OPTIMIZATION FRAMEWORK FOR SIMULTANEOUS TRANSMIT AND RECEIVE OPERATIONS IN WIRELESS LOCAL AREA NETWORK

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

INDUSTRIAL ENGINEERING

By

Ege Bilaloğlu

May 2022

Optimization Framework For Simultaneous Transmit and Receive Operations In Wireless Local Area Network
By Ege Bilaloğlu
May 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Oya Karaşan(Advisor)

Bahar Yetiş

Serpil Erol

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

OPTIMIZATION FRAMEWORK FOR SIMULTANEOUS TRANSMIT AND RECEIVE OPERATIONS IN WIRELESS LOCAL AREA NETWORK

Ege Bilaloğlu
M.S. in Industrial Engineering
Advisor: Oya Karaşan
May 2022

Full duplex communication technology draws substantial interest among wireless network operators due to its ability to increase the network capacity through concurrent transmissions. Despite this advantage, interference issue caused by close distances between stations makes it challenging to integrate simultaneous transmit and receive mode into wireless networks. Motivated by the objective of minimal overhead in full duplex transmissions of access points, we provide an optimization framework to minimize the latest completion time of transmissions. In this problem, we aim to find an optimal schedule of transmissions that maximizes concurrent operations in order to reduce the makespan. We formulate the problem for both single and multiple concurrency assumptions separately. For single concurrency, we provide a mixed integer programming (MIP) model using scheduling based formulation along with a greedy heuristic. Modeling the problem as a matching problem between two disjoint sets of supplies and demands, we develop a linear programming (LP) model with a totally unimodular constraint matrix. We utilize Hopcroft-Karp algorithm for solving the resulting maximum cardinality bipartite matching problem. For multiple concurrency; we formulate a flow based integer programming model, demonstrate properties of the extreme points in its LP relaxation, develop valid inequalities and optimality cuts. As an extension, we add due dates for each station to complete their transmissions and formulate an MIP model and develop an algorithm for this variant. Additionally, we provide a proof for NP-completeness of minimum total tardiness problem with single concurrency. To evaluate the performance of the proposed formulations, we perform a range of computational experiments. Finally, we conduct sensitivity analyses to evaluate the effects of the parameters on the objective value and the solution times.

Keywords: broadcast scheduling, maximum cardinality matching, mathematical programming, discrete optimization, network design.

ÖZET

KABLOSUZ YEREL ALAN AĞINDA EŞ ZAMANLI İLETİM VE ALIM ENİYİLEMESİ

Ege Bilaloğlu
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Oya Karaşan
May 2022

Tam çift yönlü iletişim teknolojisi, eşzamanlı iletimlerle ağ kapasitesi artırma etkisiyle kablosuz ağ operatörlerinden büyük ilgi görmektedir. Buna karşın; istasyonlar arasındaki yakın mesafe kaynaklı çakışma sorunu, eşzamanlı gönderme ve alma modunu kablosuz ağlara dahil etmeyi zorlaştırmaktadır. Bu çalışmada, erişim noktalarının tam çift yönlü sinyal iletimlerini en az maliyetle gerçekleştirilebilme motivasyonu doğrultusunda sinyal iletimlerinin son tamamlanma süresini enazlama yöntemi sunulmuştur. En son tamamlanma süresini enazlamak amacıyla eşzamanlı operasyonları en üst düzeye çıkaran iletim çizelgelemesi hedeflenmiştir. Problem tekli ve çoklu eşzamanlılık seçenekleri ayrıştırılarak ikisi için de çözümlenmiştir. Tek eşzamanlılık varsayımında sezgisel algoritma yönteminin yanı sıra çizelgelemeye dayalı karma tamsayılı doğrusal programlama formülasyonı geliştirilmiştir. Problemin iki ayrık arz ve talep kümeleri arasındaki eşleşme sayısını maksimuma çıkarma amacında yorumlanmasıyla, eşleştirmeye dayalı tamamen tek modüler doğrusal programlama modeli sunulmuştur. Aynı zamanda, Hopcroft-Karp algoritması maksimum iki parçalı grafik eşleştirmesi için kullanılmıştır. Çoklu eşzamanlılık için ise akış tabanlı tamsayı modeli ve doğrusal gevşetme modeli oluşturup kesişme noktalarının sayısal özellikleri kanıtlanmıştır. Ek olarak, istasyonlara zaman sınırı eklenerek karma tamsayılı doğrusal programlama modelive sezgisel algoritma geliştirilmiştir. Tek eşzamanlılık varsayımında istasyonların toplam gecikme süresini enazlama probleminin NP-zorluğu kanıtlanmıştır. Formülasyonların performansını ölçmek için sayısal analizlerle birlikte duyarlılık analizi de yapılarak problem parametrelerinin en iyi çözüm değeri ve süreleri üzerindeki etkileri saptanmıştır.

Anahtar sözcükler: sinyal çizelgeleme, maksimum eşleştirme, matematiksel programlama, ayrık eniyileme, ağ tasarımı.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Oya Karaşan for continuous support, guidance and understanding she has provided during my study. She has always believed in me with sincerity for three years. I feel privileged to have had the chance to study under her supervision.

I am indebted to Prof. Bahar Yetiş and Prof. Serpil Erol for accepting to read and review my thesis. I appreciate their valuable recommendations.

I want to acknowledge that this study was financially supported by The Scientic and Technological Research Council of Turkey (TUBITAK) as part of 2210-A National Graduate Study Scholarship Program.

I would like to thank my friends Damla Akoluk, Dilara Sönmez and Tuğba Denktaş for their invaluable friendship and making these challenging three years much easier for me. I have shared unforgettable memories with them. I am also beholden to Milad Malekipirbazari, Parinaz Toufani, Çağla Dursunoğlu, Emin Özyörük, Serkan Turhan and Duygu Söylemez for their help and guidance.

I want to thank my dearest parents Dilek Bilaloğlu and Ertuğrul Bilaloğlu. I cannot describe how grateful I am for their unconditional love they have given to me. I want to thank my beloved sister Pelin Yurttutan for giving me her endless love and support, and motivating me anytime I need. I would not be able to reach my goals if it was not for the countless sacrifices they have made throughout my life to make me happy.

Lastly, I cannot express with words how lucky I am to have my twin and my best friend Eren Bilaloğlu in my life. I feel so appreciated that he has always been with me no matter how difficult the situations that I was in. I am forever grateful for everything he has done for me. It has been wonderful to feel his brotherhood and support along this way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background Information

Full duplex technology has attained tremendous interest among mobile operators in recent years. Even though it was first introduced more than a decade ago, its continuous benefits for efficient usage of the spectrum channel in the communication networks have augmented its popularity [1]. Especially in wireless networks, most of the devices used in the communication operations are aimed to be designed with full duplex technology rather than half duplex technology. In order to understand the advantages of the full duplex technology, it is important to comprehend the inner workings of the duplex communication systems.

Duplex communication systems are used for signal transmission. The system structure composes of two or more connection nodes (devices or parties) that can communicate with each other using both directions. Simultaneous communication property in both directions is the primary reason why they are employed in various communication networks. According to the preceding advancements in the telecommunication field, duplexing can be either achieved by using half-duplex technology or full-duplex technology [2].

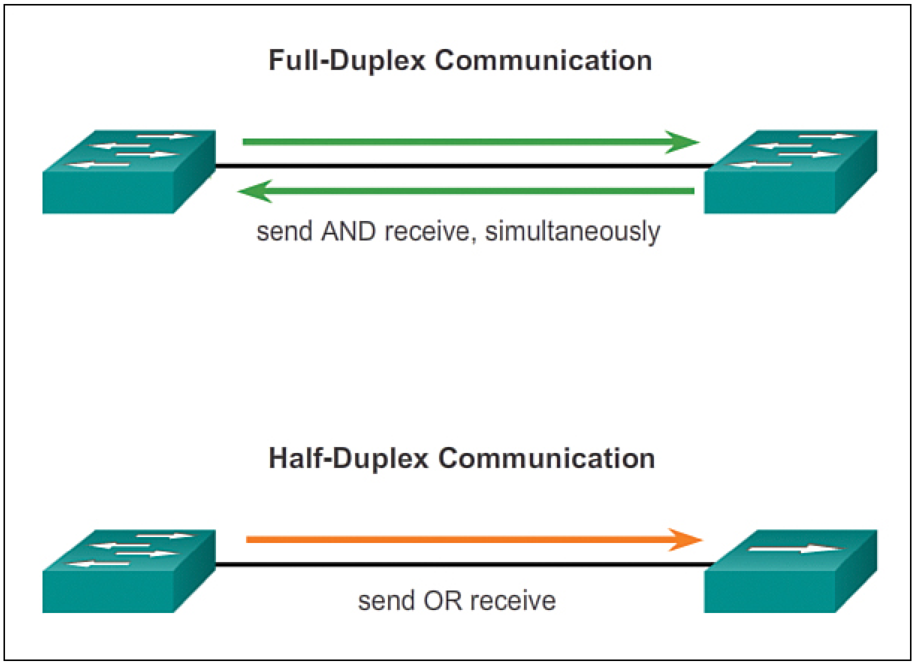

Figure 1.1: Half duplex and full duplex technology comparison

The key distinction of full duplexing is that it enables communication devices to transmit and receive signals simultaneously. For instance, during a telephone conversation, as the first speaker is talking, the microphone of the first speaker can transmit its message. At the same time, the second speaker can hear that message since earphone reproduces the talking of the other speaker through full duplex technology. On the other hand, half duplex technology is only limited to communication between both parties but incapable of simultaneous transmission of input and output signals; allowing only one direction communication at a time [3]. Whenever people do internet surfing on a daily basis, they have to wait for the response from the network after searching due to limitation of the half duplex technology in the computer network. The difference between half duplex and full duplex technologies is depicted in Figure 1.1.

The deficiency of half duplex communication mode results from self-interference, which occurs as an outcome of disparity between the signal strengths of the receiver and transmitter. In wireless networks specifically, most of the devices are classified with half duplex property. This is because transmission signals that wireless device produces are more powerful than the ones that wireless device receives [4]. Having affected by self-interference, performance of the receiver is degraded as the input signal is overwhelmed by disparity. Since wireless devices do not have full duplex property due to practical limitations, having full duplex access point and half duplex clients at the same time creates interference. This issue is illustrated in Figure 1.2. In contrast, full duplex system accomplishes cancellation of self-interference and makes it possible for radios to receive and transmit on the same frequency level [5]. Subsequently, capacity of the signal transfer in the communication system can double.

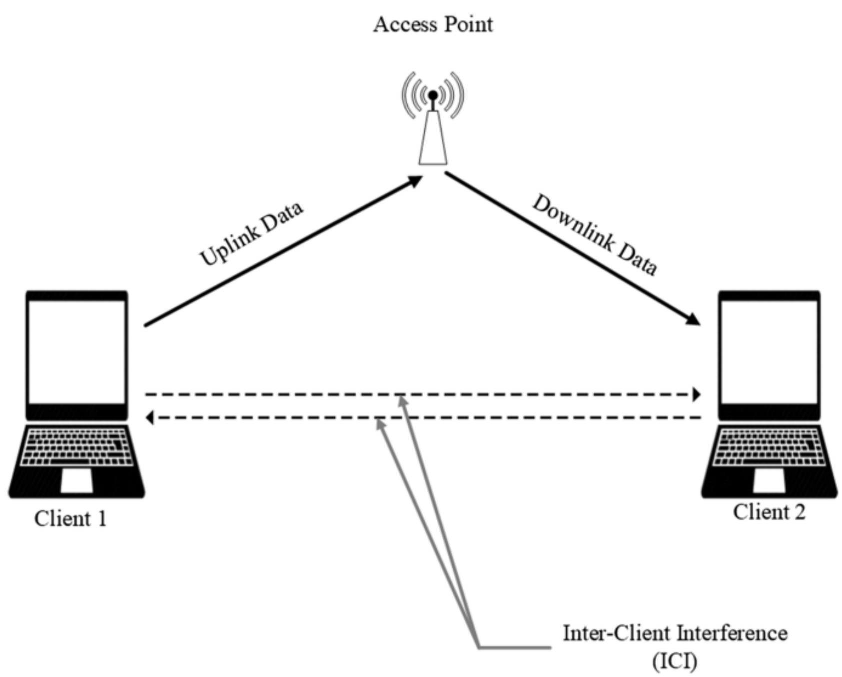

Figure 1.2: Interfering half duplex clients in a wireless network

As a requirement to efficiently deploy the emerging trends like 5G systems technology and next generation wireless local area networks (WLANs), most of the wireless networks such as cellular networks and computer networks are designed in a way to implement full duplex technology [6]. In an attempt to enable full duplex communications in infrastructure based wireless networks, medium access control (MAC) protocol designs are developed. MAC stands for the network data transfer policy deciding on how the data is transmitted from one computer terminal to another through the network cable [7]. The principle behind MAC protocol design is to facilitate the transfer of data packets among computer terminals so as to make sure that no collision between computer terminals occurs. Since collision takes place when more than one computer terminal transmit the data simultaneously, full duplex implementation into computer networks is challenging to employ.

Commonly known as Wi-Fi, the most popularly used computer networks are WLANs which are based on the IEEE 802.11 standards [8]. Simple illustration of the wireless network topology is provided in Figure 1.3. Adopting full duplex technology in IEEE 802.11 standard requires the implementation of simultaneous transmit and receive (STR) mode into the next generation WLANs. While there have been proposed solutions generally based on investigating physical layer aspects (PHY) of the wireless network, MAC layer solutions are also offered to enhance full duplex communication in WLANs [9]. Those protocols are aimed to address the issues that hinder providing the increased user throughputs and to carry out successful transmission in busy deployments by increasing the performance of WLANs.

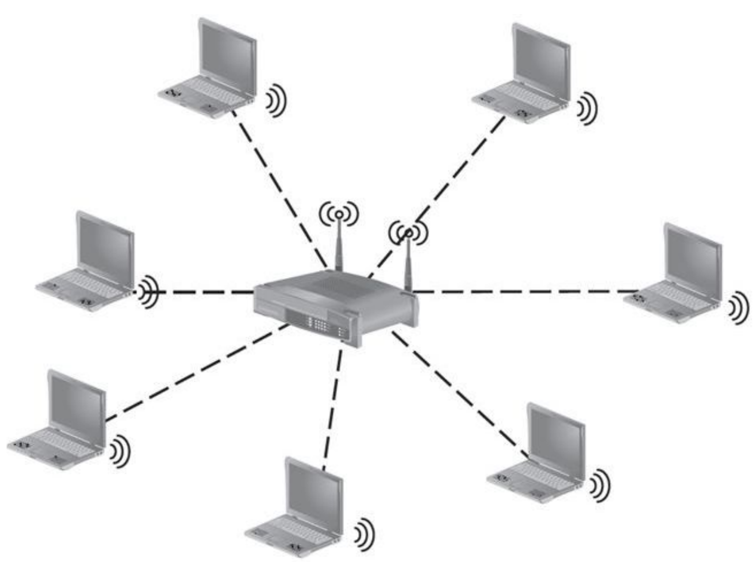

Figure 1.3: Wireless network topology

Integrating full duplex communication mode into 802.11 networks brings about several difficulties. The challenging part is to respond these issues with as minimal protocol modifications as possible. First of all, there is coexistence issue of full duplex (FD) and half duplex (HD) nodes in the network. Even after enabling STR mode to the network, there will be FD and legacy HD nodes at the same time [10]. At the beginning of FD transmissions, it is essential for FD nodes to autonomously discover which nodes have FD capability. This capability discovery due to coexistence issue can engender expenses caused by legacy frame structure modifications.

STR mode in WLANs produces two types of wireless links: unidirectional full duplex (UFD) and bidirectional full duplex (BFD) links. In BFD transmission, the pair of access point and station that are FD capable can transmit and receive simultaneously among each other. However, through the UFD links, only the access point can simultaneously transmit signal to an FD or HD capable station whilst receiving signal from another station. Enabling these two links is desired to obtain maximal benefit of FD technology, but modification to legacy channel could create backward compatibility for half duplex transmissions that need to be avoided. Backward compatibility means that current system has interoperability with the older legacy system. It is a drawback since larger bill of materials are required to support older versions that is costly.

Stations and access point can process BFD transmission after discovering that stations have FD capability whereas this is not the case for UFD transmission. In UFD transmission, node selection decision must be made to complete the communication. No matter which type of capability the stations have, either HD or FD, each station should satisfy a given criteria. If two stations are in the interference range of each other, then access point cannot simultaneously receive signal from one and transmit signal to another. That's why access point needs to possess extra function to decide on the set of nodes that are eligible for the second transmission after the first transmitter node is given. This functionality requirement adds overhead cost to the communication network.

1.2 Motivation

In this thesis, in order to assess the full duplex technology in communication networks, we mainly investigated simultaneous transmit and receive operation in next generation IEEE 802.11 WLANs. We narrowed down the scope of our investigation by focusing only on Unidirectional Full Duplex (UFD) genre of wireless links. Provided that not all the stations can become part of the UFD transmission since two nodes simultaneously served by an access point must be out of interference range, we primarily dealt with node selection process of an access point for the UFD transmission.

To incur minimal overhead for the whole process, our goal was to determine the set of stations to incorporate into the simultaneous transmit and receive mode taking into account the interference restrictions. We evaluated the overhead cost incurred to the communication network as the time slots used to broadcast station transmissions. For this reason, we have chosen our performance measure as the number of time slots and our motivation was to minimize the total time to complete all UFD transmissions.

Perusing the issue from an industrial engineer perspective, we redefined the problem as supply and demand scheduling through a base station where access point gets supply from transmitters and provides demand to receivers in the communication network. We have developed a mathematical modeling approach by assigning each supply or demand station to a time slot allowing at most two transmissions in an individual slot so that latest time slot is minimized.

Due to the complexity of the mathematical model having station time slot assignments, we obtained the optimal results after long computation times. This motivated us to identify the dimension of time slot set so that our model can be used in optimization. Therefore, a greedy heuristic methodology was followed to find and upper bound on the number of time slots required. The result of the heuristic was used as an input to our mathematical model and solution times have been significantly decreased.

As we analyzed the properties of our problem, we realized that the supply and demand stations can be considered as two disjoint sets since in our initial problem, UFD transmission did not allow two receivers or two transmitters processed in the same time slot. This prompted us to explore the problem within the scope of graph theory where disjoint data sets were represented as a bipartite graph.

Using this graph modeling approach, we were able to recast our problem as a maximum cardinality matching problem. Indeed, accomplishing maximum simultaneous receive and transmission operations as matchings could lead to minimum total duration of operation times due to the least number of unmatched signals. In other words, we maximized time slot utilization in terms of benefiting signal capacity of each time slot.

Based on maximum cardinality matching, an equivalent optimization model providing the same results within much less solution times is formulated. Hopcroft–Karp algorithm was also employed in order to find the optimal solution with reasonable solution times for large scale problems. The matching principle of supply and demands was followed within the cases where we extended our problem to the twice concurrent stations. With this extension, we allowed two supplies and two demands to be processed in the same time slot. To model this consideration, four indexed and two indexed formulations are developed.

Elaborating the scope of our problem under single concurrency, we considered the deadline constraints for each station transmission as another extension. Restricting each station to meet the deadline criteria for completing the total of supply and demand operations, the minimum total tardiness objective is embraced. The mathematical model for this restriction is formulated and a greedy algorithm is developed. We proved that this problem is NP-complete by demonstrating that the special case of our problem is identical to another NP-complete problem found in machine scheduling literature.

When we removed the deadline criteria, we inserted concurrent broadcasting allowance of stations with multiple signals at a time, expanding the interference constraint from twice concurrency into multiple concurrency cases. Instead of creating supply and demand copies for each station, we developed flow based mixed integer linear programming formulation where we aimed to achieve maximum concurrent flow of signals meeting the supply and demand amounts.

Performing LP relaxation on the flow based formulation, we analyzed the numerical properties of extreme points of the new model. The analysis was carried out by utilizing the knowledge of linear programming and graph theory. We have proved several theorems for our problem, which enabled us to observe the behavior of optimal results. As a result, we were able to provide a valid inequality and an optimality cut for the flow model.

1.3 Overview

The remaining chapters of this thesis are organized as follows:

Chapter 2 consists of the detailed review of literature about our problem and corresponding methodologies developed in the past studies. In this chapter, we focused on previous studies on broadcast scheduling problem in general, then we provided the machine scheduling literature related to our problem. At the end of this chapter, different approaches for maximum cardinality matching problem are reported.

Chapter 3 corresponds to the problem definition and includes the explanation for the essential notations, parameters and descriptions used throughout the thesis. In this section, different extensions of the problem are also introduced in brief. The problem setting with the network structure, preliminaries and our general approach are also provided.

In Chapter 4, the problem is investigated for the minimization of the latest completion time in the communication network. First, single concurrent stations are assumed in the network and scheduling based model formulation and greedy heuristic for the model are given. Matching based formulation and Hopcroft-Karp algorithm are provided afterwards for the single concurrency. Secondly, twice concurrency and multiple concurrency are studied and formulated with MILP models. For the extreme points of the LP relaxation of the flow model, significant properties are proved and further supported with valid inequalities and optimality cuts.

In Chapter 5, deadline criteria is introduced to the problem. The objective of minimum total tardiness is analyzed for the problem and the model formulation is provided with detailed explanation of its constraints and objective function. The algorithm that we developed for deadline criteria is also presented. Eventually, NP-completeness proof for the minimum total tardiness problem under single concurrency is provided.

Chapter 6 is devoted to the computational experiments and analyses for all the formulations and algorithms defined in this thesis. In addition, we provide sensitivity analyses and discuss the results of the experiments in detail. Finally in Chapter 7 we conclude our discussion and propose future research directions that can be undertaken.

Chapter 2

Literature Review

Even though full duplex technology is remarkably promising to aggrandize the communication in wireless networks, the literature on the optimization of medium access control to manage FD wireless communication is relatively narrow. As we initially approach to the node selection problem from scheduling perspective, we primarily focus on review of literature on scheduling problems. Broadcasting is mostly performed in local area network (LAN) technologies [11]. Since the main concept of broadcasts in computer networking also faces with interference issue in scheduling, we examined broadcast scheduling problem. Moreover, we inspect machine scheduling literature by considering the notion of incompatible job families around minimum total tardiness and minimum makespan objectives. Finally, since we also assess the problem with maximal matching motivation, we examine maximum cardinality matching problem and its solution methodologies studied in the literature.

2.1 Broadcast Scheduling

Broadcast scheduling problem is extensively studied in the literature. In any wireless network, broadcast is the fundamental operation. However, broadcast scheduling in wireless networks is known to be challenging since it is difficult to deal with interference. Most of the existing literature explores the problem by modeling the wireless networks using graphs $G = (V, E)$ where vertices in $V = \{1, \ldots, n\}$ represent stations and set of edges E corresponds to transmission links indicating reachability from one station to another.

