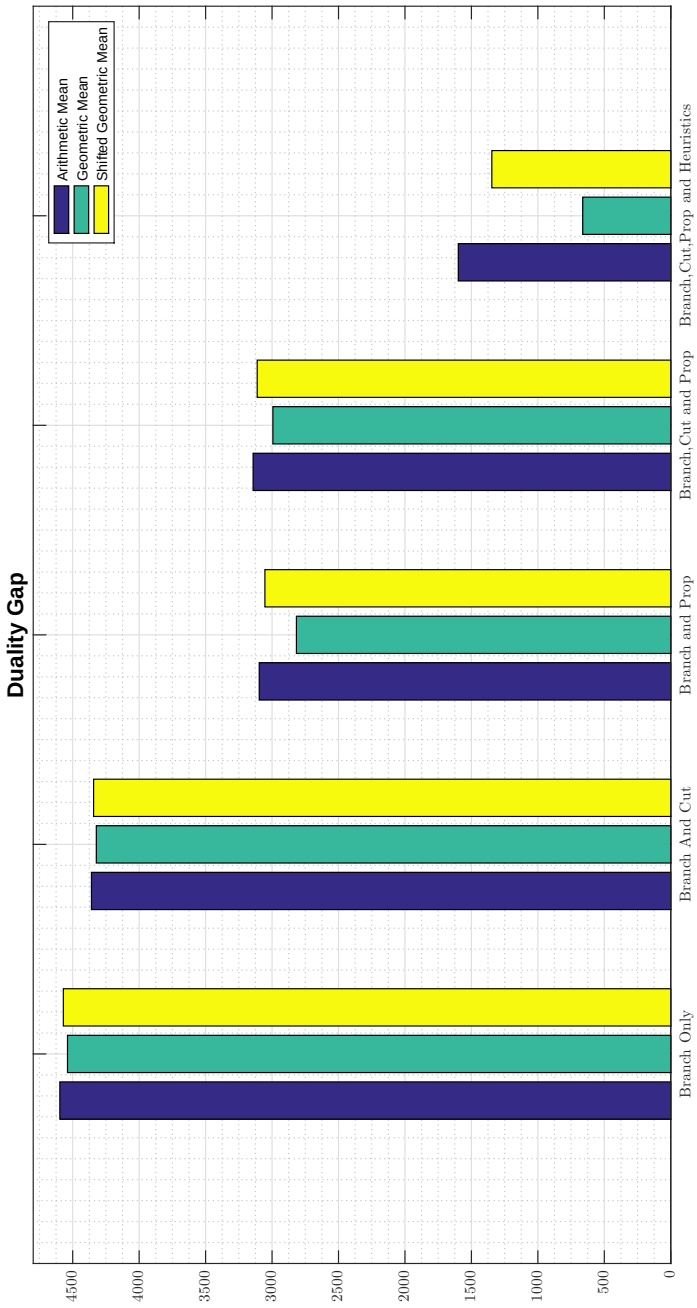



Figure 6.10: Cuts, Propagators, Heuristics: Duality Gap.

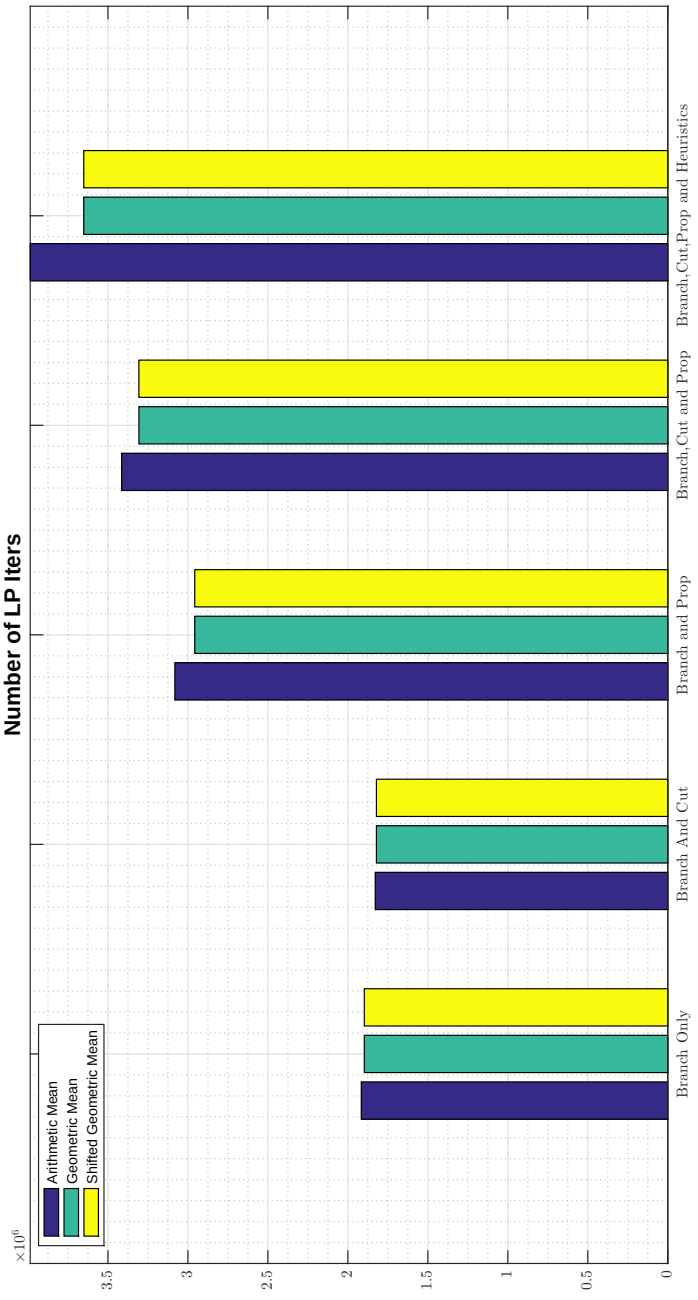



Figure 6.11: Cuts, Propagators, Heuristics: Number of LP Iterations.

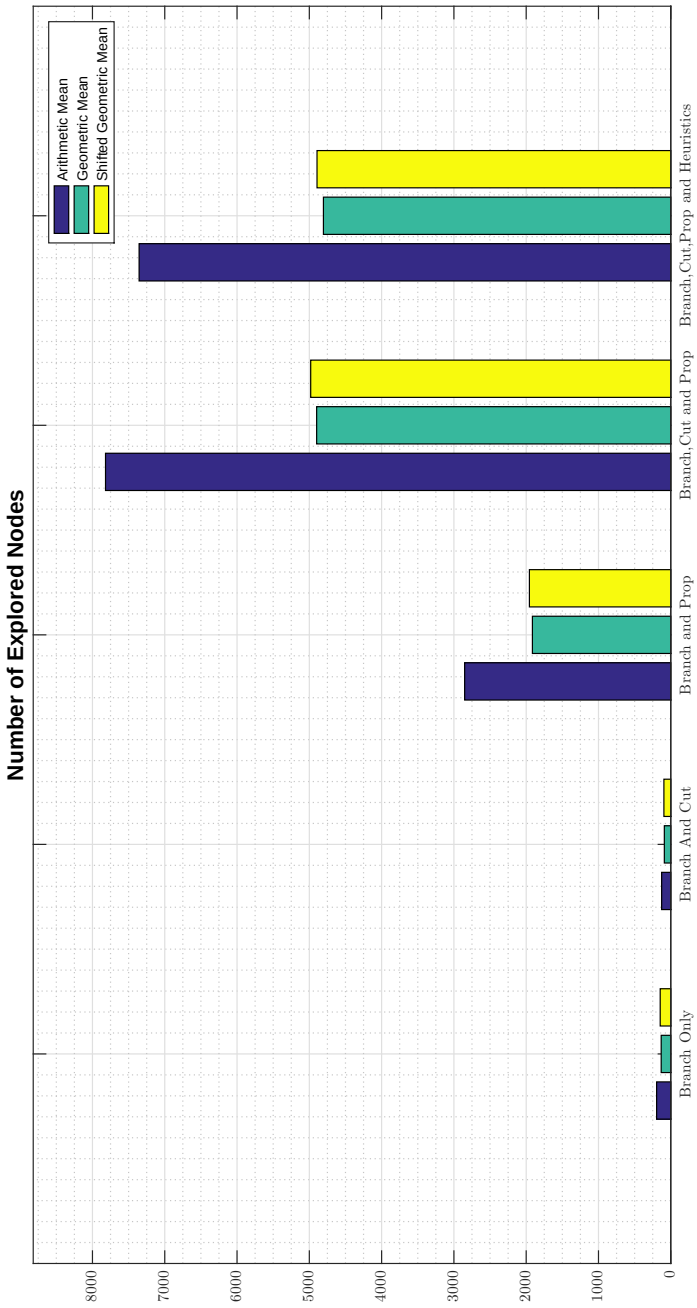



Figure 6.12: Cuts, Propagators, Heuristics: Number of BnB Nodes.

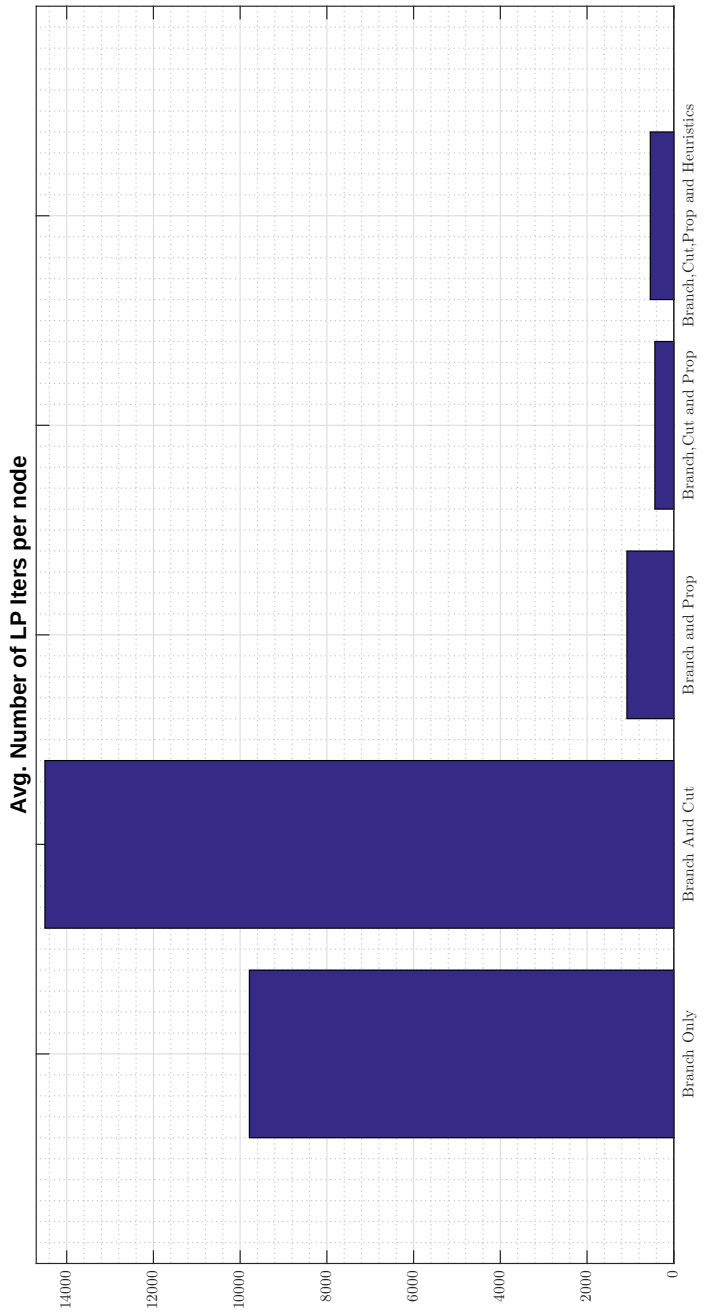



Figure 6.13: Cuts, Propagators, Heuristics: Average number of LP Iterations per Node.