The problem of finding an optimal protocol to minimize the maximum time of broadcasting in radio network is proved to be NP-hard in [12]. In this study, Chlamtac and Kutten introduce a polynomial time algorithm in which a greedy heuristic generates a collision-free broadcast spanning tree and assigns transmission to all nodes. Ephremides and Turong [13] demonstrate the NP-completeness of throughput optimization problem in multi-hop radio networks subject to interference-free broadcasting schedules. Polynomial time centralized and distributed algorithms that produces maximal schedules are developed.

Lloyd and Ramanathan [14] introduce novel algorithms for both link and broadcast scheduling. For the tree networks, the proposed algorithm yields optimal solutions and for the other networks an upper bound for the schedule length is produced. That's why they introduce the notion of network *thickness* (θ) of a graph and prove that the performance of the algorithm guarantees of $O(\theta)$ in broadcast scheduling for an arbitrary network. Another algorithm is presented in [15] which is based on an artificial neural network model. For an $n-$node $m-$timeslot radio network problem, this algorithm requires $(n \times m)$ processing elements that is verified by excessive simulation runs.

Broadcast scheduling algorithms in radio networks are examined by Huson and Sen [16]. Demonstrating the limitations of previous graph representations of network models in these algorithms, they introduce algorithms structured on planar-point graphs in a network model to represent neglected connections between network transceivers. Su et. al [17] consider the problem of minimizing average response time of users in scheduling data broadcasts. They formulate a deterministic dynamic optimization problem to provide optimal broadcast schedule. Due to low implementation complexity, their policy can be generalized into multiple broadcast channels.Hung et. al [18] propose several protocols in multi-hop wireless networks to reduce latency and avoide collisions. Protocols are designed based on Set-Covering scheme and Independent-Transmission-Set scheme to achieve desired objectives.

Maximization of revenue in mobile advertising that uses SMS text messaging is analyzed in [19] by proposing broadcast scheduling system. Given a limited capacity of broadcast time slots, Reyck and Degraeve optimizes the broadcast scheduling system introducing an integer programming model. Developed system reduces the total time required to schedule ad broadcasts and balances the preferences of both retailers and customers solved in multiobjective setting.

Salcedo-Sanz et. al [20] combine Hopfield neural network and a genetic algorithm to develop two stage algorithm for NP-hard broadcast scheduling problem. First stage finds a feasible solution and second stage outputs maximal transmission packing. In this study, the throughput of the system is calculated as $\sum_{tj} s_{tj}$ where s_{tj} denotes the allowance of station j to transmit in time slot t. It is noted that more stations transmitting at the same time slot provide better channel utilization for the radio system. In contrast, Behzad and Rubin [21] claim maximization of cardinality of independent sets does not always increase the throughput of the network by taking into account aggregated effect of interference which is neglected in Protocol Interference Model. Thus, they provide probabilistic analysis of previous graph based algorithms under the assumptions of Protocol Interference Model and show that unacceptable values of SINR at intended receivers are resulted.

Chen et. al [22] propose a low complexity broadcast scheduling algorithm. This study introduces the idea of factor graph. The factor graph strategy enables to use local constraints breaking down the complex task into simple tasks processed in parallel. Representing the graph as a bipartite graph consisting of variable nodes and agent nodes, each variable to be solved is associated with a variable node and each local constraint rule is associated with an agent node. To iteratively optimize the broadcast schedule, the proposed algorithm exchanges the soft-information among agent nodes and variable nodes in the factor graph.

Lee et. al [23] distinguish the interference from collision and provide a constant approximation algorithm that minimizes the makespan. The algorithm using both forward & backward transmissions enhances breadth-first-search (BFS) strategy and guarantees worst case performance $O(\alpha^2)$ for an arbitrary interference factor $\alpha \geq 1$. Wang and Henning [24] propose Deterministic Distributed TDMA Scheduling approach. The aim of this study is to increase energy efficiency of wireless sensor nodes. In this procedure, each node is able to schedule their own TDMA slots depending on the neighborhood information so that collisions are avoided during scheduling. Simulation results indicate better performance of DD-TDMA in schedule length, running time and message complexity.

Huang et. al [25] investigate the broadcast scheduling problem utilizing 2-Disk and the signal-to-interference-plus-noise-ratio (SINR) models. They propose a constant approximation algorithm for both models. The SINR model is explained to give a more realistic and precise analysis since the accumulative interference of nodes outside the interference range is not neglected such that many far-away nodes could still prevent simultaneous transmission due to interference. Menon and Gupta [26] introduce set covering formulation for the broadcast scheduling in packet radio network. This formulation is optimally solved by branch and price algorithm which is based on column generation solution procedure.

Together with regarding conflicts occurring inside the transmission ranges of the nodes, Mahjourian et. al [27] take into consideration other important sources of conflict such as collision at receiver, interference at receiver and contention at sender in parallel transmissions and they develop a conflict-aware network model. Under this network, a constant approximation algorithm to minimize latency in broadcast scheduling is presented and proved.

Minimizing the number of packets that miss the deadline is studied in [28] by Zhan and Xu. Focusing on network coding in broadcast scheduling, they formulate an integer linear programming using a weighted graph model and prove that the problem is NP-hard. The idea behind their proposed algorithm for this problem is to assign vertex weight as a decreasing function of the deadline that yields maximum weight clique in the graph. This algorithm effectively reduces the deadline miss ratio.

Tiwari et. al [29] study interference-aware broadcast scheduling in both 2D and 3D wireless sensor networks. They devise efficient coloring methods and greedy heuristics for both networks. Yeo et. al [30] develops a sequential vertex coloring algorithm for time division multiple access (TDMA) frame in ad-hoc networks. The aim of their study is to find the optimal collision-free time slot schedule that minimizes the system delay and maximizes the slot utilization.

Cheng and Ye [31] contribute to the literature by investigating multicast applications in multihop wireless network. Since the traditional link-based conflict graph model fails to address conflict relation in multicast, they propose a node-based conflict graph model and build a linear programming model to compute the schedule minimizing total network delay. Ji et. al [32] deal with Cognitive Radio Networks (CRNs) studying the Minimum-Latency Broadcast Scheduling problem. Providing a Mixed Broadcasting Scheduling algorithm, they achieve to significantly reduce both broadcast latency and broadcast redundancy.

2.2 Machine Scheduling

Each station served by an access point in wireless networks can be considered as jobs to be processed by single machine. Since we focus on minimum total tardiness and minimum latest completion time objectives further in this paper, machine scheduling in operations research literature is worth to explore. Interfering broadcasts can also be regarded as incompatible job families that cannot be simultaneously processed by the same machine.

There are certain papers discussing minimizing total tardiness objective. Initial group of studies restrict the problem to the case where jobs being sequenced are processed by a single machine. Lawler [33] develops a pseudopolynomial algorithm to minimize total tardiness of jobs processed in a single machine. Each job j has a fixed integer processing time p_j, due date d_j and a positive weight w_j.

Defining completion time of each job as C_j; the weighted tardiness of each job j in a given sequence is equal to $w_j \max(0, C_j - d_j)$. Assumption behind his solution methodology is that jobs are accepted as agreeable meaning that their processing time relations $p_i < p_j$ implies their weighting $w_i \geq w_j$ for the tardiness objective. Satisfying these conditions, a dynamic programming algorithm to find a sequence minimizing total weighted tardiness is developed with worst-case running time of $O(n^4 P)$ or $O(n^5 p_{max})$ where $P = \sum p_j$ and $p_{max} = \max\{p_j\}$. This algorithm is pseudopolynomial because a true polynomial bounded algorithm should be polynomial in $\sum log_2 p_j$. The proposed algorithm contributes to the literature with its running time being bounded by a polynomial function.

Even though Lawler has given a pseudo-polynomial-time algorithm to solve this problem, the question of whether the problem can be solved in polynomial time or it is NP-hard in the ordinary sense remained open throughout more than a decade. Du and Leung [34] achieves to prove that minimizing total tardiness on single machine is an NP-hard problem in the ordinary sense. They first provides the NP-completeness of this restricted Even-Odd Partition problem. Then, describing a reduction from the Restricted Even-Odd Partition problem to the Total Tardiness problem, they manage to demonstrate that the problem equivalently represents a restricted version of the NP-complete Even-Odd Partition problem.

Job interference in our problem is introduced to the literature as in the notion of jobs of different families. Motivated by the restriction for different job families processed together, Mehta and Uzsoy [35] extends the problem into batch processing machine with incompatible job families. The batch processing machine can process up to specific number of jobs simultaneously as a batch, and only jobs from the same family can be batched together as a rule. They prove that the problem of minimizing total tardiness on a single batch processing machine with incompatible job families is NP-hard in the strong sense if the number of families and the batch machine capacity are arbitrary. In addition, they show that a greedy earliest due date algorithm can be utilized to form batches for each job families and they develop a dominance rule to reduce batch sequencing efforts. Based on these results; assuming that all jobs of the same family have identical processing times, they present a dynamic programming algorithm to solve minimum total tardiness problem when batch machine capacity and number of job families are fixed. Similar to our work, they focus on models where jobs in a batch are processed simultaneously. They also outline two heuristic solution procedures providing near optimal solutions in a reasonable time.

Rather than implementing minimum total tardiness objective to the model, some studies tackle job compatibility problem with different objectives. Demange et al. [36] explores the problem as a version of weighted coloring of a graph targeting to find schedules with minimum amount of time needed to complete all jobs. Denoting each job operation as a node $v \in V$ of the graph $G = (V, E)$ having non-negative weights $w(v)$ corresponding to processing times of the jobs, edges of the graph describes the incompatibility constraint between job pairs. The assignment of the jobs to the time slots can be achieved so that no incompatible jobs are included in the same time slot. The number k of time slots required to be used has to be determined as well. As simultaneous processing is allowed between compatible jobs, this resembles our first model formulation where our objective was to find latest completion time. In contrast to our matching principle between noninterfering jobs, they proceed with coloring as a partition of stable sets that define time slots. A k-coloring of $G = (V, E)$ is denoted as a partition $S = (S_1, ..., S_k)$ of node set V of G into stable sets S_i. Therefore, objective for the time slot scheduling problem is to find a k-coloring $S = (S_1, ..., S_k)$ of G such that $w(S_1) + ... + w(S_k)$ being minimized where $w(S_i) = \max \{w(v) : v \in V\}$. Making latest completion time decision is shown to be NP-complete for bipartite graphs. It is also noted that if the bipartite graph takes vertex-weights values from the set $\{1, t\}$ where $t > 1$, the problem could be solved in polynomial time.

As minimizing total tardiness in single machine scheduling known to be NP-hard, Alhawari et al. [37] extends the problem to include non-zero ready times and preemption (interruption of a job by another job) is not allowed. It is assumed that all jobs arrive to the system at different times which are known in advance. In addition to providing a mathematical model to find the sequence of N jobs on a single machine such that the total tardiness is minimized, they propose a Genetic Algorithm approach. This approach enables them to find optimal solution for small problems and near optimal solutions for large problems.

For the chromosome representation, each gene corresponds to the position of a job in the sequence. There can be either delay or non-delay scheduling strategy to be assigned to every job independently so that each gene is represented by a pair of parameters (X, Y), where X denotes the job being assigned and Y shows the scheduling strategy adapted. Fitness function is chosen as the summation of tardiness values from all jobs. To iteratively improve existing solutions, a random seeding of the initial population is employed. Then, fitness function based reproduction probability is assigned to each chromosome to generate parents for mating. Crossover operator is achieved with a single cut point. For the mutation operator, job-based and scheduling strategy-based mutations are introduced. Being directly proportionate to goodness of fitness function values, a probability of being selected is assigned to individuals for the selection strategy. The algorithm terminates with a specified maximum number of generations as a stopping criterion.

Minimization of the total tardiness is discussed in the order scheduling setting by Framinan and Gonzalez [38]. In this setting, customer orders are composed of some product types that can only be produced by different machines. To deal with the problem, they propose a constructive heuristic and following matheuristic strategy. First heuristic assesses the potential contribution of candidate orders to the total tardiness and estimates the contribution of non-scheduled orders to the objective. Second heuristic provides very high-quality solutions based on Job-Position Oscillation procedure. Matheuristic framework is also developed by Mönch and Roob [39] considering batch machine scheduling where incompatible job families cannot be batched together and total weighted tardiness objective is followed. Matheuristic they developed exploits the insight that when batches are assigned and sequenced in machines, remaining batch formation problem can be formulated as a transportation problem. Thus, a biased random-key genetic algorithm is used for the assignment and sequence in a single machine; then parallel batch processing machines case is explained as second application.

2.3 Maximal Matching

Concurrent broadcasts possibility among non-interfering stations gives us consideration of matching principle in graph theory. Since the aim of our study is to utilize simultaneous transmission operation as much as possible, we consider our objective as to find maximum number of matching corresponding to number of concurrent broadcasts. There is plenty of literature studying maximum cardinality matching problem. We restrict the scope to maximum matching problem in bipartite graphs as we will later represent the input network in a bipartite graph structure.

Costa [40] makes the 3-partitioning of the edge set E in an undirected bipartite graph $G = (U, V, E)$ as defining E_1 (1-persistent edges); E_0 (0-persistent edges) and E_w (weakly persistent edges). E_1 has edges included in all maximum matchings, E_0 has edges belonging to no maximum matchings and E_w has edges included in at least one maximum matching but not all of them. The aim of this partitioning is determining the characteristics of the optimal solutions. Based on this classification, Costa developes procedures for finding imperfect and perfect maximum matchings and demonstrates that overall complexity of the procedure is $O(|U||E|)$. The proposed methodology is especially advantageous for the cases where the aim is to find several maximum matchings.

Steiner and Yeomans [41] investigate maximum matchings in convex bipartite graphs. Given an undirected bipartite graph $G = (V_1, V_2, E)$ with node partition sets V_1 and V_2; and edge set E, they provide an $O(n)$ greedy algorithm with $|V_1| = n$. The algorithm consists of three stages. Initial two stages efficiently make preprocessing of the graph. Third stage outputs greedy maximum matching. By reformulating the matching problem as instance of an off-line minimum problem, the solution is found in time linear in n.

Coullard et. al [42] are motivated from selective assembly applications in manufacturing for which they consider several matching problems. Creating two disjoint sets for each component types of an assembly, each matching in a bipartite graph $G = (U \cup V, E)$ corresponds to an "acceptable" assembly. The objective of their problem is to maximize the total number of assemblies produced (yield), which is the maximum cardinality matching in graph representation. In addition to $1-$matching problem, they define $d-$matching problem in which each final assembly has one node in type X matched with a given number of d nodes in type Y. Showing that $d-$ matching is NP-hard problem, linear time greedy algorithm is proposed. Algorithm gives optimal results for bipartite graphs having *cascading* property defined in the study. Optimality proof is made via a min-max theorem.

Different application of matching in bipartite graphs is studied by Wang et. al [43]. As their motivation, they examine the protein molecules that has 3D structure defining their functions. To find the correspondences between structural elements in two proteins, they follow bipartite matching framework. As opposed to max-clique methods formerly used in protein structure alignment, bipartite matching optimally solves in polynomial time.

Chapter 3

Problem Description

In this chapter, we provide the general description for the node selection problem in UFD transmission of medium access control (MAC) protocol design approach. Main objective of node selection process is to implement full duplex technology in wireless networks in an efficient way. The problem setting in wireless network with a sample is first explained and interference illustration in the network is represented. After the problem definition is provided, required mathematical definitions and expressions used to define parameters in our formulations are introduced. Finally, our approach to the problem is summarized.

3.1 Problem Setting

In this section, we demonstrate a simple communication mode used in wireless local area networks. An example is provided in Figure 3.1 illustrating the representation of the wireless communication network. In the scope of our problem, we specifically deal with 802.11 networks in WLANs, but the following illustration is also valid for our case.

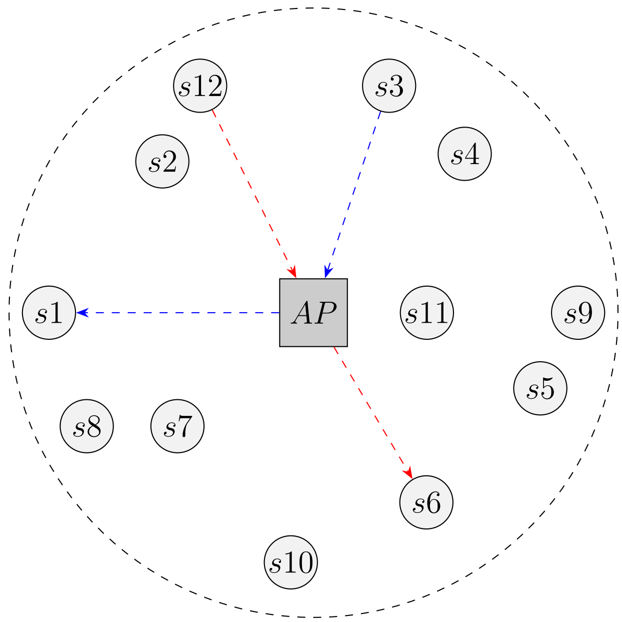

Figure 3.1: Wireless communication network with UFD transmission

As shown in Figure 3.1, there is single access point (AP) and finite number of stations {s1, s2, ...s12} in the network. Access point has the function of connecting devices to a local area network. It can be interpreted as acting as a portal for these devices to achieve the internet connection. One of the most common objective to use access point is to extend the wireless coverage of the communication network in attempt to increase the number of users that can be connected to it. For the example in Figure 3.1, the coverage area of the access point is represented with dashed circle circumscribing around the stations.

Through the wireless connectivity, access point manages to establish transmission links with end-devices using Wi-Fi. For our problem described within wireless local area networks, end-devices can be assumed as computers in the communication network. In the terminology for IEEE 802.11 standards for local area networks, the computer devices are denoted as stations (STA for the abbreviation) having the capability to use the 802.11 protocol. We can also refer to each station as transmitter or receiver depending on its transmission characteristics defined on the network. For instance, nodes corresponding to station 1 (s1) and station 6 (s6) are receivers whereas station 3 (s3) and station 12 (s12) are transmitters for the example in Figure 3.1.

Since our solution approach is developed for the node selection process of the access point, we only consider UFD transmission. This is because in bidirectional transmission (BFD), transmission in two directions occurs between access point and only one station. In UFD mode, property of simultaneous transmit and receive (STR) mode for the next generation WLANs is applied when an access point (AP) supplies from a transmitter and sends demand to another receiver at the same time. Even though only one direction transmissions are completed in UFD, more than one station can be assigned to access point. A simple illustration for UFD transmission can be observed in Figure 3.1 where station 1 (s1) and station 3 (s3) are simultaneously served by access point. Similarly, station 6 (s6) and station 12 (s12) achieve UFD transmission through full duplex.

UFD transmission in Figure 3.1 could be carried out only if $\{s1\}$ does not interfere with $\{s3\}$; similarly non-interference condition is required between nodes $\{s6\}$ and $\{s12\}$. Since some stations can be within the interference range of each other, simultaneous transmission and receive mode could not be achieved for certain nodes in the network.

An example for the interference in UFD transmission is depicted in Figure 3.2. Station i behaves as transmitter when station j is receiver in that case. During simultaneous transmission and receive mode; when station i transmits signal to the access point, station j cannot receive signal from the access point due to the interference within two stations. In that case, either station i or station j could be served in a given time slot. Red arrow in Figure 3.2 represents the interference occurence from station i to station j that hinders UFD transmission. The blue and red circles in Figure 3.2 circumscribing the station nodes refer to the signal coverage area of station i and j respectively. Interference problem is caused by the intersection of these areas. Even though access point has the power to cover both stations, it is not capable to take advantage of full duplex technology for this pair of stations.

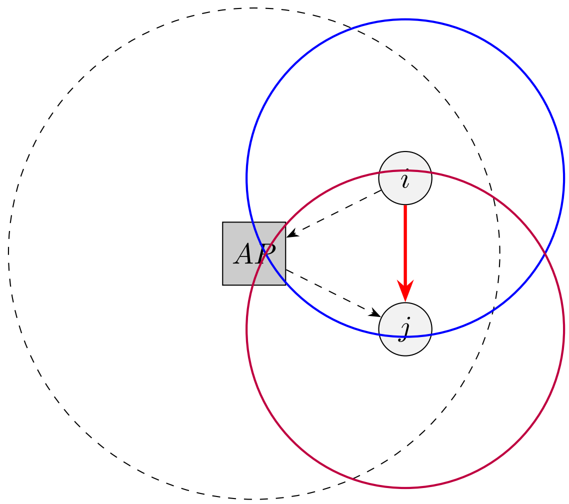

Figure 3.2: Interference problem during UFD transmission

3.2 Preliminaries

Let $\mathcal{V} = \{0, 1, \ldots, N\}$ denote the set of nodes corresponding to the access point and stations in the communication network. We assume that node 0 represents the access point (AP) and the remaining nodes are stations to be served. The cartesian coordinate of each station $i \in \mathcal{V}$ is given as (a_i, b_i) and access point is located at $(0, 0)$. We also denote $d(i, j)$ as the Euclidean distance between node i and node j for all $i, j \in \mathcal{V}$.

As we illustrate in Figure 3.2, stations have coverage area. In this thesis, we define the coverage area in terms of their signal power. We indicate signal power of node j representing the transmission ability as p^j_{sig} for all $j \in V \setminus \{0\}$. To denote the coverage of stations, we assume that signal power is proportional to the following ratio:

$$
\begin{aligned}
p_{\text{sig}}^{j} \quad \sim \quad \frac{1}{d(0,j)^{\delta}} &= \frac{1}{\left(\sqrt{(a_j-0)^2+(b_j-0)^2}\right)^{\delta}} \\
&= \frac{1}{\left(\sqrt{a_j^2+b_j^2}\right)^{\delta}}
\end{aligned}
$$

where δ denotes the path loss exponent which is non-negative and fixed parameter. Path loss symbolizes the reduction in power density of an electromagnetic wave in general. In wireless communication; path loss exponent, the distance between the transmitter and the receiver, and constant value for system losses combined create total path loss that decreases the signal power. For the simplicity, we disregard the constant value for the system losses where we only incorporate the exponent and the distance values. As it can be observed, when $d(0,j)$ which is the distance between access point and node j increases, the signal power of node j decreases. In addition, path loss exponent δ is also inversely proportional to the signal power.