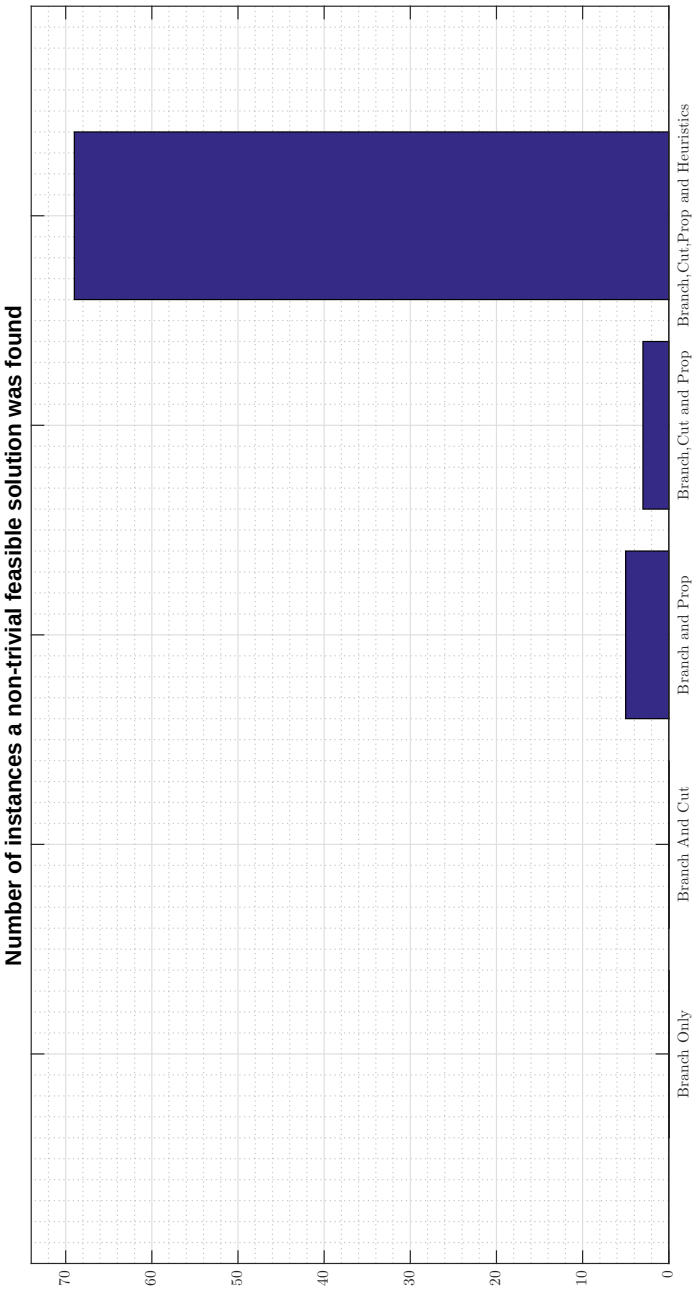



Figure 6.14: Cuts, Propagators, Heuristics: percentage of instances for which at least one feasible solution was found.