Since the receiver and transmitter nodes simultaneously served by the AP in the UFD transmission must be out of the interference range, we need to define the interference power. Let the interference power at node j while AP receives signal from node i be denoted as p_{int}^{ij} for all $i \in V \setminus \{0\}$ and $j \in V \setminus \{0\}$ where $i \neq j$. As our assumption, we establish the following relationship to define the quantity of interference power:

$$
p_{\text{int}}^{ij} \quad \sim \quad \frac{1}{d(i,j)^{\beta}} = \frac{1}{\left(\sqrt{(a_i-a_j)^2+(b_i-b_j)^2}\right)^{\beta}}
$$

where β denotes the interference path loss exponent which is non-negative and fixed parameter. Clearly, interference power decreases as distance between node i and node j increases. In the communication wireless network, AP should know which node(s) can be eligible to receive signal while node i transmits its signal to AP. This decision of AP depends on whether interference range is violated or not. The criteria for the eligibility is measured by signal to interference ratio. Let SIR^{ij} denote the signal to interference ratio for the transmitter node i and receiver node j for all $i \in V \setminus \{0\}$ and $j \in V \setminus \{0\}$ where $i \neq j$. Then,

$$\begin{aligned} SIR^{ij} &= \frac{p^{j}_{\text{sig}}}{p^{ij}_{\text{int}}} = \frac{\frac{1}{d(0,j)^{\delta}}}{\frac{1}{d(i,j)^{\beta}}} = \frac{d(i,j)^{\beta}}{d(0,j)^{\delta}} \\ &= \frac{\frac{1}{\left(\sqrt{a_j^2+b_j^2}\right)^{\delta}}}{\frac{1}{\left(\sqrt{(a_i-a_j)^2+(b_i-b_j)^2}\right)^{\beta}}} = \frac{\left(\sqrt{(a_i-a_j)^2+(b_i-b_j)^2}\right)^{\beta}}{\left(\sqrt{(a_j^2+b_j^2)}\right)^{\delta}} \end{aligned}$$

There exists a predefined threshold value Ω which denotes the interference range limit. This value is fixed for all SIR values of each $(i, j) \in (V \setminus \{0\}) \times (V \setminus \{0\})$ pair. If the SIR value exceeds the threshold value Ω for receiver i and transmitter j, it indicates that simultaneous receive and transmit mode can be applied successfully. Otherwise the stations are within the interference range of each other and no simultaneous operation is possible.

Table 3.1: Signal to interference ratio requirement for STR Mode

Cases	**Interference**	**STR Mode**	**Node Eligibility**
$SIR^{ij} < \Omega$	Yes	Not Applicable	i or j
$SIR^{ij} \geq \Omega$	No	Applicable	i and j

The summary of interference effect to the system is described in Table 3.1. Based on the criteria above, access point can be interpreted as the decision maker. Since both nodes are eligible to proceed when there is no interference between station pairs, access point might choose to process them simultaneously.

3.3 Our Approach

In this study, we consider formulating our problem based on the objective we want to improve. First, we would like to determine the minimum time required to complete all transmissions of the stations given the interference relations and total signal production of the wireless network. Then, as our second objective, we intend to find the minimum total lateness of the stations given that each station has due dates to complete all transmissions. In that scope, we assume that each transmission occupies 1 timeslot. Since each station can be either transmitter and receiver in the network, we need to complete the transmission assignments in both directions. The assumptions and extensions differ depending on our objectives. Thus, detailed explanations of the formulations will be explained in Chapter 4 and Chapter 5.

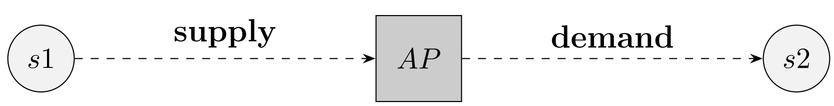

Figure 3.3: Supply and demand definition in the network

Throughout the paper, we express the transmission from the station to an access point as supply operation; and transmission from the access point to the station as demand operation to be completed as in Figure 3.3. The main motivation behind this is to evaluate the general problem from an IE perspective. As our interpretation, we contemplate the problem so as to satisfy these supply and demand requirements of the stations. Depending on the objective and additional parameters, we follow scheduling or matching based formulations in our approach.

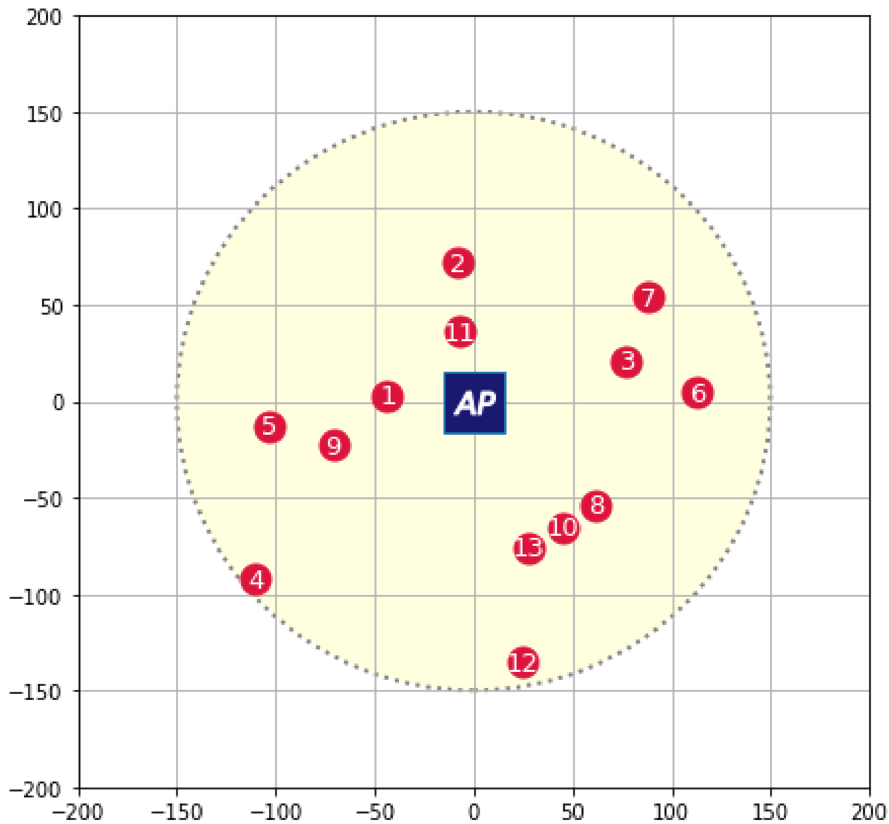

Figure 3.4: Sample locations of stations in Euclidean space

Randomly assigning the locations of the stations on the plane as in Figure 3.4, we calculate SIR values of the stations. Comparing these values, we constitute "interference matrix" that denote the STR mode applicability for each station pairs. When we incorporate the simultaneous supply and demand transmissions into the formulations, we consider single, twice and multiple concurrency extensions in several sections of this paper. Concurrency corresponds to the allowance for the number of signals to be transmitted simultaneously. According the concurrency limit we define in the formulations, we update our interference matrix to extend the concurrency relations among stations.

Chapter 4

Minimizing Latest Completion Time

In this chapter, we study our problem based on the objective of finding minimum latest completion time. At the beginning of each section in this chapter, we introduce concurrency options we have defined for the signal transmissions. Then the methodologies developed for each concurrency alternative used in the formulations and algorithms are explained.

4.1 Single Concurrency

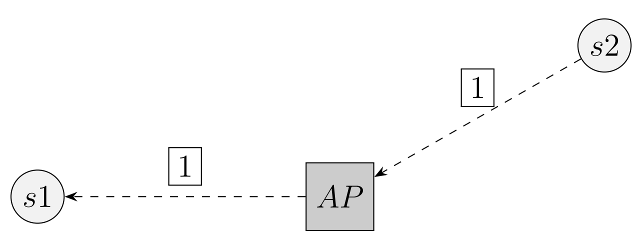

Figure 4.1: Single Concurrency Representation

In this section, single concurrency principle is embraced throughout the following formulations defined. When there is not any interference occurrence provided between stations, stations can broadcast their signals simultaneously: as processed from the access point to a station, and from another station to the access point. Under single concurrency assumption, UFD transmission in full duplex technology is achieved as illustrated in Figure 4.1. Since there is no interference, both transmissions are simultaneously processed without any collision. Since there is single concurrency, they can broadcast at most 1 signal at a time. Hence, the single supply and demand values are satisfied for each station. For instance, $\{s1\}$ station meets 1 demand value and $\{s2\}$ station meets 1 supply value in Figure 4.1 due to single concurrency assumption.

4.1.1 Scheduling Based Formulation

We first consider our problem as a timeslot scheduling problem. In this formulation, the problem can be considered as single machine scheduling problem. Capacity of machine could be assumed to be equal to 2 because of single concurrency assumption. In terms of machine scheduling perspective, jobs of the machine (access point) correspond to supply from a station (transmitter) to an access point and demand from access point to another station (receiver).

We assume that each job operation has the same processing time and we call this duration as "timeslot". We aim to minimize the maximum completion time of all jobs. Hence, the latest timeslot used for a job operation will yield the optimal objective value of our model. Assignment of a job to a timeslot can be made by one of the followings:

i. *There can be only one supply of a station assigned to the timeslot.*

ii. *There can be only one demand of a station assigned to the timeslot.*

iii. *There can be one supply of a station and one demand of another station in the same timeslot as long as they are out of interference ranges of each other.*

Sets

$I = \{1,...,S\}$ set of stations

$T = \{1,...,M\}$ set of timeslots

Parameters

m_i : number of timeslots required for supply of station (i), $i \in I$

n_i : number of timeslots required for demand of station (i), $i \in I$

$$a_{ij} = \begin{cases} 1, & \text{if station } (i) \text{ does not interfere with station } (j), \quad i,j \in I \\ 0, & \text{otherwise} \end{cases}$$

Each station behaves both as a transmitter and a receiver when they can send supply to the access point and can get demand from the access point. Therefore, we define parameter values for number of supplies and for number of demands for each station as above. Since each demand and supply operation has equal processing time as one timeslot duration, the numerical values are given in terms of total number of timeslots.

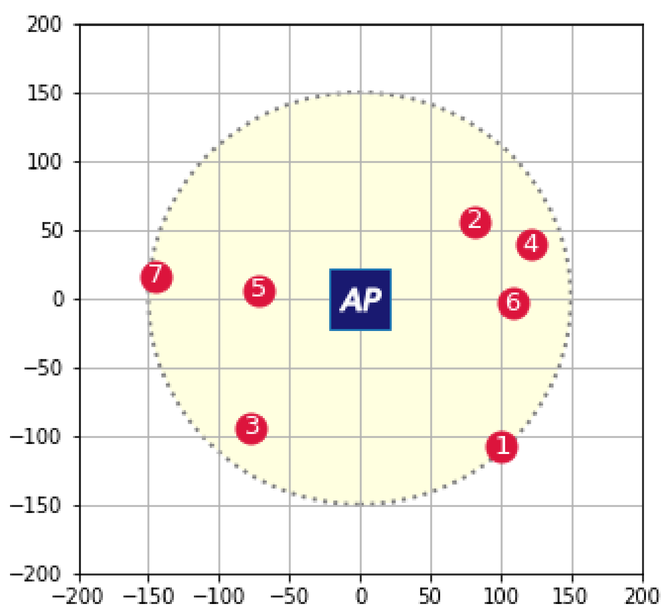

Figure 4.2: Example for the distribution of seven stations in the network

Interference ranges of each station are determined as parameters. First, stations are randomly assigned to the location. Then, for a given path loss exponents δ,β and threshold value Ω; SIR values are calculated based on Euclidean distances to access point and distances among each other as explained in Chapter 3. As an example, let us consider the positions of the following seven stations in Figure 4.2. Given that $\delta = 2.5$, $\beta = 3$ and $\Omega = 30$ for this example, station interference relations are established.

For the simplicity, we convert these relations to a binary parameter to indicate that the simultaneous scheduling is possible or not. If $a_{ij} = 0$, it is not possible to simultaneously schedule supply and demand of different stations. If $a_{ij} = 1$, stations are eligible for simultaneous transmission with 1 signal at a time. Interference matrix A for the example in Figure 4.2 is provided below.

$$A = [a_{ij}]_{7\times 7} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Decision Variables

$$x_{it} = \begin{cases} 1, & \text{if station } (i) \text{ supplies at timeslot } (t), \;\; i \in I, t \in T \\ 0, & \text{otherwise} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{if station } (i) \text{ demands at timeslot } (t), \;\; i \in I, t \in T \\ 0, & \text{otherwise} \end{cases}$$

$z =$ auxiliary variable denoting the latest timeslot to be used

Mathematical Model

$$
\begin{aligned}
\min \quad & z && && (4.1)\\
\text{s.t.} \quad & z \geq x_{it}t && \forall i \in I, t \in T && (4.2)\\
& z \geq y_{it}t && \forall i \in I, t \in T && (4.3)\\
& 1 + a_{ij} \geq x_{it} + y_{jt} && \forall i \neq j \in I, t \in T && (4.4)\\
& \sum_{i \in I} x_{it} \leq 1 && \forall t \in T && (4.5)\\
& \sum_{i \in I} y_{it} \leq 1 && \forall t \in T && (4.6)\\
& x_{it} + y_{it} \leq 1 && \forall i \in I, t \in T && (4.7)\\
& \sum_{t \in T} x_{it} = m_i && \forall i \in I && (4.8)\\
& \sum_{t \in T} y_{it} = n_i && \forall i \in I && (4.9)\\
& x_{it} \in \{0, 1\} && \forall i \in I, t \in T && (4.10)\\
& y_{it} \in \{0, 1\} && \forall i \in I, t \in T && (4.11)
\end{aligned}
$$

The objective function (4.1) denotes the latest timeslot to be used and it is minimized. Constraints (4.2) ensure that latest timeslot cannot be reached before last supply operation is finished. Similarly, constraints (4.3) imply that latest timeslot can be achieved after last demand operation is finished. Constraints (4.4) indicate simultaneous scheduling of supply and demand of different stations cannot be possible if they interfere with each other. It is required to have constraints (4.5) to make sure that no more than one station supply can be assigned to the same timeslot. Constraints (4.6) prevent multiple station demands to appear in the same timeslot. Constraint (4.7) is needed so that no station can have its supply and demand to be assigned to the same timeslot. All supply requirements of stations can be met through constraints (4.8); and all demand requirements of stations can be met through constraints (4.9). Constraints (4.10) and constraints (4.11) are domain constraints for the decision variables of the model.

4.1.2 Greedy Heuristics For Scheduling Model

Even though mathematical model provides optimal solutions and achieves to find minimum latest timeslot, when the number of stations are increased, the solution is not generated in a reasonable time. Time complexity of the mathematical model used in the scheduling based formulation is considerably high mostly because of the 3 different indices in constraints (4.4). In order to reduce the total time to solve the problem, we developed a greedy heuristics. The algorithm defined in this section contributes to the final solution by providing initial number of timeslots to be used in the model as a parameter.

Maximum number of timeslots which is M used in model is simply calculated by adding all of the supply and demand values. However, we realized that this does not take into account any simultaneous processing even if it is possible. In other words, as scheduling based formulation uses M as an upper bound on number of timeslots; it considers the worst case scenario where each station is in the interference range of each other. The idea behind this heuristics is that by paying attention to the interference matrix A, how we can attain an upper bound T for minimum number of total timeslots such that $T \leq M$.

The steps of the algorithm can be briefly explained as follows. We are given supply and demand requirements for each station as the amount of timeslots needed. Starting from the first station without any sorting, algorithm compares the first station's supply value with demand value of other stations in a given order. Restricting to only searching ones where interference does not occur, if the supply value of first station bigger than demand value of another station; algorithm matches these stations and take the bigger value as the total timeslot to cover their supply and demand. For instance, if first station has 6 supplies and second station has 4 demands and they are out of their interference ranges; first 4 slots are used to cover 4 supply and 4 demand values where in each timeslot 1 supply and 1 demand operation is assigned. Updating the bigger supply value by abstracting the smaller demand value, searching is continued if there is enough timeslot left in bigger supply value of a station to include another station's demand value. In that example if another non-interfering station's demand is smaller than or equal to 2, then they are processed simultaneously with remaining supply values of first station. During searching, the finished values are updated as 0 in order not to increase timeslot number unnecessarily. Similarly, if demand value of a station is bigger than supply value of another one, the algorithm chooses demand values as coverage and checks smaller supply values of other stations to include in the same timeslots as long as interference does not occur. The algorithm terminates when all the supply and demand values of each station's searching is completed. At the end, unmatched supply and demand timeslots are summed and added to the matched supply and demand timeslots found in the algorithm.

We introduce certain new notations representing the corresponding additional variables used in the algorithm. Descriptions of these variables can be found in Table 4.1.

Table 4.1: Variables used in scheduling algorithm

Notation	**Variable Definition**	**Type**
T	Total number of timeslots at the end of the algorithm	Integer
L	List of timeslots used only for simultaneous jobs	Integer
S	Limiting variable to prevent overscheduling	Integer
C	Boolean variable to indicate simultaneous scheduling	Binary

Pseudocode of the algorithm is provided below:

Algorithm 1 Greedy Heuristics For timeslot Entry

Initialize: $L \leftarrow [\ ]$
Set: $T \leftarrow 0$
for each $i \in I$ **do**
    $S \leftarrow 0$ ▷ Each station begins with no timeslot used
    $C \leftarrow 0$ ▷ Each station initially has no pair schedule assigned
    if $m_i \geq n_i$ **then**
        for each $j \in I$ **do**
            if $a_{ij} \neq 0$ & $i \neq j$ & $m_i \geq S + n_j$ **then**
                $S \leftarrow S + n_j$ ▷ Add smaller demand value to be scheduled with larger supply
                $n_j \leftarrow 0$ ▷ Remove the scheduled smaller demand value as completed
                $C \leftarrow C + 1$ ▷ Verify the pair as scheduled together
            end if
        end for
        if $C \neq 0$ **then**
            $L \leftarrow L \cup m_i$ ▷ Assign timeslot amount of larger supply value for matched pair
            $m_i \leftarrow 0$ ▷ Remove the scheduled larger supply value as completed
        end if
    else
        for $j \in I$ **do**
            if $a_{ij} \neq 0$ & $i \neq j$ & $n_i \geq S + m_j$ **then**
                $S \leftarrow S + m_j$ ▷ Add smaller supply to be scheduled with larger demand
                $m_j \leftarrow 0$ ▷ Remove the scheduled smaller supply value as completed
                $C \leftarrow C + 1$ ▷ Verify the pair as scheduled together
            end if
        end for
        if $C \neq 0$ **then**
            $L \leftarrow L \cup n_i$ ▷ Assign timeslot amount of larger demand value for matched pair
            $n_i \leftarrow 0$ ▷ Remove the scheduled larger demand value as completed
        end if
    end if
end for
T $\leftarrow \sum_{i \in I} L_i + \sum_{i \in I} (m_i + n_i)$ ▷ Calculate total timeslots to complete all operations

4.1.3 Matching Based Formulation

Considering the objective of our problem as minimizing the latest timeslot, we provide another formulation in this section. Unlike the assignment of stations to timeslots approach in the previous formulation, this formulation is not based on a single machine scheduling problem with machine capacity 2. In this formulation, we use the mathematical discipline of graph theory in attempt to solve the problem in a more time-efficient way. Our motivation for the new formulation is to decrease number of constraints in the previous model.

For each station of the network, there is given number of supply and demand values as explained previously. Since each timeslot is used for operating single supply or single demand or single supply and single demand together when there is no interference; supply and demand copies of each station are formed and represented as nodes in a graph. Before constructing the general network of the formulation, smaller graph representation between two stations is demonstrated below:

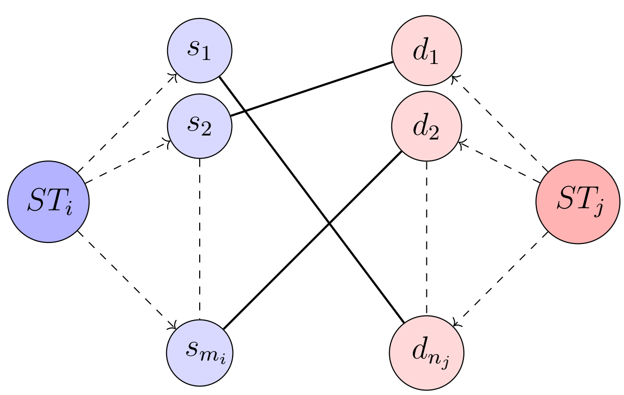

Figure 4.3: Representation of supply and demand nodes between two stations

In Figure 4.3, node ST_i represents station i and nodes $s_1, s_2, \ldots, s_{m_i}$ correspond to the supply nodes of i^{th} station. There is also another station j given as ST_j. Demands of station j are indicated with nodes $d_1, d_2, \ldots, d_{n_j}$. Notice that there are certain edges between supply and demand nodes on the Figure 4.3. Those edges represent the decision of simultaneous scheduling operation of one supply of a station and one demand of another station when there is no interference between corresponding stations. In that case, each edge corresponds to 1 timeslot. Therefore, set of all the edges formed in this graph is a matching because they represent the edges that share no common vertices in an undirected graph.

According to our problem definition, it is stated that no more than one supply or no more than one demand can be assigned in the same timeslot. Therefore, there cannot be any edge in between supplies themselves and in between demands themselves. This leads us to evaluate the network structure as a bipartite graph as we can divide the nodes into two disjoint and independent sets S (supply set) and D (demand set).

In order to know the interference relation between each nodes, we need to define a new matrix in addition to station interference matrix. For instance, let us assume that we have 4 stations and the interference matrix A is the following.

$$A = [a_{ij}]_{4\times 4} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

We are also given that supply values of stations are $[3, 1, 2, 1]$ and demand values are $[4, 1, 2, 2]$ respectively. To form a node to node interference matrix, we can represent each supply node as a row and each demand node as a column.

Preserving the order of the stations, we can expand the matrix A above as follows:

$$B = [b_{kl}]_{7x9} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Drawing from previous implications regarding the structure of the problem, we can convert our problem into similar form of maximum bipartite matching problem. Maximum number of matchings will be the total number of timeslots that are used with simultaneous supply and demand operations. However, there can be unmatched nodes in the graph since some stations can be in the interference range of all other stations thereby not being eligible for simultaneous operation. By this reason, each unmatched nodes will correspond to 1 timeslot and will be added to the number of matched edges to find the latest timeslot to complete all the operations.