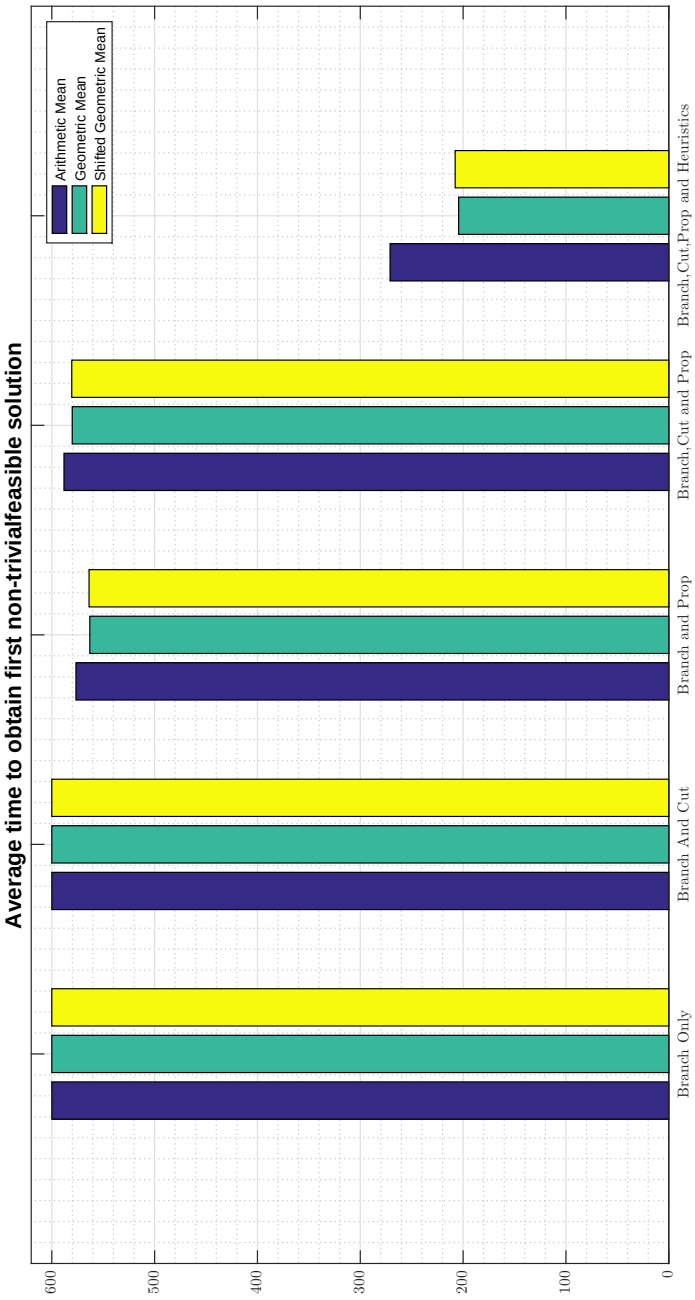



Figure 6.15: Cuts, Propagators, Heuristics: Time needed to obtain the first feasible solution.

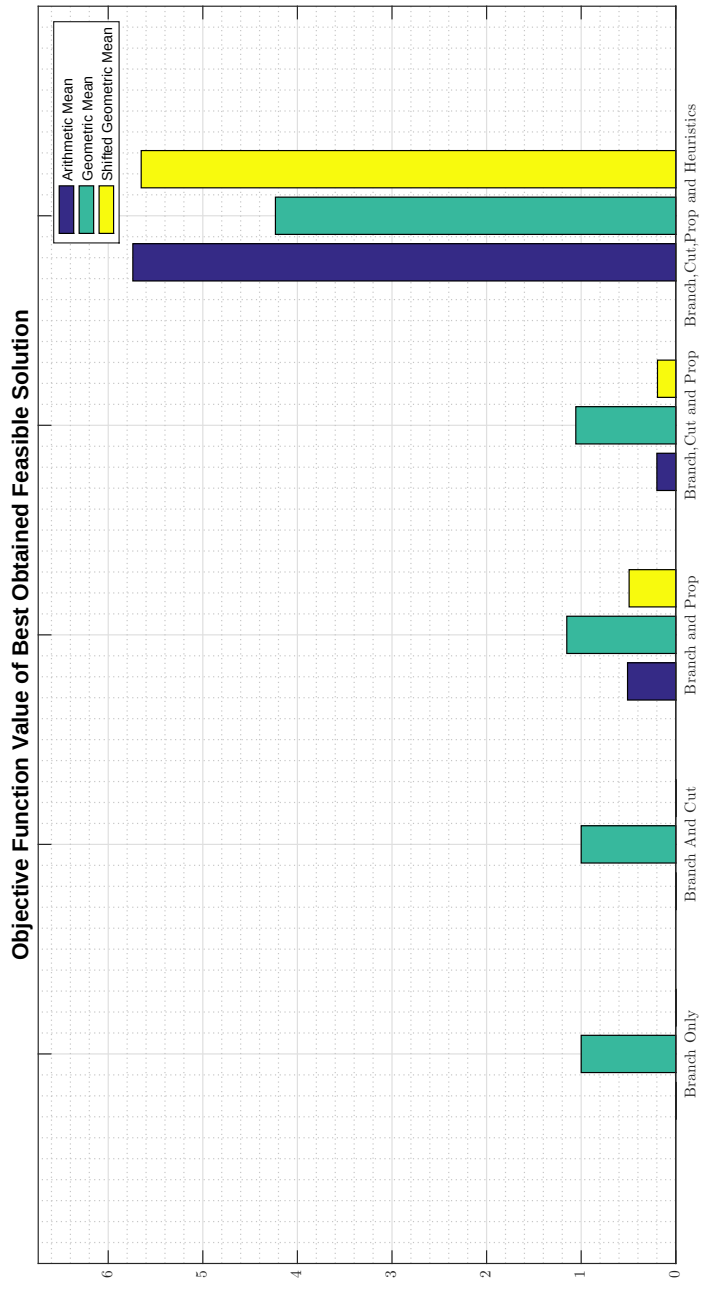



Figure 6.16: Cuts, Propagators, Heuristics: Objective function value for the best solution found.

# Chapter 7

# Conclusions and Future Work

In this chapter, the thesis conclusion and possible future work for extending the research work done in the thesis are provided.

## 7.1 Conclusion

In this thesis, a primary and extended system models were considered for joint AC-RRA for multicasting in an OFDMA based HAP system. The radio resources that were allocated were radio power, subchannels, time slots and antenna selection (in the extended system model). The users were assigned to the multicast groups so that the spectrum utilization is maximized in the primary system model while the numbers of highest priority users are maximized in the extended system model.

For the primary model (P-SysMod), the formulated problem was too complex which encouraged us to decompose the problem into two subproblems and use Lagrangian relaxation to dualize the complicating sets of constraints, thereby adding a penalty term to the objective function. An iterative *primal solution algorithm* was used to obtain the primal solution of the decomposed subprobelms. Also, since the

dualization of constraints introduced dual variables, the *subgradient algorithm* was used to obtain the optimum solution of the dual problem. Different step size rules were compared and different constraint set dualizations were tested to decide the ones that gave the sharpest solution bounds. The obtained bounds are intended to be used for the pruning process in BnB.