Sets

$S = \{1, ..., \sum_{i \in I} m_i\}$ set of supply nodes
$D = \{1, ..., \sum_{i \in I} n_i\}$ set of demand nodes

Parameters

$$b_{kl} = \begin{cases} 1, & \text{if supply node } (k) \text{ does not interfere with demand node } (l), \\ 0, & \text{otherwise } \forall \text{k} \in S, l \in D \end{cases}$$

Decision Variables

$$x_{kl} = \begin{cases} 1, & \text{supply node } (k) \text{ is connected to demand node } (l), \\ 0, & \text{otherwise } \forall \text{k} \in S, l \in D \end{cases}$$

Mathematical Model

$$\min \quad \sum_{k\in S}\sum_{\substack{l\in D\\ b_{kl}=1}} x_{kl} + \sum_{k\in S}(1-\sum_{\substack{l\in D\\ b_{kl}=1}} x_{kl}) + \sum_{l\in D}(1-\sum_{\substack{k\in S\\ b_{kl}=1}} x_{kl}) \tag{4.12}$$

$$\text{s.t.} \quad \sum_{\substack{l\in D\\ b_{kl}=1}} x_{kl} \le 1 \qquad \forall k\in S \tag{4.13}$$

$$\sum_{\substack{k\in S\\ b_{kl}=1}} x_{kl} \le 1 \qquad \forall l\in D \tag{4.14}$$

$$x_{kl} \ge 0 \qquad \forall k\in S, l\in D \tag{4.15}$$

Objective function (4.12) minimizes total number of unmatched nodes and total number of matched edges; equivalently minimizes latest timeslot used. Constraints (4.13) ensure that any supply node in the supply set cannot be matched to more than one demand node in the demand set even if they do not interfere with each other. Constraints (4.14) guarantee the same relation for the demand nodes so that a demand node can be connected to at most one supply node. Constraints (4.15) denote the domain constraints for the decision variables. Note that there is no point in using the integrality constraint since the totally unimodular structure of the constraint matrix ensures that linear programming model provide integral optimal solution.

4.1.4 Hopcroft-Karp Algorithm

Embracing the matching principle to calculate the total number of timeslots required, linear programming model we formulated is carried on with polynomial time algorithm. Hopcroft-Karp algorithm [44] is utilized in this paper in attempt to calculate the total number of matched supply and demand in terms of number of timeslots.

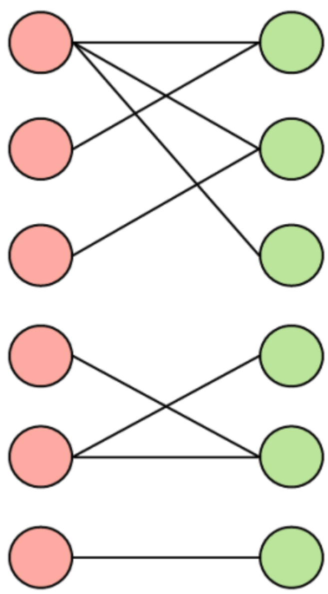

Figure 4.4: Bipartite graph

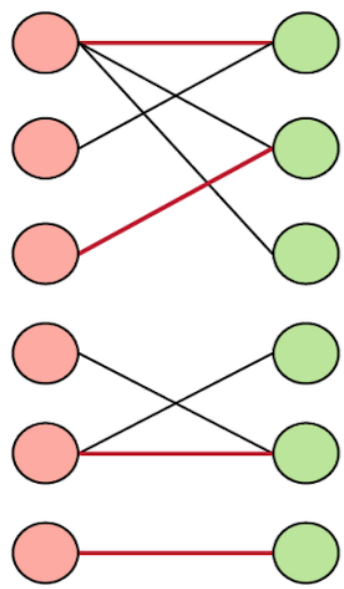

Figure 4.5: Bipartite matching

Hopcroft-Karp algorithm takes input of a bipartite graph. An example of a bipartite graph can be observed in Figure 4.4. Note that red and green nodes are assigned in two disjoint sets so that there is no edge connecting the members of the same set in the figure. The steps of the algorithm proceed with the objective of forming matchings between two sets. A matching in a bipartite graph corresponds to a set of pairwise non-adjacent edges. In Figure 4.5, red lines represent the matched edges in the graph and their union is called as matching.

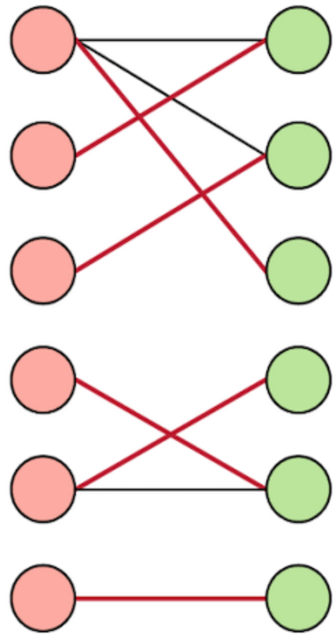

Figure 4.6: Maximum bipartite matching

As an output, the algorithm produces a set that includes as many edges as possible by following the rule that no two edges can share an endpoint, i.e., maximal matching. Figure 4.6 demonstrates the maximal matching for the bipartite graph shown in Figure 4.4. It is very similar to our problem because it is also our motivation to achieve as much as matching between supply and demand possible so that total timeslot number is minimized. Hopcroft-Karp algorithm yields maximum cardinality matching and is known to diminish the time complexity of maximum cardinality matching problem very efficiently. Denoting the set of edges in the bipartite graph as E and set of vertices as V, the algorithm runs in $O\big(|E|\sqrt{|V|}\big)$ time as a worst-case performance. It is an improvement on Ford Fulkerson algorithm [45] whose time complexity is $O\big(|E||V|\big)$ which is also used to solve the maximum cardinality matching problem.

The algorithm has certain phases to be followed but it is better to begin with explaining some of the terminology used in the steps of the algorithm. Names and descriptions used in Hopcroft-Karp Algorithm can be found in Table 4.2 below:

Table 4.2: Terminology of Hopcroft-Karp algorithm

Term	**Description**
Free Vertex	A vertex which is not one of the end-points of some part of a given matching M
Alternating Path	A path of a given matching M whose edges belong alternatively to the matching and not matching; single edges paths are also alternating paths
Augmenting Path	Any alternating path of a given matching M that begins with a free vertex and ends with another free vertex

Given a bipartite graph G, let A and B be the two disjoint sets of this graph over the network $G = (A \cup B, E)$. Any matching between two sets can be represented with M. The algorithm is based on the following two fundamental principle:

- Matching M is not accepted as maximum matching if and only if there exists a candidate augmenting path.
- The algorithm looks for a candidate augmenting path and add the paths found to the current matching.

Outline of the algorithm is described below:

Algorithm 2 Hopcroft-Karp Algorithm

Input: Bipartite graph $G = (A \cup B, E)$

$M \leftarrow \emptyset$
repeat
    Using breadth-first search, build alternating graph rooted at unmatched vertices in A
    Using depth-first search, augment matching M with maximal set of vertex disjoint shortest-length paths
until There are no augmenting paths

Output: Matching $M \subseteq E$

In the initial stage, a breadth-first search is followed to partition vertices into layers. To form the first layer of the partitioning, starting vertices are chosen as the free vertices in set A. For the first search, only unmatched edges going from these vertices to set B are formed due to definition of the free vertices. Then in the second layer, from the vertices in set B, the matched edges to set A are traversed. To alternate between matched and unmatched edges, next layer is formed by searching successors from vertices in A so that only unmatched edges into set B can be traversed. Then from set B to A only matched edges can be traversed. Searching terminates when one or more free vertices are reached. With these steps, an alternating graph rooted at unmatched vertices in A is built.

In the second stage, the procedure continues by utilizing the tree formed by breadth-first search at the previous stage. In this tree, starting from the ending free vertices in set A appeared in the last layer of the tree, augmenting paths are created by moving down the tree until reaching an unmatched vertices in set A appeared at the root of the tree. Depth-first search is used to form augmenting paths that must alternate between matched and unmatched edges. The crucial part of this step is that the augmenting paths should be disjoint, i.e., no common vertex can appear in more than one augmenting path. This stage is finished when there is no more unmatched vertices unused in the tree.

Every path found in the second stage is used to augment the current matching.

As adding these paths to the matching M; the edges of a path that are currently in the matching are removed from matching M and the currently unmatched edges of a path are added to the matching M. This procedure is applied to all the paths found in the second stage and new matching M' is built by enlarging the matching M. Notice that it is ensured that size of M at least increases by 1 at each iteration because any augmenting path added starts and ends with a free vertex whose adjacent unmatched edges are converted to matched edges by aforementioned steps. The algorithm is terminated when there is no more possible augmenting paths found in the breadth first search. In other words, at final matching, there are no augmenting paths so the maximal matching is found.

4.2 Twice Concurrency

In this section, we consider increasing the capability of STR mode in full duplex technology. As an extension to the assumption made in Section 4.1, we develop twice concurrency principle in the formulations defined in this section. Our motivation for this extension is to obtain better minimum latest completion time through higher utilization of simultaneous transmissions.

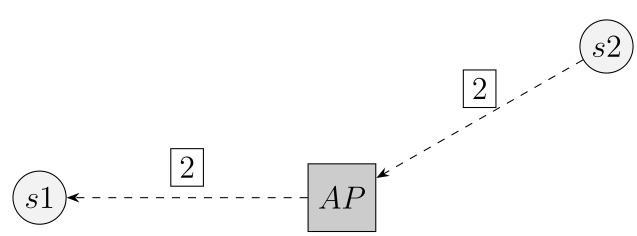

Figure 4.7: Twice concurrency representation

Demonstrating {s1} and {s2} nodes as the stations, the flow numbers on the dashed edges in Figure 4.7 correspond to the maximum number of signals that can be transmitted in a single timeslot. Thus, we can consider the transmission capacity of each station as 2. If the stations are twice concurrent between each other, a timeslot can be used to complete 4 signal transmissions in this case.

Note that the single concurrency assumption depicted in Figure 4.1 can still be followed in practice because any twice concurrent station is also single concurrent by definition.

For the mathematical formulations we develop assuming twice concurrency extension, we follow matching principle between supply and demand nodes instead of timeslot - station assignments used in scheduling based formulation. During a timeslot, if stations can carry out twice concurrency, the completed signal transmissions can be interpreted from matching principle as follows:

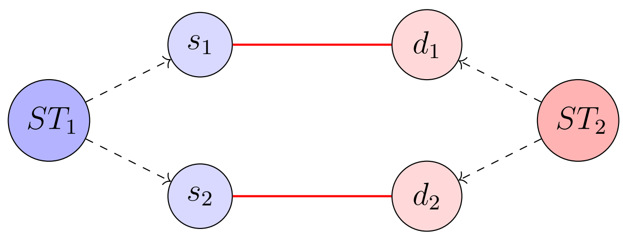

Figure 4.8: Matching supply and demand nodes of twice concurrent stations

Twice concurrent stations are represented in Figure 4.8 above. Even though stations ST_1 and ST_2 have two copy supply and demand nodes in Figure 4.8, stations can have higher or less supply and requirements. In that example, $\{s_1\}, \{s_2\}, \{d_1\}, \{d_2\}$ nodes are processed simultaneously. Matching between $\{s_1\}$ & $\{d_1\}$; and another matching between $\{s_2\}$ & $\{d_2\}$ are performed. It implies that 4 transmissions are finished in the same timeslot.

Station eligibility for twice concurrency is determined by randomly generated interference matrix as a parameter. In addition to the interference matrix described in single concurrency, the entries of the interference matrix are newly defined as $\{0, 1, 2\}$. When the entry of the matrix is 2, both stations can transmit 2 signals at the same timeslot. Including twice concurrency assumption, an instance for the station interference matrix of 7 stations is given below.

$$A = [a_{ij}]_{7\times 7} = \begin{bmatrix} 0 & 1 & 0 & 1 & 2 & 0 & 2 \\ 2 & 0 & 0 & 2 & 2 & 0 & 2 \\ 2 & 1 & 0 & 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 2 & 0 \end{bmatrix}$$

However, since we follow matching principle in the following formulations developed in this section, we extend matrix A into a larger matrix B on which we indicate interference relations between each supply and demand nodes of the stations. For the example matrix A above, supply values of stations are $[2, 3, 2, 1, 3, 2, 3]$ and demand values of stations are $[4, 1, 1, 1, 1, 4, 1]$ respectively. Creating rows for each supply nodes and columns for each demand nodes, the supply-demand interference matrix B is provided below.

$$B = [b_{kl}]_{16\times 13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 2 \\ 2 & 2 & 2 & 2 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 2 & 2 & 1 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 2 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 \end{bmatrix}$$

4.2.1 Two-Indexed Formulation

Sets

$I = \{1, \ldots, n\}$ set of stations

$S = \{1, \ldots, \sum_{i \in I} m_i\}$ set of supply nodes

$D = \{1, \ldots, \sum_{i \in I} n_i\}$ set of demand nodes

Definitions

$s(k)$: station identification for supply $k \in S$ where $s(k) \in I$

$d(l)$: station identification for demand $l \in D$ where $d(l) \in I$

$i(k)$: index number of supply $k \in S$ within its station where $i(k) \in \{1, \ldots, m_{s(k)}\}$

$j(l)$: index number of demand $l \in D$ within its station where $j(l) \in \{1, \ldots, n_{d(l)}\}$

Additional Sets

$S' = \{k \in S \mid i(k) < m_{s(k)}\}$ set of supplies excluding last supplies of stations

$D' = \{l \in D \mid j(l) < n_{d(l)}\}$ set of demands excluding last demands of stations

Decision Variables

$$x_{kl} = \begin{cases} 1, & \text{if } (k^{th}) \text{ supply node is concurrent with } (l^{th}) \text{ demand node} \\ 0, & \text{otherwise} \;\; k \in S,\ l \in D : b_{kl} \neq 0 \end{cases}$$

$$y_{kl} = \begin{cases} 1, & \text{if } (k^{th}) \text{ and } (k+1)^{st} \text{ supplies are concurrent with } (l^{th}) \text{ and} \\ & \quad (l+1)^{st} \text{ demands} \\ 0, & \text{otherwise } k \in S,\ l \in D : b_{kl} = b_{(k+1)(l+1)} = 2, \\ & \quad s(k) = s(k+1),\ d(l) = d(l+1) \end{cases}$$

Mathematical Model

$$\min \quad \sum_{k\in S}\sum_{l\in D} x_{kl} + \sum_{k\in S'}\sum_{l\in D'} y_{kl} + \sum_{k\in S}\Big(1-\sum_{l\in D} x_{kl}\Big) + \sum_{l\in D}\Big(1-\sum_{k\in S} x_{kl}\Big)$$
$$-2\sum_{k\in S'}\sum_{l\in D'} y_{kl} - 2\sum_{l\in D'}\sum_{k\in S'} y_{kl} \tag{4.16}$$

$$\text{s.t.} \quad \sum_{l\in D}(x_{kl}+y_{kl}) \le 1 \qquad k\in S \tag{4.17}$$
$$\sum_{k\in S}(x_{kl}+y_{kl}) \le 1 \qquad l\in D \tag{4.18}$$
$$\sum_{l\in D}(x_{(k+1)l}) + \sum_{l\in D'}(y_{kl}+y_{(k+1)l}) \le 1 \qquad k\in S' \tag{4.19}$$
$$\sum_{k\in S}(x_{k(l+1)}) + \sum_{k\in S'}(y_{kl}+y_{k(l+1)}) \le 1 \qquad l\in D' \tag{4.20}$$
$$x_{kl}, y_{kl} \in \{0,1\} \qquad k\in S, l\in D \tag{4.21}$$

In the objective function (4.16), total number of timeslots used is minimized. Constraints (4.17) enable supply nodes to be either single or twice concurrent with no more than one demand node. Constraints (4.18) allow demand nodes as being concurrent with at most one supply node and hinder them to be single and twice concurrent at the same time. Constraints (4.19) guarantee that if supply node k is twice concurrent, then its consecutive node $(k+1)$ is also twice concurrent in the same timeslot and supply node $(k+1)$ cannot be single concurrent. Constraints (4.20) imply that if demand node l is twice concurrent, then its consecutive node $(l+1)$ is also twice concurrent in the same timeslot and demand node $(l+1)$ cannot be single concurrent. (4.21) denote domain constraints for decision variables.

4.3 Multiple Concurrency

In this section, we study the minimization of latest completion time under multiple concurrency assumption. Multiple concurrency can be regarded as the generalized version of twice concurrency assumption previously explained in Section 4.2. In addition to single concurrency and twice concurrency, stations are also allowed to transmit more than 2 signals in a given timeslot when multiple concurrency is followed.

Whereas there could be at most 1-1 matching in single concurrency and 2-2 matching in twice concurrency between station pairs as long as there is no interference, multiple concurrency assumption enables more utilization of simultaneous transmissions. This option can provide an opportunity for $n - n$ matching of station supply and demands where $n \in \mathcal{N}$.

Unlike the previous assumptions studied, we also include unequal signal transmissions in terms of different quantity of flow amounts between matching pairs. This can be viewed as a relaxation of the previous assumption. In the case of multiple concurrency, there can be different possibilities for the utilization of simultaneous transmission and receive mode. The summary of multiple concurrency possibilities are provided below in Table 4.3.

Table 4.3: STR Mode implementation for multiple concurrency Assumption

	# of Transmissions		
Interference Coefficient	**Station 1 (Supply)**	**Station 2 (Demand)**	**Max Matching**
0	{0} {1}	{1} {0}	N/A
1	{0,1}	{0,1}	1-1 Matching
2	{0,1,2}	{0,1,2}	2-2 Matching
$n \in \mathcal{N}$	{0,1,. . . ,n}	{0,1,. . . ,n}	n-n Matching

As it can be seen in Table 4.3, interference coefficient can take any integer value. When the interference coefficient is 0, it implies that stations cannot transmit and receive in the same timeslot due to interference. In that case, no matching is possible and each signal should be transmitted in different timeslots. "# of transmissions" column corresponds to the signal amount that can be processed in a single timeslot. Among the interfering stations, only one of them can achieve {1} signal transmission when interference coefficient is 0.

As long as the interference coefficient is different than 0, it indicates that matching between supply and demand is possible. 1-1 Matching and 2-2 Matching cases are the maximum matching possibilities when the coefficient is 1 and 2 respectively. 1-1 Matching and 2-2 Matching follow same principle explained in previous sections. However, for instance, the transmission can also take values from the set {0,1,2} when the interference coefficient is 2. That means that {2} supply transmissions can be simultaneously processed with {0} demand transmission. On the other hand, we restrict STR mode to involve at least {1} supply and {1} demand transmission within the same timeslot in previous sections.

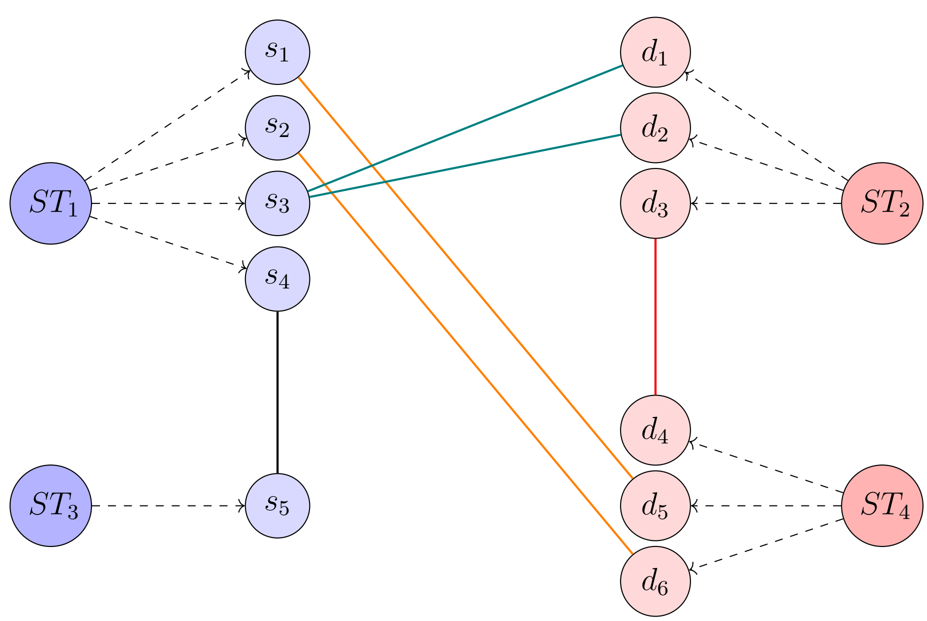

Figure 4.9: Multiple concurrency illustration

Implementing STR mode under multiple concurrency assumption brings more advantages compared to single and twice concurrency. For instance, if there are remaining n supply nodes that cannot be matched with demand nodes, when interference coefficient is nonzero, n supplies can be assigned to single timeslot. This would decrease the latest completion time due to more utilization of STR mode.

To exemplify, an instance of the problem under multiple concurrency assumption is provided in Figure 4.9. Supply and demand nodes matched with the same colored arcs are processed in the same timeslot. Nodes $\{s_1, s_2, d_2, d_3\}$ are simultaneously served. This 2-2 matching can also be observed under twice concurrency assumption since supply and demand nodes of twice concurrent stations are matched. On the other hand, nodes $\{s_3, d_1, d_2\}$ are assigned to another timeslot with 1-2 matching under multiple concurrency assumption, which is not possible under twice concurrency assumption. Similarly, supply nodes $\{s_4, s_5\}$ are served in the same slot whereas $\{d_3, d_4\}$ are served simultaneously in another slot, which is an extension defined for multiple concurrency.

4.3.1 Flow-Based Formulation

Sets and Parameters

$I = \{1, \ldots, n\}$ set of stations

s_k = number of supplies of station k for $k \in I$

d_l = number of demands of station l for $l \in I$

p_{kl} = max number of concurrent broadcasts between station k and l for $k, l \in I$

$A = \{(k, l) : p_{kl} \geq 1, k \in I, l \in I\}$ edge set for possible concurrent broadcasts

$G = (I \cup I, A)$ bipartite network for stations that can concurrently broadcast

Decision Variables

x^i_{kl} = number of k^{th} station supplies and l^{th} station demands that are i times concurrent $i \in \{1, \ldots, p_{kl}\}, (k, l) \in A$

Mathematical Model

$$\max \quad \sum_{(k,l)\in A} \sum_{i=1}^{p_{kl}} (2i-1)x_{kl}^{i} \tag{4.22}$$

$$\text{s.t.} \quad \sum_{l:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le s_k \qquad k \in I \tag{4.23}$$

$$\sum_{k:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le d_l \qquad l \in I \tag{4.24}$$

$$x_{kl}^{i} \ge 0 \qquad (k,l) \in A, i \in \{1,\ldots,p_{kl}\} \tag{4.25}$$

$$x_{kl}^{i} \in \mathbb{Z} \qquad (k,l) \in A, i \in \{1,\ldots,p_{kl}\} \tag{4.26}$$

The objective function (4.22) maximizes total number of concurrent supply and demand transmissions of all stations. Constraints (4.23) ensures that total number of supply transmissions are satisfied. Constraints (4.24) are necessary for meeting total number of demand transmissions. Constraints (4.25) are non-negativity constraints for the flow decision variables. Constraints (4.26) make sure that flow decision variables can only take integer values.