In the extended model (E-SysMod), a user was allowed to join any session being transmitted in all the neighboring cells where in P-SysMod a user only can join a subset of the sessions being transmitted in the cell it resides. Moreover, E-SysMod allowed a multicast group to receive transmission of a session on more than one antenna simultaneously, which is a more flexible option that was not available in the primary model. Also, the extended system model took into account heterogeneous user and session priorities which were considered to be homogeneous in P-SysMod. We were successful in obtaining $\mathcal{HAP}^{Eff}_{MBQCP}$, a formulation that is far more efficient in terms of the problem size compared to $\mathcal{HAP}^{Lagrange}_{2}$, the formulation used for P-SysMod. The more efficient formulation $\mathcal{HAP}^{Eff}_{MBQCP}$ was a separate achievement on its own that was used in E-SysMod. In other words, if E-SysMod is reduced to P-SysMod, $\mathcal{HAP}^{Eff}_{MBQCP}$ would still be much more efficient than $\mathcal{HAP}^{Lagrange}_{2}$.

McCormick underestimators were used for linear relaxation of the bilinear terms in the nonconvex quadratic constraints of $\mathcal{HAP}^{Eff}_{MBQCP}$. A branch and cut algorithm was used for solving the problem. Different constraint and objective function specific domain propagation techniques, that are available in the SCIP solver were integrated. Moreover different types of heuristic techniques were combined with the branch and cut algorithm which were shown to obtained more feasible solutions in the instances solved in shorter time. The experiments showed that domain propagation had a significant performance improvement as compared to the cutting planes used.

Furthermore different branching techniques were compared and the results showed *cloud branching* to achieve the best balance in the trade-off between the duality gap and the computational effort. A presolving reformulation linearizion technique for a specific set of quadratic constraints in $\mathcal{HAP}^{Eff}_{MBQCP}$ was used and the experiments indicated its effect on performance for different numbers of presolving rounds. According to the results, using the technique for a number of presolving rounds that is greater than or equal to 100 could improve the performance significantly.

## 7.2 Future Work

We believe that the work done in this thesis could be extended in at least three ways. One way is by using solution techniques from the optimization literature to solve the mixed binary polynomial constrained program (MBPCP) in Chapter 4 before reducing it to an MBQCP. The performances for solving both the MBPCP versus solving the MBQCP can be compared in terms of the goodness of the solutions, the computational effort and the memory requirements. For solving nonconvex MBPCP, recent developments for solution techniques were achieved by Nataraj et al. in [85], [86] and [87]. The approach they proposed is a Bernstein branch and prune algorithm that is based on the Bernstein polynomial approach. They introduced features like Bernstein box consistency and Bernstein hull consistency algorithms to prune the search regions. Using a Bernstein contraction algorithm, the computation of Bernstein coefficients after the pruning operation can be avoided which speeds up their algorithm. A Bernstein cutoff test based on the vertex property of the Bernstein coefficients is a key ingredient for the approach to solve a MBPCP.

Another possible extension is to use different nonlinear convex relaxation schemes for $\mathcal{HAP}^{Eff}_{MBQCP}$ aside from the linear relaxation, perhaps in a successive fashion like

in [88]. These could include different combinations of second order cone programming relaxations, semidefinite program relaxations or rank-2 valid linear inequalities [89], [90]. Furthermore, the MIQCP disjunctive cuts by Saxena et al. can be used to improve the relaxations further [91]. The two extension directions and the work done in this thesis for solving $\mathcal{HAP}^{Eff}_{MBQCP}$ could then be compared through extensive experiments to find which of all these techniques would yield the tightest bounds.

The second way the work in this thesis could be extended, is by relaxing the assumption that the measured channel gain values known to the HAP are ideally accurate. In a practical problem, measurement errors and prediction errors are encountered. Since the channel gain values $g_{i,k,c,t}$ constitute the problem formulation's parameters, it is important to know whether the solution obtained would be significantly different than an error free (ideal) situation, and whether it would still remain feasible to the formulation in the first place. It would be important to understand the tolerable ranges of errors for which the solution would not change and/or remain feasible, where large ranges are obviously desirable. By using *sensitivity analysis* we can evaluate these *insensitive* ranges and learn about the sensitivity of the problem's solution to channel measurements and estimation errors outside the calculated ranges [92]. The sensitivity of the problem's solution would essentially be the rate of the change in the objective value to the errors outside the calculated insensitive ranges. This type of knowledge could be used to determine the robustness of the optimization problem's solution and whether the channel estimation scheme is suitable for our multicasting AC-RRA problem. The observations could then be used to decide whether the channel estimation scheme would need improvement to achieve higher estimation accuracy, which would then increase the hardware complexity and cost.

The third way we can extend our work is by considering multi-rate multicast transmissions. Since in this thesis we proposed variants of the *least channel gain* (LCG) single rate multicasting, multi-rate transmissions would be a reasonable extension to the type of multicasting used in both system models we considered. Multi-rate multicasting exploits the multi-user multi-channel diversity within a multicast group enabling users to receive heterogeneous qualities for the information transmitted to each group. This is very useful when the information transmitted is of graphical multimedia content, like video for example. Each user in the group, could then perceive different video resolutions depending on its channel and interference conditions. For that purpose, *Information Decomposition Techniques* (IDT), where mutlimedia content is split to multiple sub-streams as we explained in Chapter 2, can be a good candidate for a multi-rate transmission scheme. Every user would receive a subset of the sub-streams, which correspond to a particular video resolution.

# Bibliography

[1] E. Falletti, M. Laddomada, M. Mondin, and F. Sellone, "Integrated Services from High-Altitude Platforms: a Flexible Communication System," *IEEE Communications Magazine*, vol. 44, no. 2, pp. 85–94, 2006.

[2] G. M. Djuknic, J. Freidenfelds, and Y. Okunev, "Establishing Wireless Communications Services via High-Altitude Aeronautical Platforms: A Concept Whose Time Has Come?," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 128–135, 1997.

[3] A. Mohammed, A. Mehmood, F. Pavlidou, and M. Mohorcic, "The Role of High-Altitude Platforms (HAPs) in the Global Wireless Connectivity," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1939–1953, 2011.

[4] T. Tozer and D. Grace, "High-Altitude Platforms for Wireless Communications," *Electronics & Communication Engineering Journal*, vol. 13, no. 3, pp. 127–137, 2001.

[5] ITU-R, "Preferred Characteristics of Systems in the Fixed Service Using High Altitude Platforms Operating in the Bands 47.2-47.5 GHz and 47.9-48.2 GHz," Recommendation F.1500, International Telecommunication Union, Geneva, May 2000.

[6] ITU-R, "Technical and Operational Characteristics for the Fixed Service Using High Altitude Platform Stations in the Bands 27.5-28.35 GHz and 31-31.3 GHz," Recommendation F.1569, International Telecommunication Union, Geneva, May 2002.

[7] ITU-R, "Minimum Performance Characteristics and Operational Conditions for High Altitude Platform Stations Providing IMT-2000 in the Bands 1,885-1,980 MHz, 2,010-2,025 MHz and 2,110-2,170 MHz in Regions 1 and 3 and 1,885- 1,980 MHz and 2,110-2,160 MHz in Region 2," Recommendation M.1456, International Telecommunication Union, Geneva, May 2000.

[8] ITU-R, "Technical and Operational Characteristics of Gateway Links in the Fixed Service Using High Altitude Platform Stations in the Band 5,850-7,075 MHz to Be Used in Sharing Studies," Recommendation F.1891, International Telecommunication Union, Geneva, May 2011.

[9] D. Grace and M. Mohorcic, *Broadband Communications via High-Altitude Platforms*. John Wiley & Sons, 2011.

[10] T. Tozer, G. Olmo, and D. Grace, "The European HeliNet Project," 2000.

[11] F. Dovis, L. L. Presti, E. Magli, G. Olmo, and F. Sellone, "HeliNet: A Network of UAV-HAVE Stratospheric Platforms. System Concepts and Applications to Environmental Surveillance," in *Data Systems in Aerospace*, vol. 457, p. 551, 2000.

[12] D. Grace, M. Mohorcic, M. Oodo, M. Capstick, M. B. Pallavicini, and M. Lalovic, "CAPANINA-Communications from Aerial Platform Networks De-

livering Broadband Information for All," *Proceedings of the 14th IST Mobile and Wireless and Communications Summit*, 2005.

[13] " CAPANINA project for the development of broadband communications capability HAPs." http://www.capanina.org.

[14] D. Grace, M. H. Capstick, M. Mohorcic, J. Horwath, M. B. Pallavicini, and M. Fitch, "Integrating Users into the Wider Broadband Network via High Altitude Platforms," *IEEE Wireless Communications*, vol. 12, no. 5, pp. 98–105, 2005.

[15] R. v. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Artech House, Inc., 2000.

[16] L. Zhao, L. Cong, F. Liu, K. Yang, and H. Zhang, "Joint Time-Frequency-Power Resource Allocation for Low-Medium-Altitude Platforms-Based WiMAX Networks," *IET communications*, vol. 5, no. 7, pp. 967–974, 2011.

[17] M. L. Fisher, "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces*, vol. 15, no. 2, pp. 10–21, 1985.

[18] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter 2004*.

[19] G. P. McCormick, "Computability of Global Solutions to Factorable Nonconvex Programs: Part IConvex Underestimating Problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[20] E. L. Lawler and D. E. Wood, "Branch-and-Bound Methods: A Survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[21] T. Berthold, A. M. Gleixner, S. Heinz, and S. Vigerske, "Analyzing the Computational Impact of MIQCP Solver Components," *Numerical Algebra, Control and Optimization*, vol. 2, no. 4, pp. 739 – 748, 2012.

[22] M. L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, vol. 27, no. 1, pp. 1–18, 1981.

[23] T. Berthold, S. Heinz, and S. Vigerske, *Extending a CIP Framework to Solve MIQCPs*. Springer, 2012.

[24] A. Abrardo and D. Sennati, "Centralized Radio Resource Management Strategies with Heterogeneous Traffics in HAPS WCDMA Cellular Systems," *IEICE Transactions on Communications*, vol. 86, no. 3, pp. 1040–1049, 2003.

[25] Y. C. Foo, W. L. Lim, and R. Tafazolli, "Centralized Total Received Power Based Call Admission Control for High Altitude Platform Station UMTS," in *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002.*, vol. 4, pp. 1596–1600, IEEE, 2002.

[26] Y. C. Foo, W. L. Lim, and R. Tafazolli, "Centralized Downlink Call Admission Control for High Altitude Platform Station UMTS with Onboard Power Resource Sharing," in *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, vol. 1, pp. 549–553, IEEE, 2002.

[27] Y. C. Foo, W. L. Lim, and R. Tafazolli, "Call Admission Control Schemes for High Altitude Platform Station and Terrestrial Tower-Based Hierarchical UMTS," in *The Ninth International Conference on Communications Systems, 2004. ICCS 2004.*, pp. 531–536, IEEE, 2004.

[28] Y. C. Foo and W. L. Lim, "Speed and Direction Adaptive Call Admission Control for High Altitude Platform Station (HAPS) UMTS," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pp. 2182–2188, IEEE, 2005.

[29] G. Araniti, A. Molinaro, and A. Iera, "Multicast in Terrestrial-HAP Systems," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 154–158, IEEE, 2007.

[30] Y. Liu, D. Grace, and P. D. Mitchell, "Exploiting Platform Diversity for GoS Improvement for Users with Different High Altitude Platform Availability," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 196–203, 2009.

[31] L. Zhao, J. Yi, F. Adachi, C. Zhang, and H. Zhang, "Radio Resource Allocation for Low-Medium-Altitude Aerial Platform Based TD-LTE Networks against Disaster," in *2012 IEEE 75th, Vehicular Technology Conference (VTC Spring)*, pp. 1–5, May 2012.