Let P be the feasible region of the formulation above.

$$P = \{ \sum_{l:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le s_k \ , \ \forall k \in I; \sum_{k:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le d_l \ , \ \forall l \in I;$$
$$x_{kl}^{i} \ge 0 \ , \ x_{kl}^{i} \in \mathbb{Z} \ , \ \forall i \in \{1,\ldots,p_{kl}\} \ , \ \forall (k,l) \in A\}$$

Let S be the feasible region of the LP relaxation of the formulation.

$$S = \{ \sum_{l:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le s_k \ , \ \forall k \in I; \sum_{k:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i)x_{kl}^{i} \le d_l \ , \ \forall l \in I;$$
$$x_{kl}^{i} \ge 0 \ , \ \forall i \in \{1,\ldots,p_{kl}\} \ , \ \forall (k,l) \in A\}$$

4.3.2 Extreme Point Properties

Lemma 4.3.1. *In any extreme point solution of S, for any $(k,l) \in A$, it is not possible to have $x^i_{kl} > 0$ and $x^j_{kl} > 0$ for some distinct $i, j \in \{1, \ldots, p_{kl}\}$.*

Proof. Assume to the contrary that there exists an extreme point solution x of S that is not satisfying the property. In other words there exists $(k,l) \in A$, $i, j \in \{1, \ldots, p_{kl}\}, i \neq j$ such that $x^i_{kl} > 0$ and $x^j_{kl} > 0$. Then, the following two vectors y and z are both feasible in S:

$$y^m_a = \begin{cases} x^m_a \text{ for } a \neq (k,l), m \in \{1, \ldots, p_a\} \\ x^m_a \text{ for } a = (k,l), m \neq i, j \\ x^m_a + \epsilon/i \text{ for } a = (k,l), m = i \\ x^m_a - \epsilon/j \text{ for } a = (k,l), m = j \end{cases}$$

$$z^m_a = \begin{cases} x^m_a \text{ for } a \neq (k,l), m \in \{1, \ldots, p_a\} \\ x^m_a \text{ for } a = (k,l), m \neq i, j \\ x^m_a - \epsilon/i \text{ for } a = (k,l), m = i \\ x^m_a + \epsilon/j \text{ for } a = (k,l), m = j \end{cases}$$

for some $0 < \epsilon < \min\{x^i_{kl}, x^j_{kl}\}$. However, this contradicts the extremality of the given x vector since $x = \frac{1}{2}y + \frac{1}{2}z$. $\square$

Definition 4.3.1. For an extreme point x in S, let

$$B_{kl} = \sum_{i=1}^{p_{kl}} (i) x^i_{kl}$$

for any $(k,l) \in A$ over the network $G = (I \cup I, A)$.

Definition 4.3.2. Let the edge set $C \subset A$ be defined as follows:

$$C = \{(k,l) \in A : B_{kl} > 0\}$$

Consider the network $G' = (I \cup I, C) \subset G$ where we discard the edges with zero flow.

Lemma 4.3.2. *In any extreme point solution of S, there is no cycle in the network G', thus network is union of trees.*

Proof. Assume to the contrary that network $G' = (I \cup I, C)$ contains a cycle P in an extreme point solution $x \in S$. Let $\{e_1, e_2, \ldots, e_q\}$ denote the ordering of the edges along the cycle with corresponding vertices $v_1 \xrightarrow{e_1} v_2, \ldots, v_q \xrightarrow{e_q} v_1$. Since G is bipartite, G' is also bipartite as it is subset of G. Thus, the cycle $P = \{e_1, e_2, \ldots, e_q\}$ in G' has even length. By *Lemma 4.3.1*; for any $a = (k, l) \in C$, we know $\exists!\ j_a \in \{1, \ldots, p_a\}$ such that $x_a^{j_a} > 0$. Then, following two vectors are both feasible in S:

$$y_a^m = \begin{cases} x_a^m \text{ for } a \notin P, m \in \{1, \ldots, p_a\} \\ x_a^m \text{ for } a \in P, m \neq j_a \\ x_a^m + \epsilon/m \text{ for } a \in P, \text{ a is odd}, m = j_a \\ x_a^m - \epsilon/m \text{ for } a \in P, \text{ a is even}, m = j_a \end{cases}$$

$$z_a^m = \begin{cases} x_a^m \text{ for } a \notin P, m \in \{1, \ldots, p_a\} \\ x_a^m \text{ for } a \in P, m \neq j_a \\ x_a^m - \epsilon/m \text{ for } a \in P, \text{ a is odd}, m = j_a \\ x_a^m + \epsilon/m \text{ for } a \in P, \text{ a is even}, m = j_a \end{cases}$$

for some $0 < \epsilon < \min_{a \in P}\{x_a^{j_a}\}$. However this contradicts the extremality of the given x vector since $x = \frac{1}{2}y + \frac{1}{2}z$. □

Lemma 4.3.3. *In any extreme point solution of S, considering an arbitrary finite path P with the nodes $u_1 \xrightarrow{e_1} u_2, \ldots, u_{q-1} \xrightarrow{e_{q-1}} u_q$ in the network G', the followings hold:*

(i) If u_1 is supply node and u_q is demand node, at most one of the following is true

$$\sum_{l:(u_1,l) \in C} B_{u_1 l} < s_{u_1} \ , \quad \sum_{k:(k,u_q) \in C} B_{k u_q} < d_{u_q}$$

(ii) If u_1 is demand node and u_q is supply node, at most one of the following is true

$$\sum_{k:(k,u_1)\in C} B_{ku_1} < d_{u_1} \ , \quad \sum_{l:(u_q,l)\in C} B_{u_q l} < s_{u_q}$$

(iii) If both u_1 and u_q are supply nodes, at most one of the following is true

$$\sum_{l:(u_1,l)\in C} B_{u_1 l} < s_{u_1} \ , \quad \sum_{l:(u_q,l)\in C} B_{u_q l} < s_{u_q}$$

(iv) If both u_1 and u_q are demand nodes, at most one of the following is true

$$\sum_{k:(k,u_1)\in C} B_{ku_1} < d_{u_1} \ , \quad \sum_{k:(k,u_q)\in C} B_{ku_q} < d_{u_q}$$

Proof. Let B_a correspond to an extreme point solution of S where $a = (k,l) \in C$ in the network G'. Consider the following two vectors B'_a and B''_a.

$$B'_a = \begin{cases} B_a \text{ for } a \notin P \\ B_a + \epsilon \text{ for } a \in P, \text{ a is odd} \\ B_a - \epsilon \text{ for } a \in P, \text{ a is even} \end{cases}$$

$$B''_a = \begin{cases} B_a \text{ for } a \notin P \\ B_a - \epsilon \text{ for } a \in P, \text{ a is odd} \\ B_a + \epsilon \text{ for } a \in P, \text{ a is even} \end{cases}$$

(i) Assume to the contrary that for an extreme point solution in S, the network G' satisfies both $\sum_{l:(u_1,l)\in C} B_{u_1 l} < s_{u_1}$ and $\sum_{k:(k,u_q)\in C} B_{ku_q} < d_{u_q}$. Then for any $a = (k,l) \in C$ in the network G', two vectors B'_a and B''_a are both feasible in S for some

$$0 < \epsilon < \min_{(k,l)\in P} \{B_{kl}, (s_{u_1} - \sum_{l:(u_1,l)\in C} B_{u_1 l}), (d_{u_q} - \sum_{k:(k,u_q)\in C} B_{ku_q})\}$$

However, this contradicts the extremality of B_a because $B_a = \frac{1}{2}B'_a + \frac{1}{2}B''_a$.

(ii) Assume to the contrary that for an extreme point solution in S, the network

G' satisfies both $\sum_{k:(k,u_1)\in C} B_{ku_1} < d_{u_1}$ and $\sum_{l:(u_q,l)\in C} B_{u_ql} < s_{u_q}$. Then for any $a = (k,l) \in C$ in the network G', two vectors B'_a and B''_a are both feasible in S for some

$$0 < \epsilon < \min_{(k,l)\in P}\{B_{kl}, (s_{u_q} - \sum_{l:(u_q,l)\in C} B_{u_ql}), (d_{u_1} - \sum_{k:(k,u_1)\in C} B_{ku_1})\}$$

However, this contradicts the extremality of B_a because $B_a = \frac{1}{2}B'_a + \frac{1}{2}B''_a$.

(iii) Assume to the contrary that for an extreme point solution in S, the network G' satisfies both $\sum_{l:(u_1,l)\in C} B_{u_1l} < s_{u_1}$ and $\sum_{l:(u_q,l)\in C} B_{u_ql} < s_{u_q}$. Then for any $a = (k,l) \in C$ in the network G', two vectors B'_a and B''_a are both feasible in S for some

$$0 < \epsilon < \min_{(k,l)\in P}\{B_{kl}, (s_{u_1} - \sum_{l:(u_1,l)\in C} B_{u_1l}), (s_{u_q} - \sum_{l:(u_q,l)\in C} B_{u_ql})\}$$

However, this contradicts the extremality of B_a because $B_a = \frac{1}{2}B'_a + \frac{1}{2}B''_a$.

(iv) Assume to the contrary that for an extreme point solution in S, the network G' satisfies both $\sum_{k:(k,u_1)\in C} B_{ku_1} < d_{u_1}$ and $\sum_{k:(k,u_q)\in C} B_{ku_q} < d_{u_q}$. Then for any $a = (k,l) \in C$ in the network G', two vectors B'_a and B''_a are both feasible in S for some

$$0 < \epsilon < \min_{(k,l)\in P}\{B_{kl}, (d_{u_1} - \sum_{k:(k,u_1)\in C} B_{ku_1}), (d_{u_q} - \sum_{k:(k,u_q)\in C} B_{ku_q})\}$$

However, this contradicts the extremality of B_a because $B_a = \frac{1}{2}B'_a + \frac{1}{2}B''_a$. □

Lemma 4.3.4. *In any extreme point solution of S in the network G', any leaf node cannot have an adjacent edge (k,l) with fractional B_{kl} values.*

Proof. Assume to the contrary that in an extreme point solution of S there is a leaf node a with fractional value of B_{ab} where node b is adjacent to node a. Without loss of generality, assume that node a is supply node. Then node b is demand node since G' is bipartite. Since s_a is integer and B_{ab} is fractional, we have $\sum_{l:(a,l)\in C} B_{al} = B_{ab} < s_a$. By *Lemma 4.3.3*, we know that for any other node on this path, s_k and d_l values are satisfied with equality. Then $\sum_{k:(k,b)\in C} B_{kb} = d_b$.

Since B_{ab} is fractional and included in $\sum_{k:(k,b)\in C} B_{kb}$; there must be another node c that is adjacent to b such that B_{cb} is fractional because d_b is integer. By *Lemma 4.3.3*, we know that $\sum_{l:(c,l)\in C} B_{cl} = s_c$. As B_{cb} is fractional and involved in the $\sum_{l:(c,l)\in C} B_{cl}$, there must be another node adjacent to c whose flow is fractional. However, since network is a finite tree by *Lemma 4.3.2*, at the end we will reach another leaf node d with fractional B_{kd} value by this reasoning. This would contradict with *Lemma 4.3.3*, because total flow for node d would not result in equality as s_d being integer. Otherwise, we will reach a visited node with another fractional valued edge which would contradict with *Lemma 4.3.2*. □

Theorem 4.3.5. *In any extreme point solution of S in the network G', B_{kl} values are integral for each edge $(k,l) \in C$.*

Proof. Assume to the contrary that there is an extreme point solution of S that has an edge $(a,b) \in C$ such that B_{ab} is fractional. By *Lemma 4.3.4*, node a and node b cannot be leaf nodes. By *Lemma 4.3.3*, at most one of s_a and d_b is not satisfied with equality. Without loss of generality, let $\sum_{k:(k,b)\in C} B_{kb} < d_b$. We have that $\sum_{l:(a,l)\in C} B_{al} = s_a$ for node a. Since a is not a leaf node, there is at least one edge $(a,c) \in C$ with fractional B_{ac} value since B_{ab} is fractional and $\sum_{l:(a,l)\in C} B_{al} = s_a$. For node c, we have $\sum_{k:(k,c)\in C} B_{kc} = d_c$ by *Lemma 4.3.3*. Since B_{ac} is fractional, there is another node d with fractional B_{dc} value since $\sum_{k:(k,c)\in C} B_{kc} = d_c$. However, since network G' is finite tree, at one point we should reach a leaf node z with fractional B_{kz} value. This contradicts with *Lemma 4.3.4*. □

Proof. (*Alternative*) Let us define a new variable y^i_{kl} as $y^i_{kl} = (i)x^i_{kl}$ for all $i \in \{1,\ldots,p_{kl}\}$ and $(k,l) \in C$. Then, new feasible region S' is defined as

$$S' = \{ \sum_{l:(k,l)\in A} \sum_{i=1}^{p_{kl}} y^i_{kl} \le s_k \ , \ \forall k \in I; \sum_{k:(k,l)\in A} \sum_{i=1}^{p_{kl}} y^i_{kl} \le d_l \ , \ \forall l \in I;$$

$$y^i_{kl} \ge 0 \ , \ \forall i \in \{1,\ldots,p_{kl}\} \ , \ \forall (k,l) \in A\}$$

Then, we can update objective function as

$$\max \quad \sum_{(k,l)\in A} \sum_{i=1}^{p_{kl}} \left(2 - \frac{1}{i}\right) y_{kl}^i$$

Since the coefficient of $(2 - 1/i)$ is a constant value; LP model can be rewritten as

$$\max \ \{cy \ | \ y \geq 0, \ My \leq b\}$$

where $c = (2-1/i)$, $b = [s \ \ d]^T$ and M is the constraint matrix for the coefficients in the model. Note that b is integral since s_k is integer $\forall k \in I$ and d_l is integer $\forall l \in I$. We also know that M is a totally unimodular matrix because determinant of each square submatrix of M takes values from the set $\{-1, 0, 1\}$. Therefore, any extreme point solution y is integral. It implies that $y_{kl}^i = (i)x_{kl}^i$ is integer $\forall i \ \in \{1, \ldots, p_{kl}\}$ and $\forall (k,l) \in C$. As a conclusion $B_{kl} = \sum_{i=1}^{p_{kl}} (i)x_{kl}^i$ is integer for any $(k,l) \in C$. □

Theorem 4.3.6. *In any extreme point solution of S in the network G', for any x_{kl}^i value, following holds for some $j \in \mathbb{Z}$:*

$$x_{kl}^i = \left(\frac{1}{i}\right) j$$

Proof. Let x be an extreme point solution of S. By *Theorem 4.3.5*, $B_{kl} = \sum_{i=1}^{p_{kl}} (i)x_{kl}^i$ for any $(k,l) \in C$ is integer in the network G'. By *Lemma 4.3.1*, we know for any $(k,l) \in C$; $x_{kl}^r > 0$ for some $j \in \{1, \ldots, p_{kl}\}$ and $x_{kl}^t = 0$ for all $t \in \{\{1, \ldots, p_{kl}\} \setminus \{j\}\}$. Thus, $B_{kl} = (j)x_{kl}^j$. Since B_{kl} is integer by *Theorem 4.3.5*, there exists an integer $a = B_{kl}$ such that $x_{kl}^j = \left(\frac{1}{j}\right) a$. □

Theorem 4.3.7. *Following expression is valid for any $x \in P$*

$$x_{kl}^i \leq \left\lfloor \frac{\max\{s_k, d_l\}}{i} \right\rfloor \qquad \forall (k,l) \in A, i \in \{1, \ldots, p_{kl}\}$$

Proof. Let x be an arbitrary feasible vector in P. Then, by definition of P;

$$\sum_{l:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i) x_{kl}^{i} \leq s_k \ , \ \forall k \in I$$

$$\sum_{k:(k,l)\in A} \sum_{i=1}^{p_{kl}} (i) x_{kl}^{i} \leq d_l \ , \ \forall l \in I$$

These imply that $(i)x_{kl}^i \leq s_k$ and $(i)x_{kl}^i \leq d_l \ , \ \forall (k,l) \in A \ , \ i \in \{1, \ldots, p_{kl}\}$.

Then, $(i)x_{kl}^i \leq \max\{s_k, d_l\}$; and thereby $x_{kl}^i \leq \frac{\max\{s_k,d_l\}}{(i)} \quad \forall (k,l) \in A \ , \ i \in \{1, \ldots, p_{kl}\}$. Since $x \in P$, we know that x can take only integer values. Thus, the assertion follows. $\square$

Theorem 4.3.8. *Following expression can be added to the LP relaxation of the flow model as an optimality cut:*

$$\sum_{i=1}^{p_{kl}-1} x_{kl}^{i} = 0 \qquad \forall (k,l) \in A \tag{*}$$

Proof. Let $x \in S$ be an optimal solution for the LP relaxation of the flow model. By *Lemma 4.3.1*, we know that for any $(k,l) \in A$ with $\sum_{i=1}^{p_{kl}} x_{kl}^i > 0$, we have that $\sum_{i=1}^{p_{kl}} x_{kl}^i = x_{kl}^{c_{kl}}$ for some c_{kl} value with $1 \leq c_{kl} \leq p_{kl}$. Then the optimal objective value of the LP relaxation is $\sum_{(k,l)\in A}(2c_{kl} - 1)x_{kl}^{c_{kl}}$.

Consider adding the equality (*) to the formulation of the LP relaxation. Let $\overline{x}$ denote the optimal solution of this formulation. Then, optimal objective value is equal to $\sum_{(k,l)\in A}(2p_{kl} - 1)\overline{x}_{kl}^{p_{kl}}$. Let us choose $\overline{x}_{kl}^{p_{kl}}$ values as follows:

$$\overline{x}_{kl}^{p_{kl}} = \frac{x_{kl}^{c_{kl}} c_{kl}}{p_{kl}}$$

where x denotes the optimal solution of the LP relaxation model. Note that $\overline{x}$ is feasible for the new formulation. Observe also that $\overline{x}$ is feasible to the LP relaxation model as well, so $\overline{x} \in S$. The objective value of LP relaxation for the solution $\overline{x}$ is equal to $\sum_{(k,l)\in A}(2c_{kl} - 1)x_{kl}^{c_{kl}}$. Since this value gives the same result as the optimal solution x has; we conclude that $\overline{x}$ is also an optimal solution for the LP relaxation. Thus, the assertion follows. $\square$

In the light of *Theorem 4.3.8*, after adding the optimality cut to the LP relaxation of the flow model; we can provide the equivalent formulation to the model as follows.

$$
\begin{aligned}
\max \quad & \sum_{(k,l)\in A} (2p_{kl} - 1)x_{kl} & \\
\text{s.t.} \quad & \sum_{l:(k,l)\in A} (p_{kl})x_{kl} \leq s_k & k \in I \\
& \sum_{k:(k,l)\in A} (p_{kl})x_{kl} \leq d_l & l \in I \\
& x_{kl} \geq 0 & (k,l) \in A
\end{aligned}
$$

Note that with this formulation above, the we can use the two indexed decision variable. In the remaining parts of this thesis, we name this mathematical formulation as the LP flow model with optimality cut.

Chapter 5

Minimizing Total Tardiness

In this chapter, we consider integrating deadline extension for each station in the network. According to the due dates to complete all the transmissions provided for each station, we aim to minimize total lateness of the broadcasts as our motivation. The tardiness of each job operation is defined to be the difference between the completion time and the due date. As due dates are given as parameters, we develop our methodology to calculate completion times of transmissions. For the simultaneous transmit and receive mode property, we only consider single concurrency assumption for this objective.

5.1 Model Formulation

Sets

$I = \{1,...,S\}$ set of stations

$T = \{1,...,M\}$ set of timeslots

Parameters

m_i : number of timeslots required for supply of station (i), $i \in I$

n_i : number of timeslots required for demand of station (i), $i \in I$

d_i : deadline of completion time for station (i), $i \in I$

$$a_{ij} = \begin{cases} 1, & \text{if station } (i) \text{ does not interfere with station } (j), \ \ i, j \in I \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables

$$x_{it} = \begin{cases} 1, & \text{if station } (i) \text{ supplies at timeslot } (t), \ \ i \in I, t \in T \\ 0, & \text{otherwise} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{if station } (i) \text{ demands at timeslot } (t), \ \ i \in I, t \in T \\ 0, & \text{otherwise} \end{cases}$$

$c_i =$ completion time for station (i), $i \in I$

$s_i =$ difference between deadline and completion time of station (i), $i \in I$

Mathematical Model

$$
\begin{aligned}
\min \quad & \sum_{i \in I} s_i && && (5.1)\\
\text{s.t.} \quad & \sum_{t \in T} x_{it} = m_i && \forall i \in I && (5.2)\\
& \sum_{t \in T} y_{it} = n_i && \forall i \in I && (5.3)\\
& \sum_{i \in I} x_{it} \leq 1 && \forall t \in T && (5.4)\\
& \sum_{i \in I} y_{it} \leq 1 && \forall t \in T && (5.5)\\
& x_{it} + y_{jt} \leq a_{ij} + 1 && \forall i \neq j \in I, t \in T && (5.6)\\
& c_i \geq x_{it} t && \forall i \in I, t \in T && (5.7)\\
& c_i \geq y_{it} t && \forall i \in I, t \in T && (5.8)\\
& c_i \leq d_i + s_i && \forall i \in I && (5.9)\\
& x_{it} \in \{0, 1\} && \forall i \in I, t \in T && (5.10)\\
& y_{it} \in \{0, 1\} && \forall i \in I, t \in T && (5.11)\\
& c_i \geq 0 && \forall i \in I && (5.12)\\
& s_i \geq 0 && \forall i \in I && (5.13)
\end{aligned}
$$

The objective function (5.1) minimizes the total tardiness of supply and demand operations of stations in the network. Constraints (5.2) ensure that all supply operations of each station are assigned to the available timeslots. Constraints (5.3) yield all demand operations of each station are assigned to the available timeslots. In (5.4), it is implied that each supply of stations can be assigned to single timeslot. Constraints (5.5) forbid each demand of stations to be assigned to multiple timeslots. Simultaneous supply and demand operations are enabled by (5.6) when there is no interference between stations. Both (5.7) and (5.8) indicate the completion time of stations. Constraints (5.9) calculate the tardiness amount of completion times of the stations from the deadlines. Constraints (5.10)-(5.13) are the domain constraints for the decision variables.

5.2 Deadline Algorithm

In this section, we demonstrate our algorithm for the problem under deadline restriction. The algorithm is proposed to find feasible solution with less time complexity compared to mathematical model. Each station has a predetermined deadline to complete all supply and demand operations.

Before the algorithm starts, each station is randomly located on the cartesian coordinate system. Access point is always positioned at origin $(0, 0)$ in the system. Calculating distance between stations and access point, a distance matrix is created. Based on Euclidian distances $d(i, j)$ and given path loss exponent δ and interference path loss exponent β values, signal to interference ratio is calculated for each station. Station interference matrix is formed based on signal to interference ratios.

Similar to the procedure applied in 4.1.3, station interference matrix A is used to create supply-demand copy matrix B. We also define x_{it} and y_{it} variables as in 4.1.1 to assign supply and demand to timeslots. In addition, we introduce certain parameters used in the algorithm shown in Table 5.1.