[32] L. Zhao, C. Zhang, H. Zhang, X. Li, and L. Hanzo, "Power-Efficient Radio Resource Allocation for Low-Medium-Altitude Aerial Platform Based TD-LTE Networks," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, Sept 2012.

[33] R. O. Afolabi, A. Dadlani, and K. Kim, "Multicast Scheduling and Resource Allocation Algorithms for OFDMA-Based Systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 240–254, 2013.

[34] P. Agashe, R. Rezaiifar, and P. Bender, "CDMA2000® High Rate Broadcast Packet Data Air Interface Design," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 83–89, 2004.

[35] J. Xu, S.-J. Lee, W.-S. Kang, and J.-S. Seo, "Adaptive Resource Allocation for MIMO-OFDM Based Wireless Multicast Systems," *IEEE Transactions on Broadcasting*, vol. 56, no. 1, pp. 98–102, 2010.

[36] K. Bakanoglu, W. Mingquan, L. Hang, and M. Saurabh, "Adaptive Resource Allocation in Multicast OFDMA Systems," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pp. 1–6, IEEE, 2010.

[37] J. Liu, W. Chen, Z. Cao, and K. B. Letaief, "Dynamic Power and Sub-carrier Allocation for OFDMA-Based Wireless Multicast Systems," in *IEEE International Conference on Communications, 2008. ICC'08.*, pp. 2607–2611, IEEE, 2008.

[38] P. K. Gopala and H. El Gamal, "On the Throughput-Delay Tradeoff in Cellular Multicast," in *International Conference on Wireless Networks, Communications and Mobile Computing, 2005*, vol. 2, pp. 1401–1406, IEEE, 2005.

[39] H. Won, H. Cai, K. Guo, A. Netravali, I. Rhee, K. Sabnani, *et al.*, "Multicast Scheduling in Cellular Data Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4540–4549, 2009.

[40] C. Suh and J. Mo, "Resource Allocation for Multicast Services in Multicarrier Wireless Communications," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, April 2006.

[41] C. Suh and J. Mo, "Resource Allocation for Multicast Services in Multicarrier Wireless Communications," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 27–31, Jan 2008.

[42] P. Eusiebio and A. Correia, "Two QoS Regions Packet Scheduling for MBMS," in *2nd International Symposium on Wireless Communication Systems, 2005.*, pp. 777–781, Sept 2005.

[43] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-Time Video Multicast in WiMAX Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[44] M. Shao, S. Dumitrescu, and X. Wu, "Layered Multicast With Inter-Layer Network Coding for Multimedia Streaming," *IEEE Transactions on Multimedia*, vol. 13, pp. 353–365, April 2011.

[45] C. H. Koh and Y. Y. Kim, "A Proportional Fair Scheduling for Multicast Services in Wireless Cellular Networks," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pp. 1–5, Sept 2006.

[46] T. Han and N. Ansari, "Energy Efficient Wireless Multicasting," *IEEE Communications Letters*, vol. 15, pp. 620–622, June 2011.

[47] N. Shrestha, P. Saengudomlert, and Y. Ji, "Dynamic Subcarrier Allocation with Transmit Diversity for OFDMA-Based Wireless Multicast Transmissions," in *2010 International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON),*, pp. 410–414, May 2010.

[48] S. M. Elrabiei and M. H. Habaebi, "Energy Efficient Cooperative Multicasting for MBS WiMAX Traffic," in *2010 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC),*, pp. 600–605, May 2010.

[49] J. Liu, W. Chen, Z. Cao, and K. B. Letaief, "Dynamic Power and Sub-Carrier Allocation for OFDMA-Based Wireless Multicast Systems," in *IEEE International Conference on Communications, 2008. ICC '08.*, pp. 2607–2611, May 2008.

[50] N. Sharma and A. S. Madhukumar, "Genetic Algorithm Aided Proportional Fair Resource Allocation in Multicast OFDM Systems," *IEEE Transactions on Broadcasting*, vol. 61, pp. 16–29, March 2015.

[51] H. Won, H. Cai, D. Y. Eun, K. Guo, A. Netravali, I. Rhee, and K. Sabnani, "Multicast Scheduling in Cellular Data Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1172–1180, May 2007.

[52] B. Da, C. C. Ko, and Y. Liang, "An Enhanced Capacity and Fairness Scheme for MIMO-OFDMA Downlink Resource Allocation," in *International Symposium on Communications and Information Technologies, 2007. ISCIT '07.*, pp. 495–499, Oct 2007.

[53] A. Ibrahim and A. S. Alfa, "Using Lagrangian Relaxation for Radio Resource Allocation in High Altitude Platforms," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5823–5835, 2015.

[54] A. Ibrahim and A. S. Alfa, "Radio Resource Allocation for Multicast Transmissions over High Altitude Platforms," in *IEEE Globecom Workshops (GC Wkshps), 2013*, pp. 281–287, IEEE, 2013.

[55] A. Ibrahim and A. S. Alfa, "Solving Binary and Continuous Knapsack Problems for Radio Resource Allocation over High Altitude Platforms," in *Wireless Telecommunications Symposium (WTS), 2014*, pp. 1–7, IEEE, 2014.

[56] D. A. Pearce and D. Grace, "Optimum Antenna Configurations for Millimetre-Wave Communications from High-Altitude Platforms," *IET Communications*, vol. 1, no. 3, pp. 359–364, 2007.

[57] J. Thornton, D. Grace, M. H. Capstick, and T. C. Tozer, "Optimizing an Array of Antennas for Cellular Coverage from a High Altitude Platform," *IEEE Transactions on Wireless Communications*, vol. 2, no. 3, pp. 484–492, 2003.

[58] P. A. Jensen and J. F. Bard, *Operations Research Models and Methods*. John Wiley & Sons Incorporated, 2003.

[59] http://stevehanov.ca/blog/index.php?id=122.

[60] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen, *Introduction to Algorithms*. The MIT press, 2001.

[61] J. F. Bard and P. A. Jensen, "Operations Research Models and Methods," 2003.

[62] S. Burer and A. Saxena, "The MILP Road to MIQCP," in *Mixed Integer Nonlinear Programming*, pp. 373–405, Springer, 2012.

[63] E. Eriksen, "Lecture 5: Principal Minors and The Hessian," 2010.

[64] T. Achterberg, "SCIP: Solving Constraint Integer Programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009. `http://mpc.zib.de/index.php/MPC/article/view/4`.

[65] T. Achterberg, *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007.

[66] W. L. Winston, M. Venkataramanan, and J. B. Goldberg, *Introduction to Mathematical Programming*, vol. 1. Thomson/Brooks/Cole, 2003.

[67] J.-M. Clochard and D. Naddef, "Using path inequalities in a branch and cut code for the symmetric traveling salesman problem.," in *IPCO*, pp. 291–311, 1993.

[68] T. Berthold and D. Salvagnin, "Cloud Branching," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 28–43, Springer, 2013.

[69] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, "Gomory Cuts Revisited," *Operations Research Letters*, vol. 19, no. 1, pp. 1–9, 1996.

[70] M. W. Savelsbergh, "Preprocessing and Probing Techniques for Mixed Integer Programming Problems," *ORSA Journal on Computing*, vol. 6, no. 4, pp. 445–454, 1994.

[71] R. Borndörfer and Z. Kormos, "An Algorithm for Maximum Cliques," *Unpublished working paper, Konrad-Zuse-Zentrum für Informationstechnik Berlin*, 1997.

[72] F. Domes and A. Neumaier, "Constraint Propagation on Quadratic Constraints," *Constraints*, vol. 15, no. 3, pp. 404–429, 2010.

[73] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an Efficient SAT Solver," in *Proceedings of the 38th Annual Design Automation Conference*, pp. 530–535, ACM, 2001.

[74] M. Fischetti, F. Glover, and A. Lodi, "The Feasibility Pump," *Mathematical Programming*, vol. 104, no. 1, pp. 91–104, 2005.

[75] L. Bertacco, M. Fischetti, and A. Lodi, "A Feasibility Pump Heuristic for General Mixed-Integer Problems," *Discrete Optimization*, vol. 4, no. 1, pp. 63–76, 2007.

[76] T. Achterberg and T. Berthold, "Improving the Feasibility Pump," *Discrete Optimization*, vol. 4, no. 1, pp. 77–86, 2007.

[77] T. Berthold, S. Heinz, M. E. Pfetsch, and S. Vigerske, "Large Neighborhood Search Beyond MIP," in *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pp. 51 – 60, 2011.

[78] G. Gamrath, T. Berthold, S. Heinz, and M. Winkler, "Structure-Based Primal Heuristics for Mixed Integer Programming," in *Optimization in the Real World*, vol. 13, pp. 37 – 53, 2015.

[79] T. Berthold, "RENS The Optimal Rounding," Tech. Rep. 12-17, ZIB, Takustr.7, 14195 Berlin, 2012.

[80] T. Berthold and A. M. Gleixner, "Undercover: A Primal MINLP Heuristic Exploring a Largest sub-MIP," *Mathematical Programming*, vol. 144, no. 1-2, pp. 315 – 346, 2014.

[81] E. Danna, E. Rothberg, and C. Le Pape, "Exploring Relaxation Induced Neighborhoods to Improve MIP Solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[82] E. Rothberg, "An Evolutionary Algorithm for Polishing Mixed Integer Programming Solutions," *INFORMS Journal on Computing*, vol. 19, no. 4, pp. 534–541, 2007.

[83] T. Berthold, "Primal Heuristics for Mixed Integer Programs," 2006.

[84] J. Currie and D. I. Wilson, "OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User," in *Foundations of Computer-*

*Aided Process Operations* (N. Sahinidis and J. Pinto, eds.), (Savannah, Georgia, USA), 8–11 January 2012.

[85] P. Nataraj and M. Arounassalame, "Constrained Global Optimization of Multivariate Polynomials using Bernstein Branch and Prune Algorithm," *Journal of Global Optimization*, vol. 49, no. 2, pp. 185–212, 2011.

[86] B. V. Patil, P. S. Nataraj, and S. Bhartiya, "Global Optimization of Mixed-Integer Nonlinear (Polynomial) Programming Problems: the Bernstein Polynomial Approach," *Computing*, vol. 94, no. 2-4, pp. 325–343, 2012.

[87] P. Nataraj and M. Arounassalame, "A New Subdivision Algorithm For the Bernstein Polynomial Approach to Global Optimization," *International Journal of Automation and Computing*, vol. 4, no. 4, pp. 342–352, 2007.

[88] M. Kojima and L. Tunçel, "Cones of Matrices and Successive Convex Relaxations of Nonconvex Sets," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 750–778, 2000.

[89] S. Kim and M. Kojima, "Second Order Cone Programming Relaxation of Nonconvex Quadratic Optimization Problems," *Optimization Methods and Software*, vol. 15, no. 3-4, pp. 201–224, 2001.

[90] S. Kim and M. Kojima, "Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations," *Computational Optimization and Applications*, vol. 26, no. 2, pp. 143–154, 2003.

[91] A. Saxena, P. Bonami, and J. Lee, "Convex Relaxations of Non-Convex mixed Integer Quadratically Constrained Programs: Extended Formulations," *Mathematical Programming*, vol. 124, no. 1-2, pp. 383–411, 2010.

[92] F. S. Hillier, *Introduction to operations research*. Tata McGraw-Hill Education, 2012.