Table 5.1: Parameters used in deadline algorithm

Notation	**Parameter Definition**
ls^i	row index in matrix B for the first supply of station i
us^i	row index in matrix B for the last supply of station i
ld^i	column index in matrix B for the first demand of station i
ud^i	column index in matrix B for the last demand of station i

For a given station set I, we define another station set V. In this set, stations are ordered in non-decreasing values of the following:

$$\{d_i - (m_i + n_i)\} \quad \forall i \in I$$

The main idea behind this way of sorting is to give priority to process the stations that have the smallest gap between summation of supply and demand operations, and the deadline restriction. Following this ordering, algorithm chooses first station and matches its supply values with demand values of all other stations if there is no interference. In this part, searching is performed by fixing a row interval corresponding to copy supply values of the station and moving along the columns of matrix B to check the interference relation.

After completing all the possible matching of its supplies, algorithm fixes the column interval of matrix B corresponding to the demand values of the station; and searches along the rows of matrix B. As long as there is no interference and there are demand values unmatched, the algorithm matches demand values of the station with all available supply values of other stations. Each matching increases latest timeslot by 1.

Once the process described above is repeated for all stations in the network, remaining unmatched supply and demand values are distributed to individual timeslots and latest timeslot is updated by adding these timeslots. An example result of the algorithm and mathematical model from the same instance can be observed on Gantt charts in Figures 5.1 and 5.2. In that example, mathematical model finds optimal total tardiness value of 0 whereas our algorithm results in 8 as four of the stations are tardy and could not meet their deadline.

In Figures 5.1 and 5.2, star sign indicates the deadline timeslot that corresponding station on y-axis should complete all of its transmissions. We also provide dashed lines on the Gantt charts, which correspond to the optimal minimum latest completion times found by Hopcroft-Karp algorithm. It is notable to mention that mathematical model can find a schedule with 0 total tardiness value, and also achieves to complete all transmissions with the optimal minimum makespan value.

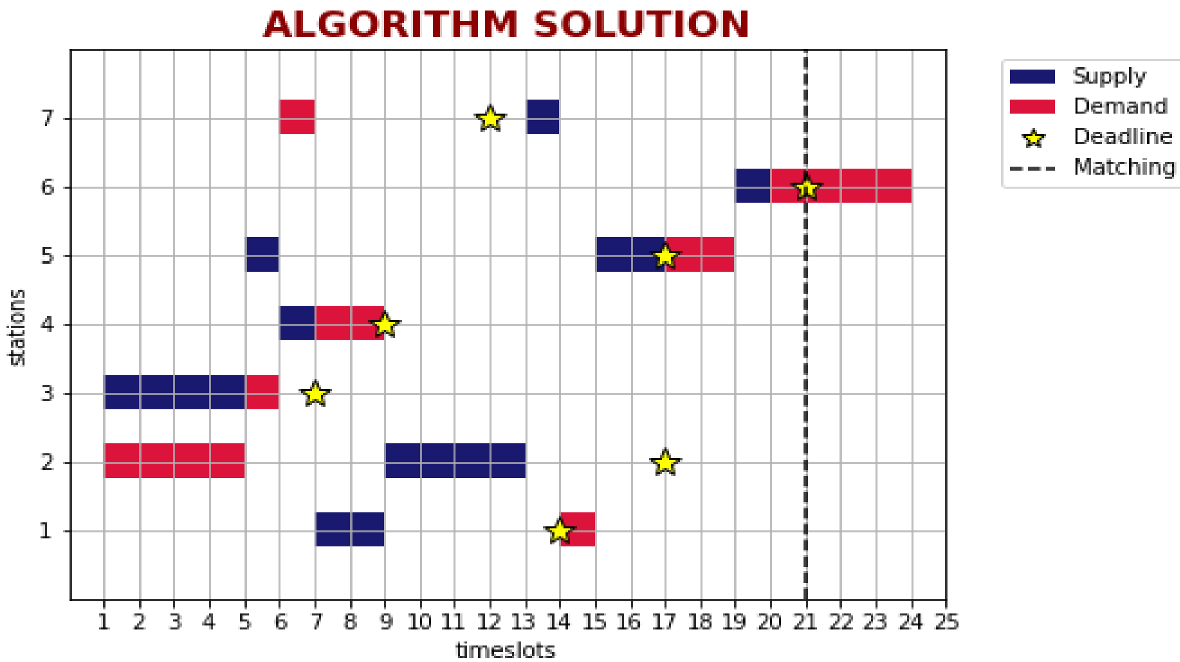

Figure 5.1: Deadline algorithm solution schedule of an instance with 7 stations

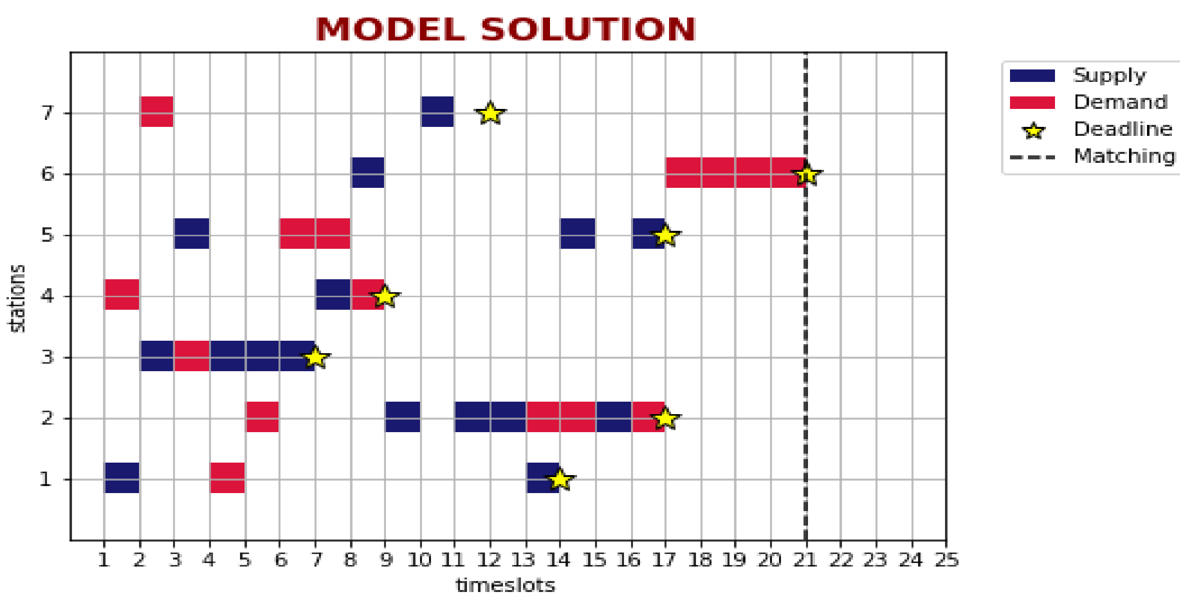

Figure 5.2: Deadline model solution schedule of an instance with 7 stations

Pseudocode of the algorithm is provided below.

Algorithm 3 Deadline Algorithm

Initialize: $t \leftarrow 0$
Set: $M \leftarrow \sum_{i \in I}(m_i + n_i)$
for each $i \in I$ **do**
    for each $j \in I$ **do**
        if $i \neq j$ **then**
            for $m \in [ls^i, us^i]$ **do**
                for $n \in [ld^i, ud^i]$ **do**
                    if $B_{mn} = 1$ **and** $\sum_{t=1}^{M} y_{jt} < n_j$ **and** $\sum_{t=1}^{M} x_{it} < m_i$ **then**
                        $x_{it} = 1$; $y_{jt} = 1$; $t \leftarrow t + 1$
                    end if
                end for
            end for
        end if
    end for
    for each $k \in I$ **do**
        if $i \neq k$ **then**
            for $n \in [ld^i, ud^i]$ **do**
                for $m \in [ls^k, us^k]$ **do**
                    if $B_{mn} = 1$ **and** $\sum_{t=1}^{M} y_{it} < n_i$ **and** $\sum_{t=1}^{M} x_{kt} < m_k$ **then**
                        $x_{kt} = 1$; $y_{it} = 1$; $t \leftarrow t + 1$
                    end if
                end for
            end for
        end if
    end for
end for
for $i \in I$ **do**
    while $\sum_{t=1}^{M} x_{it} < m_i$ **do**
        $x_{it} = 1$; $t \leftarrow t + 1$
    end while
    while $\sum_{t=1}^{M} y_{it} < n_i$ **do**
        $y_{it} = 1$; $t \leftarrow t + 1$
    end while
end for

5.3 NP-Completeness

In this section, we prove that minimizing total tardiness in broadcast scheduling is NP-complete by demonstrating that our problem can be reduced to a known NP-complete problem. We will use proof by restriction considering a special case of our problem.

Lemma 5.3.1. *For a given β, δ and arbitrarily located stations $i, j \in V \setminus \{0\}$ on Euclidean Space, there can always be found a threshold value Ω satisfying the following:*

$$\frac{d(i,j)^{\delta}}{d(0,j)^{\beta}} < \Omega \tag{5.14}$$

Proof. Let β, δ and location of stations $i, j \in V \setminus \{0\}$ on the Euclidean Space be arbitrary. Let us define the following parameter for all $j \in V \setminus \{0\}$:

$$x_j = \max_{i \in V \setminus \{0\}} \left\{ d(i,j)^{\delta} \right\}$$

Note that x_j values can easily be found by calculating distance from station j to the farthest point to station j. Since the value of x_j gives the maximum distance from station j, one can clearly observe that the following holds:

$$\frac{d(i,j)^{\delta}}{d(0,j)^{\beta}} \leq \frac{x_j}{d(0,j)^{\beta}}$$

Since we now found an upper bound for the left hand side, let us calculate the following and denote the result with an expression $M \in \mathbb{R}$:

$$M = \max_{j \in V \setminus \{0\}} \left\{ \frac{x_j}{d(0,j)^{\beta}} \right\}$$

Notice that M is a constant number and can be calculated for any instance of our problem as there is finite number of stations on the Euclidean Space. We can also observe that M is an upper bound on the following:

$$\frac{d(i,j)^{\delta}}{d(0,j)^{\beta}} \leq \frac{x_j}{d(0,j)^{\beta}} \leq M$$

Since M is a finite number and it always exists, there always exists another finite number Ω such that the following holds:

$$\frac{d(i,j)^{\delta}}{d(0,j)^{\beta}} \leq \frac{x_j}{d(0,j)^{\beta}} \leq M < \Omega$$

Thus, our claim directly follows and we can conclude that there is always an Ω that satisfies (5.14). $\square$

Evaluating the general problem from machine scheduling perspective, each station $i \in V$ can be considered as task to be processed by a batch processing machine; the access point in our case. In general, it can be interpreted as in the following instance.

INSTANCE: Set V of tasks, each having processing time $p_i = m_i + n_i$, single machine with batch processing capacity $n \in \mathcal{N}$, a positive integer T, for each task $i \in V$ a deadline $d_i \in \mathbb{Z}^+$, and for each task pair $(i,j) \in V \times V$ an indicator parameter $a_{ij} \in \{0,1\}$ denoting "compatibility relations" to be satisfied for batch processing.

QUESTION: Is there schedule S for set of tasks V processed by single machine with batch capacity n that obeys the compatibility relations and tasks having completion times c_i and tardiness s_i satisfying $c_i \leq d_i + s_i$ for all $i \in V$ such that $\sum_{i \in V} \max\{0, s_i\} \leq T$?

Theorem 5.3.2. *Minimizing total tardiness in broadcast scheduling problem under single concurrency is NP-complete.*

Proof. Let $V = \{0, 1, \ldots, n\}$ denote the set of nodes in the telecommunication network where $\{0\} \in V$ corresponds to the access point and $d(i,j)$ denote the distance between station i and j. For a given path loss exponent β, the interference path loss exponent δ; consider the restricted case where the threshold value Ω is chosen such that it satisfies (5.14) by Lemma 5.3.1.

Satisfying the relation (5.14) directly implies that $a_{ij} = 0$ for all $i \in V \setminus \{0\}$ and $j \in V \setminus \{0\}$. In other words, by choosing a specific threshold Ω, we can restrict our compatibility matrix A to be zero matrix so that all operations are incompatible among each other.

Since $a_{ij} = 0$ for all $i \in V \setminus \{0\}$ and $j \in V \setminus \{0\}$, batch size should be equal to 1 for each timeslot due to incompatibility. Note that with this restriction, the problem becomes single machine scheduling with minimizing total tardiness. Du and Leung [34] proved that single machine scheduling with minimizing total tardiness is NP-complete. Since restricted case of our problem is equal to an NP-complete problem, we conclude that our general problem is NP-complete.

$\square$

Chapter 6

Computational Experiments

In this chapter, we demonstrate the outputs of our computational study. In this study, we conduct several experiments where we mainly analyze the results of mathematical formulations and algorithms that we developed in our solution methodology. All of computational experiments have been performed on macOS with 1.4 GHz Quad-Core Intel Core i5 processor. Coding of the formulations and algorithms have been carried out utilizing Python programming language, and solved with a connection to IBM ILOG CPLEX Optimization Studio 21.1.0 Beta Version.

In the beginning of this chapter, we explain our data generation process to construct the general setting we used before conducting the experiments in Section 6.1. Afterwards, we present a detailed comparison of the performances of all the mathematical formulations and algorithms in terms of their solving times and their final solution values in Section 6.2. Finally, we conduct sensitivity analysis for the flow model formulations in Section 6.3. In this analysis, we use different parameter values for the inputs of the problem and aim to gain several insights regarding the effects of the setting of parameters on the performance of the formulations. The detailed results for 50 randomly generated instances for each solution methodology can be found in Appendix A.

6.1 Data Generation

Before we implement our solution methodology into the optimization software, we create random instances for the general problem setting. The intervals for the number of stations and the radius of the network are randomly determined for each instance.

As described on the example in the Chapter 3; after we assign the location of the access point to the origin on the Euclidean space as the center of the circle, we generate the random polar coordinates for the positioning of each station within the circle circumscribing the origin. For the data generation, we input the following parameters to the generator shown in Table 6.1.

Table 6.1: Input parameters used to generate random data points for the station locations in the network

Input parameters	Symbol & value used
Number of stations in the network	$\|S\|$
Minimum distance between stations	d
Circle radius value	R
Circle center coordinate	$C = (0, 0)$
Number of maximum trials	$t = 10000$
Generation time limit	3600 seconds

Sampling of the locations is randomly generated based on the uniform distribution of the polar coordinates (r, θ) within predetermined circle where $r \in (0, R]$ and $\theta \in [0, 2\pi]$. In that principle; for each random instance it is assured that no more than one station is assigned on the same coordinate and we also consider adding restriction of the minimum distance requirement between each station, denoted as d where $0 < d \leq R$.

Since data generation can take long duration, we limit the time duration and number of trials in terms of iteration count for the location assignments, as in

Table 6.1.

In order to use the distances $d(x, y)$ based on Euclidean metrics, polar coordinates of each station are transformed into cartesian coordinates $(x, y) \in \mathbb{R}^2$ using the following formula:

$$x = r \times \cos\theta$$
$$y = r \times \sin\theta$$

Using those cartesian coordinates of the station set and given δ,β and Ω values, interference matrix of the stations that denote the possible concurrent transmissions is calculated as indicated in Chapter 3. Since we perform computational experiments both for single concurrency and multiple concurrency; cardinality of Ω value varies based on the maximum concurrency number. We consider evenly spaced sequences in intervals we specified for Ω values. As an example of Ω arrays produced for multiple concurrency; Table 6.2 can be observed below:

Table 6.2: Example for Ω output based on interval input

Maximum concurrency	Interval bounds for Ω	Ω array produced
7	{5.0, 10.0}	[5.0 5.83 6.66 7.5 8.33 9.16 10.0]
6	{2.5, 7.5}	[2.5 3.5 4.5 5.5 6.5 7.5]
5	{4.5, 8.5}	[4.5 5.5 6.5 7.5 8.5]
4	{4.0, 9.5}	[4.0 5.83 7.66 9.5]
3	{2.0, 10.0}	[2.0 6.0 10.0]
2	{3.0, 6.5}	[3.0 6.5]

6.2 Performance Comparison

In this section, we provide a detailed comparison of the performances of mathematical formulations and algorithms in terms of their CPU times and objective values calculated through the software. Initially, we analyze the performance of scheduling based formulation and the greedy heuristics we developed for the single concurrency case that minimize the latest completion time (makespan).

Table 6.3: Scheduling algorithm & model output for different radius values when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ and minimum distance = 30

Parameters		Minimum Makespan Values			CPU Times (s)		
# of STAs	Circle Radius	Greedy Heur.	Sch. Model	Decrease	Greedy Heur.	Sch. Model	Difference
35	500	95	94.0	1.05%	0.0007	7.1681	7.17
	600	105	94.0	10.48%	0.0008	7.3890	7.39
	700	105	94.0	10.48%	0.0007	6.7607	6.76
	800	114	94.0	17.54%	0.0005	8.3629	8.36
	900	123	94.0	23.58%	0.0006	1.3083	1.31
40	500	103	95.0	7.77%	0.0009	12.2618	12.26
	600	104	95.0	8.65%	0.0010	9.6317	9.63
	700	108	95.0	12.04%	0.0008	14.7765	14.78
	800	127	95.0	25.20%	0.0008	8.5628	8.56
	900	140	95.0	32.14%	0.0006	11.6946	11.69
45	500	136	125.0	8.09%	0.0016	113.9754	113.97
	600	135	125.0	7.41%	0.0017	27.3339	27.33
	700	145	125.0	13.79%	0.0008	23.1478	23.15
	800	169	125.0	26.04%	0.0007	21.9032	21.90
	900	172	125.0	27.33%	0.0006	20.4312	20.43
50	500	136	135.0	0.74%	0.0013	24.5389	24.54
	600	136	135.0	0.74%	0.0013	31.2396	31.24
	700	141	135.0	4.26%	0.0011	29.4367	29.44
	800	154	135.0	12.34%	0.0010	25.8809	25.88
	900	169	135.0	20.12%	0.0009	25.4289	25.43
Average:	700	130.85	112.3	13.49%	0.0009	21.5616	21.56

Table 6.4: Scheduling algorithm & model output for different minimum distance values when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ and radius = 700

Parameters		Minimum Makespan Values			CPU Times (s)		
# of STAs	Minimum Distance	Greedy Heur.	Sch. Model	Decrease	Greedy Heur.	Sch. Model	Difference
35	50	91	87.0	4.40 %	0.0009	7.7227	7.72
	60	92	87.0	5.43 %	0.0006	8.5091	8.51
	70	93	87.0	6.45 %	0.0005	8.0553	8.05
	80	108	87.0	19.44 %	0.0005	8.8751	8.87
	90	124	87.0	29.84 %	0.0006	1.4094	1.41
40	50	109	106.0	2.75 %	0.0010	12.7986	12.80
	60	110	106.0	3.64 %	0.0008	10.9005	10.90
	70	121	106.0	12.40 %	0.0007	10.8146	10.81
	80	134	106.0	20.90 %	0.0007	9.8873	9.89
	90	149	106.0	28.86 %	0.0006	11.8745	11.87
45	50	123	122.0	0.81 %	0.0014	24.7432	24.74
	60	128	122.0	4.69 %	0.0011	22.6765	22.68
	70	136	122.0	10.29 %	0.0008	24.9359	24.94
	80	154	122.0	20.78 %	0.0008	16.0973	16.10
	90	174	122.0	29.89 %	0.0006	15.7691	15.77
50	50	147	145.0	1.36 %	0.0014	70.7431	70.74
	60	157	145.0	7.64 %	0.0013	51.6203	51.62
	70	167	145.0	13.17 %	0.0010	39.8970	39.90
	80	179	145.0	18.99 %	0.0009	38.2066	38.21
	90	212	145.0	31.60 %	0.0009	33.4913	33.49
Average:	70	135.4	115.0	13.67 %	0.0009	21.4514	21.45

Table 6.3 demonstrate the computational results when minimum Euclidean distance between stations is chosen to be 30. Supply and demand amounts of each station take randomly values from the interval $[1, 5]$. For that analysis, we

fix the minimum distance value and aim to see the effect of different radius values to the performance of the formulation and the algorithm. We note the minimum makespan values, CPU times and also the percentage of the improvement of the objective value on tableau. Note that the result of the greedy heuristics is used as the initial number of timeslots used in the model since it provides an upper bound value to the model. As expected, when number of stations increase from 35 to 50, CPU time to solve the model increase. The time difference between the algorithm and the model is at most 113.97 seconds and the average difference is 21.56 seconds among 20 instances. In general, the time difference increases when $|S|$ (# of stations) gets larger because the CPU time of the model increase more compared to the algorithm. CPU time of the algorithm is at most 0.0017 seconds. The average improvement from the algorithm to the model is calculated as 13.49%, which indicates that our greedy heuristics provides near to optimal results on average. When the radius increases within the set $\{500, 600, 700, 800, 900\}$, model CPU time does not change much as expected. The reason is the optimal solution does not change since the supply and demand values are same for same $|S|$ value and the interference relation is more stable for the single concurrency case.

Another performance comparison between algorithm and scheduling model is presented in Table 6.4. During this analysis, we choose different minimum distance values between stations from the set $\{50, 60, 70, 80, 90\}$ and use fix radius value of 700 to observe the effect of distance values to the performance of the solution methodologies. The algorithm solves the problem with 13.67% closeness to the optimal solution on average. The average CPU time difference is calculated as 21.45 that is very close to the difference value found in Table 6.3. For the larger number of stations, CPU time of the scheduling model increases more than the algorithm does as we expected. However, changing minimum distance value between stations does not affect the optimal results of the model. This is probably due to the fact that the interference matrix does not change. It might result from the discrepancy between $|S|$ and the radius values. Since radius is very large in each case, even smaller minimum distance values can yield large SIR values, which take advantage of concurrent transmissions. In addition, supply and demand values are kept same for the same $|S|$ values as an another factor.

Table 6.5 and Table 6.6 indicate performance comparison of the matching based formulation and Hopcroft-Karp algorithm for the single concurrency case. In this analysis, we are able to work with large number of stations from the set $\{200, 250, 300, 350\}$ because of less solution times of LP model.

Table 6.5: Hopcroft-Karp algorithm & matching model output for different radius values when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ and minimum distance = 30

Parameters		Minimum Makespan Values		CPU Times (s)		
# of STAs	Circle Radius	HK Algorithm	Matching Model	HK Algorithm	Matching Model	Difference
200	500	505	505	0.0271	0.3058	0.28
	600	505	505	0.0317	0.4215	0.39
	700	505	505	0.0263	0.8377	0.81
	800	505	505	0.0269	0.5995	0.57
	900	505	505	0.0355	0.5062	0.47
250	500	631	631	0.0373	0.5719	0.53
	600	628	628	0.0500	0.7526	0.70
	700	628	628	0.0542	0.8385	0.78
	800	628	628	0.0408	1.0067	0.97
	900	628	628	0.0608	1.0455	0.98
300	500	770	770	0.0732	0.7943	0.72
	600	770	770	0.0945	1.3418	1.25
	700	770	770	0.0678	1.3198	1.25
	800	770	770	0.0842	1.4182	1.33
	900	770	770	0.0841	1.5551	1.47
350	500	880	880	0.0948	1.3948	1.30
	600	880	880	0.0948	1.6713	1.58
	700	880	880	0.1046	2.0268	1.92
	800	880	880	0.1039	1.9552	1.85
	900	880	880	0.1056	1.9784	1.87
Average:	700	695.9	695.9	0.0649	1.1171	1.05

Table 6.6: Hopcroft-Karp algorithm & matching model output for different minimum distance values when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ and radius = 1000

Parameters		Minimum Makespan Values		CPU Times (s)		
# of STAs	Minimum Distance	HK Algorithm	Matching Model	HK Algorithm	Matching Model	Difference
100	50	260	260	0.0070	0.0788	0.07
	60	260	260	0.0066	0.0943	0.09
	70	260	260	0.0061	0.1052	0.10
	80	260	260	0.0071	0.1317	0.12
	90	260	260	0.0065	0.1041	0.10
150	50	379	379	0.0142	0.1650	0.15
	60	379	379	0.0143	0.2429	0.23
	70	379	379	0.0154	0.2558	0.24
	80	379	379	0.0210	0.2598	0.24
	90	379	379	0.0211	0.2528	0.23
200	50	507	507	0.0301	0.3387	0.31
	60	503	503	0.0335	0.4284	0.39
	70	503	503	0.0544	0.4741	0.42
	80	503	503	0.0278	0.5410	0.51
	90	503	503	0.0379	0.5153	0.48
250	50	632	632	0.0786	0.5044	0.43
	60	632	632	0.0506	0.8454	0.79
	70	632	632	0.0574	0.8459	0.79
	80	632	632	0.0876	1.0598	0.97
Average:	68.94	433.789	433.789	0.0304	0.3812	0.35

Table 6.5 demonstrates the results for the changing radius values whilst keeping minimum distance values fixed; whereas Table 6.6 provides the results for the changing minimum distance values for fixed radius values. Since HK algorithm solves the problem optimally, objective values are always the same. As it can be

seen on Table 6.5, increasing radius values do not change the optimal values for most of the instances. However, we realize that for the instance when $|S| = 250$, objective decreases from 631 to 628 when radius goes up from 500 to 600. Since we have larger number of stations compared to the ones in Table 6.3 and 6.4 while using radius values from the same set, the discrepancy between radius and $|S|$ is smaller, which results in the closeness of the station locations. That's why when radius increases, minimum latest completion time can decrease since stations become distant from each other and achieves simultaneous signal transmissions. Average CPU time difference between model and HK algorithm is 1.05 seconds, which is much less than the scheduling based solution methodologies as expected since both methodologies solve the problem optimally. As shown in Table 6.5, we were only able to increase number of stations up to 350 because we exceed the limit for the maximum number of trials on the data generation for the station locations when $|S| > 350$.

In Table 6.6, we present the results for different minimum distance values from the set $\{50, 60, 70, 80, 90\}$ for the fixed radius value 1000. We prefer to use larger radius value for the consistency since minimum distance value set has larger values than 30. Compared to the results in Table 6.5, $|S|$ can be increased up to 250. Expectedly, larger minimum distance values lead to larger maximum number of trials needed for the data generation process. Average CPU time difference is 0.35 seconds which is close to the result seen in Table 6.5. As expected, in the instance of $|S| = 200$, the increase in minimum distance value from 50 to 60 yields slight decrease in the objective from 507 to 503. For the instances of $|S| = 250$, we cannot test the results for the minimum distance value of 90 since the maximum number of trials 10000 is exceeded, which takes more than 3600 seconds for data generation.

To evaluate the performance of the solution methodologies developed to minimize total tardiness, deadline algorithm and the mathematical model results are calculated. As it can be seen in Table 6.7 and Table 6.8, we still use fixed $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ vector, and test effect of various radius values and minimum distance values respectively.

Table 6.7: Deadline algorithm and deadline model output for different radius vales when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$, minimum distance = 30

Parameters		Minimum Total Tardiness Values			CPU Times (s)		
# of STAs	Circle Radius	Deadline Algorithm	Deadline Model	Decrease	Deadline Algorithm	Deadline Model	Difference
3	500	0.0	0.0	0.00	0.0010	0.0060	0.01
	600	11.0	1.0	90.91	0.0002	0.0383	0.04
	700	12.0	0.0	100.00	0.0003	0.0079	0.01
	800	17.0	2.0	88.24	0.0003	0.0405	0.04
	900	16.0	1.0	93.75	0.0002	0.0094	0.01
6	500	10.0	7.0	30.00	0.0013	0.5756	0.57
	600	11.0	8.0	27.27	0.0011	0.1310	0.13
	700	7.0	0.0	100.00	0.0010	0.0098	0.01
	800	16.0	5.0	68.75	0.0009	0.1141	0.11
	900	37.0	5.0	86.49	0.0006	0.1283	0.13
9	500	31.0	22.0	29.03	0.0017	5.6887	5.69
	600	21.0	12.0	47.62	0.0021	1.8274	1.83
	700	46.0	11.0	78.26	0.0017	1.7134	1.71
	800	52.0	0.0	100.00	0.0015	0.0218	0.02
	900	52.0	3.0	94.23	0.0013	0.0539	0.05
12	500	17.0	9.0	47.06	0.0060	2.5299	2.52
	600	14.0	4.0	71.43	0.0063	0.4312	0.42
	700	76.0	12.0	85.53	0.0042	2.4193	2.42
	800	149.0	6.0	95.97	0.0030	0.8424	0.84
	900	91.0	0.0	100.00	0.0029	0.0612	0.06
Average:	700	34.3	5.4	71.73	0.0019	0.8325	0.83

Table 6.8: Deadline algorithm and deadline model output for different minimum distance vales when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$, radius = 700

Parameters		Minimum Total Tardiness Values			CPU Times (s)		
# of STAs	Minimum Distance	Deadline Algorithm	Deadline Model	Decrease	Deadline Algorithm	Deadline Model	Difference
3	50	2.0	2.0	0.00	0.0005	0.0315	0.03
	60	7.0	1.0	85.71	0.0002	0.0155	0.02
	70	7.0	1.0	85.71	0.0005	0.0192	0.02
	80	5.0	1.0	80.00	0.0004	0.0219	0.02
	90	8.0	1.0	87.50	0.0002	0.0151	0.01
6	50	3.0	1.0	66.67	0.0012	0.0335	0.03
	60	11.0	4.0	63.64	0.0013	0.0519	0.05
	70	39.0	8.0	79.49	0.0008	0.1658	0.17
	80	42.0	3.0	92.86	0.0008	0.0728	0.07
	90	37.0	3.0	91.89	0.0007	0.0554	0.05
9	50	27.0	12.0	59.26	0.0017	1.3994	1.40
	60	34.0	14.0	61.76	0.0020	2.4977	2.50
	70	13.0	3.0	76.92	0.0017	0.0481	0.05
	80	47.0	11.0	78.72	0.0011	0.9952	0.99
	90	76.0	15.0	80.26	0.0012	4.0879	4.09
12	50	35.0	18.0	48.57	0.0069	7.6754	7.67
	60	50.0	31.0	38.00	0.0052	159.9388	159.93
	70	56.0	2.0	96.43	0.0040	0.1496	0.15
	80	89.0	2.0	97.75	0.0030	0.1501	0.15
	90	122.0	4.0	96.72	0.0024	0.2264	0.22
Average:	70	35.5	6.9	73.39	0.0018	8.8826	8.88

For the minimum tardiness objective, the number of stations is chosen at most 12 since CPU time of the model reaches one hour when $|S| > 12$. According to the results in Table 6.7, CPU time difference on average is 0.83 when minimum distance value is fixed at 30. For most of the instances, increasing radius resulted in larger improvement percentage in the objective value between algorithm and model results. When the radius is 700, we observe 100% improvement to the optimal solution for both $|S| = 3$ and $|S| = 6$. The result of the algorithm is closest to the optimal value on the instance of $|S| = 6$ when radius is 700. We notice the most obvious pattern on the instance of $|S| = 8$ because when radius increases from 500 to 700, the minimum tardiness amount decreases gradually. This occurs likely because of the concurrency allowance when stations are positioned on a wider circle.

In Table 6.8; circle radius value is fixed at 700, and effect of different minimum distance values to the performance is compared. Average CPU time of the mathematical model and the algorithm makes 8.88 in the difference. The discrepancy could result from the instance of $|S| = 12$ while minimum distance is 60 since calculated CPU time is 159.93 seconds that is far above the average duration. The algorithm results both increase and decrease when there is an increase in the minimum distance values, which is also the case for the mathematical model. This might be because of the various deadline values that are defined randomly before solving process.

After evaluating the effects of certain parameter values on the performances in computations, we consider comparing the performances model formulations that aims to minimize latest completion time of transmissions under the single concurrency assumption. In that part of the analysis, we assess the scheduling model, matching model and the flow model by using the same values of the parameters.

Fixing radius value as 500, minimum distance value as 30 and component vector $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$ in Table 6.9, we analyze the number of iterations and CPU times to solve the models for 20 different $|S|$ values. Number of stations are increased within the interval $[30, 50]$. In Table 6.10, we increase the minimum distance value as 60 and keep other parameters as the same as in Table 6.9. Optimal minimum makespan values that are the same for each of the three models are also indicated on tables.

Table 6.9: Model performances under single concurrency when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$, radius = 500, minimum distance = 30

		Scheduling Model		Matching Model		IP Flow Model	
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	# of Iterations	CPU Time (s)	# of Iterations	CPU Time (s)
30	73.00	1199	1.4912	116	0.0057	181	0.0104
31	78.00	1156	1.2259	98	0.0065	149	0.0105
32	82.00	19421	8.3209	88	0.0059	141	0.0104
33	86.00	11258	3.5240	108	0.0088	136	0.0122
34	88.00	970	1.6678	129	0.0097	155	0.0131
35	90.00	3615	3.1683	117	0.0126	186	0.0132
36	85.00	54782	55.2884	124	0.0216	175	0.0114
37	99.00	2596	4.7026	134	0.0123	188	0.0144
38	106.00	1400	3.4047	131	0.0125	136	0.0156
39	98.00	91145	81.5410	147	0.0123	216	0.0157
40	107.00	4031	6.0774	162	0.0133	287	0.0185
41	117.00	104619	78.9174	152	0.0147	187	0.0138
42	105.00	4818	9.8425	153	0.0141	277	0.0190
43	114.00	2974	4.5262	140	0.0127	290	0.0183
44	110.00	5422	7.2966	156	0.0230	260	0.0214
45	123.00	7434	11.6078	199	0.0248	280	0.0332
46	119.00	3576	7.5115	179	0.0227	314	0.0267
47	120.00	2272	7.3789	201	0.0251	239	0.0299
48	122.00	2841	5.7435	225	0.0190	252	0.0269
49	118.00	4262	8.1415	195	0.0226	313	0.0273
50	133.00	7623	12.5616	238	0.0221	280	0.0271

Table 6.10: Model performances under single concurrency when $(\delta, \beta, \Omega) = (2.3, 2.5, 5)$, radius = 500, minimum distance = 60

		Scheduling Model		Matching Model		IP Flow Model	
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	# of Iterations	CPU Time (s)	# of Iterations	CPU Time (s)
30	73.00	1348	1.5621	123	0.0067	168	0.0094
31	76.00	3015	2.8416	116	0.0074	164	0.0090
32	81.00	23338	9.8779	109	0.0069	170	0.0103
33	82.00	2638	4.3932	133	0.0095	154	0.0124
34	94.00	25839	12.0519	111	0.0102	184	0.0097
35	93.00	2197	2.1182	98	0.0104	208	0.0176
36	96.00	2359	2.8452	137	0.0117	164	0.0109
37	104.00	3836	5.4232	163	0.0156	173	0.0137
38	92.00	3570	5.1841	141	0.0157	186	0.0195
39	96.00	3790	4.6208	124	0.0134	168	0.0160
40	109.00	1642	3.9848	148	0.0155	213	0.0161
41	117.00	48336	38.5941	144	0.0142	211	0.0177
42	108.00	49953	35.8221	142	0.0128	319	0.0186
43	117.00	6527	6.5922	165	0.0172	274	0.0227
44	106.00	7493	7.1473	195	0.0164	266	0.0249
45	131.00	1755	5.0021	203	0.0216	272	0.0244
46	132.00	76262	75.4352	160	0.0166	269	0.0228
47	117.00	495	2.7675	217	0.0140	271	0.0231
48	124.00	168937	135.3190	228	0.0176	295	0.0259
49	129.00	69267	80.3895	246	0.0229	316	0.0251
50	133.00	87074	107.4698	274	0.0270	320	0.0244

When we analyze the results in Table 6.9 and 6.10, CPU time of the scheduling model is much larger than the matching and flow models. When minimum distance is 30, the highest CPU time calculated as 55.2884 seconds in scheduling model whereas 135.3190 seconds is reached when minimum distance is 60. On the other hand, maximum duration to solve other formulations in these instances is approximately 0.02 seconds. The reason why scheduling model is outperformed by the other formulations can be due to larger number of constraints and three decision variables used in the formulation.

In addition to CPU time, number of iterations in solving of the formulations also demonstrate considerable difference. Matching model yields the least CPU time compared to the other models as expected because this is linear programming model. However, flow model yields very close results to the matching model. Even though the flow model structure is not totally unimodular and include three indexed decision variable, the number of constraints in the flow model is nearly the same as the matching model. Although for several instances ($|S| = 34, |S| = 37, |S| = 50$ when minimum distance value is 60; $|S| = 36, |S| = 41, |S| = 44$ when minimum distance value is 30) CPU time of the flow model is less than the matching model, matching model solves the minimum latest completion time problem slightly faster than the IP flow model on average.

After we analyze the performance of the formulations under single concurrency assumption, we present the computational results for the multiple concurrency allowance case with minimum latest completion time objective. For the remaining tables presented in this section, we fix the maximum concurrency allowance as 5 which denotes the cardinality of Ω array. Determining each concurrency bound from the interval $[5, 10]$, we specify evenly spaced SIR bound values such that $\Omega = [5.\ \ 6.25\ \ 7.5\ \ 8.75\ \ 10.]$ array is used in each instance. $(\delta, \beta) = (2.3, 2.5)$ and radius $= 1000$ values are also fixed.

In this analysis, we test the performance of the formulations for minimum distance values of 30 and 60; presented in Table 6.11 and Table 6.12 respectively. In this analysis, we compare the LP relaxation of the flow model and LP flow model including the added valid inequality as indicated in Section 4.3.2. Using the minimum distance value as 30 in Table 6.13, and 60 in Table 6.14, we finally demonstrate the results of the totally unimodular LP flow model and LP flow model including the optimality cut as shown in Section 4.3.2.

Table 6.11: Linear flow model and linear model with valid inequality performances under multiple concurrency when maximum concurrency is 5; $(\delta, \beta) = (2.3, 2.5)$; $\Omega = [5.\ \ 6.25\ \ 7.5\ \ 8.75\ \ 10.]$; radius = 1000; minimum distance = 30

	LP Flow Model			LP Flow Model with Valid Inequality		
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	Minimum Makespan	# of Iterations	CPU Time (s)
200	288.00	1050	0.3786	288.00	1203	0.4750
201	289.60	1063	0.1742	289.95	1176	0.5299
202	322.80	1006	0.8253	322.80	1069	0.3755
203	315.60	1066	0.1773	315.60	1083	0.3377
204	305.52	1067	0.1897	305.57	1120	0.3534
205	326.60	1164	0.1914	326.70	1274	0.3589
206	313.00	976	0.1889	313.00	1126	0.3447
207	330.37	1144	0.1871	330.37	1254	0.6371
208	342.20	1047	0.3755	342.20	1146	0.4116
209	327.55	1086	0.1926	327.65	1255	0.3968
210	299.02	1052	0.2058	299.07	1242	0.4271
211	300.95	1106	0.4166	300.95	1204	0.4587
212	327.00	1030	0.2437	327.00	1138	0.4628
213	313.60	1215	0.2491	313.60	1295	0.6245
214	351.90	1146	0.2729	351.90	1030	0.4059
215	344.80	1080	0.1990	344.80	1163	0.3717
216	330.30	1156	0.2581	330.30	1341	0.4234
217	350.50	1109	0.2459	350.50	1273	0.4013
218	321.35	1144	0.2063	321.35	1215	0.3897
219	347.40	1215	0.2065	347.40	1362	0.4538
220	343.40	1115	0.2185	343.40	1255	0.3862

Table 6.12: Linear flow model and linear model with valid inequality under multiple concurrency when maximum concurrency is 5; $(\delta, \beta) = (2.3, 2.5)$; $\Omega = [5.\ \ 6.25\ \ 7.5\ \ 8.75\ \ 10.]$; radius = 1000; minimum distance = 60

	LP Flow Model			LP Flow Model with Valid Inequality		
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	Minimum Makespan	# of Iterations	CPU Time (s)
200	337.60	903	0.1503	337.60	1023	0.3783
201	306.05	1078	0.1696	306.25	1231	0.3556
202	291.00	970	0.1647	291.05	1073	0.3971
203	314.23	1091	0.3298	314.38	1175	0.3481
204	306.20	1100	0.2095	306.30	1204	0.3461
205	339.20	966	0.1949	339.20	1043	0.3433
206	320.00	1157	0.2477	320.15	1184	0.4383
207	329.40	998	0.2403	329.40	1102	0.4068
208	326.60	1062	0.2575	326.60	1153	0.5000
209	312.40	1027	0.2439	312.40	1125	0.4664
210	348.20	1171	0.2234	348.40	1258	0.4760
211	294.20	1036	0.1862	294.25	1131	0.4013
212	356.90	1119	0.2044	357.00	1293	0.4329
213	327.05	1165	0.3154	327.05	1235	0.4395
214	307.65	1106	0.2085	307.70	1258	0.4635
215	306.47	1099	0.1994	306.77	1252	0.3863
216	304.40	1040	0.2297	304.60	1230	0.5186
217	324.40	1073	0.4763	324.40	1133	0.4314
218	325.00	1067	0.2165	325.00	1202	0.4210
219	338.85	1169	0.3091	338.85	1337	0.3984
220	324.80	1126	0.2115	324.80	1185	0.5086

Table 6.13: Linear totally unimodular flow model and linear flow model with optimality cut performances when maximum concurrency is 5; $(\delta, \beta) = (2.3, 2.5)$; $\Omega = [5.\ \ 6.25\ \ 7.5\ \ 8.75\ \ 10.]$; radius = 1000; minimum distance = 30

	Totally Unimodular LP Flow Model			LP Flow Model with Optimality Cut		
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	Minimum Makespan	# of Iterations	CPU Time (s)
200	279.60	990	0.1721	279.60	1011	0.0996
201	291.15	983	0.1522	291.15	1024	0.0822
202	317.40	1088	0.1636	317.40	1047	0.0787
203	290.32	1082	0.1646	290.32	1115	0.0856
204	293.80	1035	0.1770	293.80	1059	0.1009
205	329.38	1088	0.4668	329.38	1063	0.0960
206	355.00	1141	0.1721	355.00	1139	0.1030
207	319.07	1115	0.1692	319.07	1099	0.0947
208	316.00	1065	0.1776	316.00	1162	0.1015
209	349.20	978	0.1708	349.20	927	0.0932
210	324.60	1040	0.1715	324.60	991	0.0902
211	322.20	1106	0.1901	322.20	1142	0.1051
212	307.80	1008	0.1847	307.80	1023	0.1045
213	388.65	1112	0.1960	388.65	1278	0.1115
214	350.43	1221	0.1979	350.43	1151	0.1047
215	355.00	1160	0.1899	355.00	1217	0.1123
216	314.60	1127	0.2003	314.60	1106	0.0998
217	315.18	1121	0.1905	315.18	1148	0.1027
218	312.25	1060	0.1799	312.25	1151	0.1041
219	318.85	1111	0.1998	318.85	1107	0.1057
220	349.60	1121	0.2017	349.60	1060	0.1013

Table 6.14: Linear totally unimodular flow model and linear flow model with optimality cut performances when maximum concurrency is 5; $(\delta, \beta) = (2.3, 2.5)$; $\Omega = [5.\ \ 6.25\ \ 7.5\ \ 8.75\ \ 10.]$; radius = 1000; minimum distance = 60

	Totally Unimodular LP Flow Model			LP Flow Model with Optimality Cut		
# of STAs	Minimum Makespan	# of Iterations	CPU Time (s)	Minimum Makespan	# of Iterations	CPU Time (s)
200	287.87	1100	0.1725	287.87	1058	0.3187
201	296.60	1057	0.1725	296.60	1020	0.1053
202	294.00	973	0.1655	294.00	973	0.0780
203	342.60	1116	0.3534	342.60	1134	0.1051
204	294.80	1052	0.1771	294.80	1034	0.0919
205	293.15	1106	0.2358	293.15	1106	0.1569
206	317.00	1042	0.1834	317.00	988	0.0928
207	307.00	1049	0.1912	307.00	1022	0.1445
208	315.60	1047	0.1718	315.60	1034	0.0930
209	334.67	1121	0.1801	334.67	1090	0.1024
210	312.77	1110	0.1922	312.77	1075	0.0971
211	328.40	971	0.1711	328.40	985	0.0980
212	343.60	1043	0.1781	343.60	1008	0.0915
213	324.53	1089	0.3246	324.53	1126	0.1015
214	312.00	1098	0.1794	312.00	1121	0.1033
215	349.60	1019	0.1787	349.60	1057	0.1058
216	319.05	1075	0.1802	319.05	1078	0.1028
217	345.18	1050	0.1963	345.18	1113	0.1127
218	324.20	1116	0.1962	324.20	1066	0.0988
219	313.58	1069	0.1983	313.58	1114	0.1152
220	336.80	1091	0.2331	336.80	1119	0.1031

As it can be seen in Tables 6.11 - 6.14, number of stations can be above 200 when we solve LP formulations of the flow model. The maximum CPU time observed among four of the formulations is 0.8253 seconds. As we expected, minimum latest completion time values increase when number of stations increase since the total supply and demand values of stations also gets larger proportionately. When we analyze the results in terms of number of iterations applied in the solving process, we realize that even the maximum number among the four tables is notably smaller than the average results of the instances we obtain from the scheduling model with $|S| \in [30, 50]$ seen in Tables 6.9 - 6.10.

Most of the results from the instances noted in Table 6.11 demonstrate that LP flow model solves the problem slightly quicker than LP flow model with valid inequality, when the minimum distance between stations is 30. We observe the reverse only on the instance with $|S| = 202$ as the valid inequality model solves 0.3755 seconds whereas the other does 0.8253 seconds. Expectedly, number of iterations in the valid inequality model is mostly larger since it includes larger number of constraints in total. Additionally, we can infer from Tables 6.9 - 6.10 that the feasible region of the LP flow model does not narrow substantially when we add the valid inequality because the CPU time and number of iterations differences between two models do not turn out to be large as the objective functions are the same. For the minimum distance of 60 in Table 6.10, CPU times and number of iterations do not differ much compared to results from the minimum distance of 30; but there is a slight improvement in terms of the objective value. This might be due to the fact that the increased distance between stations enables more opportunity for concurrent transmissions with less interference issue which reduces the minimum makespan value.

According to the results in Tables 6.13 - 6.14, added optimality cut to the LP flow model decreases CPU time, which was expected since the model chooses either $x_{kl}^{p_{kl}}$ value to be positive or sets all x_{kl}^{i} values to 0 for all $1 \leq i \leq p_{kl}$. It is not surprising to see the optimal results of the totally unimodular and optimality cut model to be the same since it was proved in Theorem 4.3.8. We also realize the slight improvement in objective function value through the increased simultaneous transmissions with less interference issue when minimum distance between stations increases. However, since there is not much difference between optimal results in Table 6.13 and 6.14, we can predict this behavior being dependent on the radius value. Since radius is increased to 1000 in this analysis, stations can be distributed uniformly with less density so that considerable interference is not inspected even though their minimum distance among each other can be less in Table 6.13.

6.3 Sensitivity Analysis

In order to measure the detailed effects of the parameter values and provide managerial insights into the minimization of latest completion time problem, we conduct sensitivity analysis in this section. We analyze the effects of R (circle radius), d (minimum distance value between stations), δ (path loss exponent), β (interference path loss exponent) on the optimal objective value, number of iterations to solve the formulations, and solution time in terms of CPU seconds.

For this analysis, we consider multiple concurrency assumption when maximum concurrency allowance is 5 ($|\Omega| = 5$), and fix $\Omega = [2.\ 4.\ 6.\ 8.\ 10.]$. Since we provide flow based formulations as our solution methodology for the multiple concurrency assumption; we present the results of IP flow model, LP relaxation of IP flow model and LP flow model including valid inequality. Those results can be observed in Tables 6.15 - 6.19 where we update number of stations ($|S|$) from the set $\{100, 200, 300, 400, 500\}$ respectively. The remaining columns starting from 5th column in Tables 6.15 - 6.19 stand for the followings:

- $z_I^* \rightarrow$ *optimal objective value of IP flow model*
- $s_I \rightarrow$ *number of iterations to solve IP flow model*
- $t_I \rightarrow$ *CPU time in seconds to solve IP flow model*
- $z_L^* \rightarrow$ *optimal objective value of LP flow model*
- $s_L \rightarrow$ *number of iterations to solve LP flow model*
- $t_L \rightarrow$ *CPU time in seconds to solve LP flow model*

- $z_V^* \rightarrow$ *optimal objective value of LP flow model with valid inequality*
- $s_V \rightarrow$ *number of iterations to solve LP flow model with valid inequality*
- $t_V \rightarrow$ *CPU time in seconds to solve LP flow model with valid inequality*

In these tables, it can be observed that CPU time of IP flow model is usually longer than LP flow formulations. IP flow model has the maximum duration of 14.4392 seconds in CPU time for the instance of $|S| = 500$ when $R = 1200$, $d = 60$ and the component vector $(\delta, \beta) = (3.0, 1.5)$ are chosen. LP flow has less CPU time than the valid inequality model that is reasonable since valid inequality model has more number of constraints. Expectedly, longer CPU times appear in Table 6.19 where number of stations is maximum as 500. In addition, optimal objective values are higher in large instances since total supply and demand values increase in these instances. The difference between optimal objective values of the formulations is more evident when $|S|$ gets larger and LP flow model always gives the maximum value. Valid inequality model is able to cut certain fractional solutions when we observe the decrease in the optimal objective values from LP flow model.

When we analyze the effects of the path loss and interference path loss components, we realize that for the same δ values, larger β values lead better objective values. When we check the SIR value calculation defined in Section 3.2, these results verify that larger β values result in increase in number of concurrent transmissions. For some instances, when β increases for the same δ value, the objective might not improve such as for $|S| = 100, R = 1200, d = 305$; the objective remains 134.00 after $\beta = 2.25$, shown in Table 6.15. As our interpretation, we can claim that the ratio of SIR values might exceed the last element of Ω arrray in that case, and simultaneous transmissions are fully utilized to achieve minimal makespan from that point. We also verify that larger δ value worsens the objective. That is; when we increase δ from 7.5 to 18.75, even if the β values increase, the increase in the objective is observed less compared to amount of increase in objective that changing δ from 3.0 to 7.5 results in. Since ratio of δ value increment is higher than ratio of β increment, objective improves less as we expected. CPU times demonstrate very similar results, but mostly solving time takes slightly longer when vector component values increase.

Minimum distance between stations gives better optimal results when $d = 60$ compared to $d = 30$. In Tables 6.15 - 6.19, for the same component vector (δ, β), increasing the minimum distance leads to increase in the objective values of all the formulations. That is why we can verify that interference issue is less observed when stations are more distant from each other, so that they can take advantage of the simultaneous transmissions. Despite the closeness in CPU times, there has been slight increase in the duration when minimum distance is increased from 30 to 60.

Circle radius R is increased from 1100 to 1300 with increments of 100, for each $|S|$ values indicated in Tables 6.15 - 6.19. As expected, larger radius values lead to improvement in the objective value since stations can be distributed uniformly with more space that sets them apart from each other. The impact of increase in R on the objective is relatively lower when the minimum distance between stations is 60 than 30, because stations are able to utilize the advantage of increase in network area when the minimum required distance between them is smaller. For all of the flow formulations, CPU times increase as the radius gets larger. It may depend on the increased possibilities for the stations to use simultaneous transmission chance when the total area they are uniformly distributed gets larger.

Number of iterations to solve the models is larger when R, d and (δ, β) values increase for most of the instances. However, when $|S| \in \{200, 300, 400\}$, the number of iterations is maximum for the minimum radius value of 1100. These instances can be observed in Tables 6.15 - 6.17. We cannot conclude a direct relation between $|S|$ value and number of iterations since when $|S|$ increases; for the fixed values for the other parameters, number of iterations can both increase and decrease depending on the instance. This can be due to fluctuating changes in total supply and demand values. As we expected, IP flow model has more number of iterations than other formulations, and LP flow model has less number of iterations than valid inequality model. Since same difference is inspected among CPU times of the models, iteration results are also indicator of performances.

Table 6.15: Sensitivity analysis on flow models with $|S| = 100, \Omega = [2.\ 4.\ 6.\ 8.\ 10.]$

R	d	δ	β	z_I^*	s_I	t_I	z_L^*	s_L	t_L	z_V^*	s_V	t_V
1100	30	3.000	1.500	0.00	0	0.0006	0.00	0	0.0006	0.00	0	0.0006
			2.250	78.00	579	0.0962	79.93	6	0.0015	79.93	198	0.0027
		7.500	3.375	78.00	579	0.0496	79.93	6	0.0017	79.93	198	0.0025
			5.062	78.00	579	0.0501	79.93	6	0.0014	79.93	198	0.0026
		18.750	7.594	78.00	579	0.0667	79.93	6	0.0021	79.93	198	0.0027
			11.391	78.00	579	0.0472	79.93	6	0.0015	79.93	198	0.0025
	60	3.000	1.500	78.00	579	0.0504	79.93	6	0.0015	79.93	198	0.0025
			2.250	105.00	726	0.0571	107.43	13	0.0014	107.43	256	0.0027
		7.500	3.375	105.00	726	0.0553	107.43	13	0.0014	107.43	256	0.0029
			5.062	106.00	647	0.0487	107.97	10	0.0014	107.97	248	0.0028
		18.750	7.594	106.00	647	0.0632	107.97	10	0.0014	107.97	248	0.0030
			11.391	116.00	906	0.0701	118.77	12	0.0018	118.77	277	0.0034
1200	30	3.000	1.500	116.00	906	0.0709	118.77	12	0.0018	118.77	277	0.0034
			2.250	134.00	1029	0.0895	137.57	17	0.0017	137.57	332	0.0041
		7.500	3.375	134.00	1029	0.0895	137.57	17	0.0019	137.57	332	0.0043
			5.062	134.00	1029	0.0983	137.57	17	0.0017	137.57	332	0.0042
		18.750	7.594	134.00	1029	0.1119	137.57	17	0.0022	137.57	332	0.0054
			11.391	134.00	1029	0.1534	137.57	17	0.0019	137.57	332	0.0051
	60	3.000	1.500	134.00	1029	0.0995	137.57	17	0.0019	137.57	332	0.0050
			2.250	208.00	1195	0.1053	212.07	70	0.0023	212.07	390	0.0049
		7.500	3.375	208.00	1195	0.0942	212.07	70	0.0024	212.07	390	0.0050
			5.062	208.00	1195	0.0961	212.07	70	0.0022	212.07	390	0.0051
		18.750	7.594	208.00	1195	0.0925	212.07	70	0.0022	212.07	390	0.0051
			11.391	208.00	1195	0.0944	212.07	70	0.0024	212.07	390	0.0052
1300	30	3.000	1.500	208.00	1195	0.0970	212.07	70	0.0022	212.07	390	0.0051
			2.250	220.00	1639	0.1303	224.67	29	0.0027	224.67	607	0.0070
		7.500	3.375	220.00	1639	0.1561	224.67	29	0.0022	224.67	607	0.0068
			5.062	242.00	1529	0.1477	247.17	32	0.0023	247.17	573	0.0071
		18.750	7.594	242.00	1529	0.1575	247.17	32	0.0024	247.17	573	0.0076
			11.391	242.00	1529	0.1387	247.17	32	0.0028	247.17	573	0.0069
	60	3.000	1.500	242.00	1529	0.1311	247.17	32	0.0021	247.17	573	0.0068
			2.250	261.00	1477	0.1371	266.17	35	0.0021	266.17	571	0.0074
		7.500	3.375	261.00	1477	0.1317	266.17	35	0.0021	266.17	571	0.0066
			5.062	264.00	1764	0.1353	269.62	36	0.0027	269.62	564	0.0068
		18.750	7.594	264.00	1764	0.1615	269.62	36	0.0026	269.62	564	0.0069
			11.391	264.00	1764	0.1495	269.62	36	0.0023	269.62	564	0.0072

Table 6.16: Sensitivity analysis on flow models with $|S| = 200, \Omega = [2.\ 4.\ 6.\ 8.\ 10.]$

R	d	δ	β	z_I^*	s_I	t_I	z_L^*	s_L	t_L	z_V^*	s_V	t_V
1100	30	3.000	1.500	0.00	0	0.0006	0.00	0	0.0005	0.00	0	0.0006
			2.250	101.00	510	0.0195	101.67	10	0.0016	101.67	450	0.0038
		7.500	3.375	101.00	510	0.0334	101.67	10	0.0016	101.67	450	0.0045
			5.062	192.00	1765	0.0844	194.37	16	0.0029	194.37	784	0.0136
		18.750	7.594	192.00	1765	0.0866	194.37	16	0.0030	194.37	784	0.0149
			11.391	224.00	2071	0.1002	226.77	18	0.0054	226.77	1315	0.0200
	60	3.000	1.500	224.00	2071	0.1215	226.77	18	0.0060	226.77	1315	0.0211
			2.250	318.00	2804	0.1747	322.02	27	0.0073	321.93	1630	0.0255
		7.500	3.375	318.00	2804	0.1735	322.02	27	0.0064	321.93	1630	0.0291
			5.062	329.00	3150	0.1727	333.68	28	0.0068	333.60	1771	0.0284
		18.750	7.594	329.00	3150	0.1721	333.68	28	0.0080	333.60	1771	0.0376
			11.391	329.00	3150	0.2023	333.68	28	0.0070	333.60	1771	0.0269
1200	30	3.000	1.500	329.00	3150	0.1693	333.68	28	0.0144	333.60	1771	0.0275
			2.250	380.00	3073	0.1879	385.35	37	0.0075	385.27	1901	0.0299
		7.500	3.375	380.00	3073	0.2109	385.35	37	0.0080	385.27	1901	0.0294
			5.062	427.00	77	0.2955	433.21	44	0.0109	433.12	2287	0.0412
		18.750	7.594	427.00	77	0.2840	433.21	44	0.0106	433.12	2287	0.0409
			11.391	427.00	77	0.2946	433.21	44	0.0132	433.12	2287	0.0426
	60	3.000	1.500	427.00	77	0.3167	433.21	44	0.0111	433.12	2287	0.0447
			2.250	490.00	94	0.3412	498.26	57	0.0113	498.17	2803	0.0534
		7.500	3.375	490.00	94	0.3355	498.26	57	0.0112	498.17	2803	0.0535
			5.062	540.00	119	0.5213	549.41	67	0.0130	549.33	3190	0.0636
		18.750	7.594	540.00	119	0.5219	549.41	67	0.0135	549.33	3190	0.0661
			11.391	572.00	80	0.4290	581.81	50	0.0137	581.73	47	0.0248
1300	30	3.000	1.500	572.00	80	0.4422	581.81	50	0.0151	581.73	47	0.0244
			2.250	606.00	88	0.4096	616.51	50	0.0133	616.42	48	0.0271
		7.500	3.375	606.00	88	0.4274	616.51	50	0.0143	616.42	48	0.0258
			5.062	630.00	95	0.5303	641.41	52	0.0144	641.33	51	0.0286
		18.750	7.594	662.00	101	0.6185	673.81	53	0.0185	673.72	53	0.0270
			11.391	675.00	101	0.6023	686.61	55	0.0189	686.52	54	0.0312
	60	3.000	1.500	675.00	101	0.6157	686.61	55	0.0161	686.52	54	0.0315
			2.250	712.00	105	0.5374	723.17	60	0.0163	723.09	59	0.0299
		7.500	3.375	712.00	105	0.5005	723.17	60	0.0168	723.09	59	0.0316
			5.062	756.00	112	0.5804	768.17	65	0.0189	768.09	64	0.0338
		18.750	7.594	756.00	112	0.5686	768.17	65	0.0187	768.09	64	0.0335
			11.391	783.00	114	0.8502	795.17	76	0.0194	795.09	65	0.0373

Table 6.17: Sensitivity analysis on flow models with $|S| = 300, \Omega = [2.\ 4.\ 6.\ 8.\ 10.]$

R	d	δ	β	z_I^*	s_I	t_I	z_L^*	s_L	t_L	z_V^*	s_V	t_V
1100	30	3.000	1.500	0.00	0	0.0007	0.00	0	0.0006	0.00	0	0.0008
			2.250	110.00	1166	0.0777	112.07	8	0.0042	112.07	1066	0.0117
		7.500	3.375	110.00	1166	0.0703	112.07	8	0.0032	112.07	1066	0.0106
			5.062	201.00	2726	0.1882	205.33	607	0.0192	205.33	1638	0.0275
		18.750	7.594	201.00	2726	0.1879	205.33	607	0.0202	205.33	1638	0.0293
			11.391	222.00	2781	0.2047	226.93	511	0.0171	226.93	1659	0.0329
	60	3.000	1.500	243.00	2743	0.2398	248.53	27	0.0102	248.53	1886	0.0363
			2.250	353.00	45	0.6755	359.82	32	0.0143	359.82	27	0.0239
		7.500	3.375	353.00	45	0.6845	359.82	32	0.0341	359.82	27	0.0261
			5.062	415.00	50	0.6667	422.22	34	0.0149	422.22	34	0.0275
		18.750	7.594	415.00	50	0.6888	422.22	34	0.0148	422.22	34	0.0264
			11.391	420.00	51	0.7785	427.22	35	0.0179	427.22	35	0.0277
1200	30	3.000	1.500	420.00	51	0.7206	427.22	35	0.0247	427.22	35	0.0266
			2.250	458.00	55	0.7169	465.22	39	0.0160	465.22	38	0.0277
		7.500	3.375	458.00	55	0.6885	465.22	39	0.0166	465.22	38	0.0287
			5.062	458.00	55	0.7758	465.22	39	0.0161	465.22	38	0.0279
		18.750	7.594	458.00	55	0.7984	465.22	39	0.0156	465.22	38	0.0282
			11.391	458.00	55	0.6997	465.22	39	0.0151	465.22	38	0.0317
	60	3.000	1.500	458.00	55	0.6832	465.22	39	0.0155	465.22	38	0.0288
			2.250	529.00	62	0.5205	536.73	48	0.0198	536.73	48	0.0436
		7.500	3.375	529.00	62	0.5150	536.73	48	0.0208	536.73	48	0.0370
			5.062	582.00	69	0.5620	590.23	52	0.0209	590.23	52	0.0414
		18.750	7.594	582.00	69	0.5551	590.23	52	0.0204	590.23	52	0.0369
			11.391	607.00	71	0.7384	616.34	55	0.0234	616.34	53	0.0423
1300	30	3.000	1.500	607.00	71	0.7162	616.34	55	0.0228	616.34	53	0.0432
			2.250	683.00	76	0.8600	692.14	63	0.0300	692.14	58	0.0530
		7.500	3.375	683.00	76	0.8254	692.14	63	0.0311	692.14	58	0.0490
			5.062	753.00	83	0.8034	763.41	71	0.0265	763.41	70	0.0547
		18.750	7.594	753.00	83	0.8782	763.41	71	0.0316	763.41	70	0.0533
			11.391	788.00	97	0.9943	799.71	73	0.0293	799.71	73	0.0547
	60	3.000	1.500	788.00	97	0.9601	799.71	73	0.0297	799.71	73	0.0677
			2.250	871.00	110	1.3737	883.51	83	0.0344	883.51	80	0.0593
		7.500	3.375	871.00	110	1.1667	883.51	83	0.0356	883.51	80	0.0607
			5.062	913.00	132	1.2219	926.72	86	0.0336	926.72	83	0.0694
		18.750	7.594	913.00	132	1.2507	926.72	86	0.0356	926.72	83	0.0676
			11.391	914.00	129	1.2497	927.42	86	0.0335	927.42	83	0.0664

Table 6.18: Sensitivity analysis on flow models with $|S| = 400, \Omega = [2.\ 4.\ 6.\ 8.\ 10.]$

R	d	δ	β	z_I^*	s_I	t_I	z_L^*	s_L	t_L	z_V^*	s_V	t_V
1100	30	3.000	1.500	0.00	0	0.0008	0.00	0	0.0006	0.00	0	0.0008
			2.250	164.00	2244	0.3013	165.40	25	0.0065	165.40	1179	0.0202
		7.500	3.375	164.00	2244	0.2394	165.40	25	0.0069	165.40	1179	0.0212
			5.062	278.00	2737	0.9802	279.80	40	0.0201	279.80	1169	0.0410
		18.750	7.594	278.00	2737	1.0588	279.80	40	0.0143	279.80	1169	0.0575
			11.391	292.00	32	1.6213	294.20	29	0.0198	294.20	30	0.0270
	60	3.000	1.500	292.00	32	1.2151	294.20	29	0.0359	294.20	30	0.0272
			2.250	461.00	49	3.2797	464.81	44	0.0275	464.81	42	0.0443
		7.500	3.375	461.00	49	2.8402	464.81	44	0.0216	464.81	42	0.0385
			5.062	512.00	55	2.4272	517.11	49	0.0248	517.11	48	0.0453
		18.750	7.594	537.00	57	2.4230	542.31	50	0.0260	542.31	50	0.0480
			11.391	581.00	62	3.1276	587.38	54	0.0303	587.38	53	0.0537
1200	30	3.000	1.500	588.00	64	2.5981	594.88	55	0.0300	594.88	53	0.0528
			2.250	695.00	77	2.6797	703.08	66	0.0331	703.08	65	0.0658
		7.500	3.375	729.00	80	3.0053	737.28	69	0.0436	737.28	67	0.0885
			5.062	787.00	87	3.5579	795.78	75	0.0554	795.78	73	0.0990
		18.750	7.594	787.00	87	3.2344	795.78	75	0.0455	795.78	73	0.0954
			11.391	817.00	89	3.1530	825.78	78	0.0480	825.78	76	0.0890
	60	3.000	1.500	817.00	89	3.0008	825.78	78	0.0440	825.78	76	0.0875
			2.250	927.00	99	3.2457	937.62	86	0.0554	937.58	85	0.0982
		7.500	3.375	927.00	99	3.2440	937.62	86	0.0508	937.58	85	0.1020
			5.062	982.00	104	3.5226	993.42	93	0.0530	993.38	92	0.1045
		18.750	7.594	982.00	104	3.4244	993.42	93	0.0541	993.38	92	0.1072
			11.391	995.00	105	3.3330	1006.62	98	0.0597	1006.58	95	0.1097
1300	30	3.000	1.500	995.00	105	3.3011	1006.62	98	0.0577	1006.58	95	0.1045
			2.250	1043.00	106	3.1946	1054.62	103	0.0539	1054.58	98	0.1053
		7.500	3.375	1043.00	106	3.1840	1054.62	103	0.0617	1054.58	98	0.1033
			5.062	1152.00	140	4.3744	1166.22	120	0.0673	1166.18	113	0.1258
		18.750	7.594	1152.00	140	4.3432	1166.22	120	0.0700	1166.18	113	0.1479
			11.391	1175.00	136	4.7937	1189.62	121	0.1143	1189.58	114	0.1304
	60	3.000	1.500	1175.00	136	4.5027	1189.62	121	0.0657	1189.58	114	0.1322
			2.250	1300.00	164	3.2703	1317.12	139	0.0657	1317.08	133	0.1262
		7.500	3.375	1300.00	164	3.1292	1317.12	139	0.0637	1317.08	133	0.1215
			5.062	1368.00	183	3.5644	1387.32	146	0.0831	1387.28	140	0.1512
		18.750	7.594	1391.00	191	3.4829	1410.72	149	0.0767	1410.68	145	0.1386
			11.391	1418.00	200	3.6451	1437.72	151	0.0735	1437.68	146	0.1438

Table 6.19: Sensitivity analysis on flow models with $|S| = 500, \Omega = [2.\ 4.\ 6.\ 8.\ 10.]$

R	d	δ	β	z_I^*	s_I	t_I	z_L^*	s_L	t_L	z_V^*	s_V	t_V
1100	30	3.000	1.500	0.00	0	0.0008	0.00	0	0.0007	0.00	0	0.0007
			2.250	176.00	38	0.9552	178.93	25	0.0124	178.93	1724	0.0429
		7.500	3.375	176.00	38	0.9501	178.93	25	0.0121	178.93	1724	0.0421
			5.062	279.00	28	1.0663	283.53	24	0.0188	283.53	22	0.0327
		18.750	7.594	292.00	30	0.9734	296.87	25	0.0202	296.87	23	0.0341
			11.391	357.00	36	1.1306	362.87	30	0.0213	362.87	29	0.0380
	60	3.000	1.500	382.00	37	1.1702	387.87	31	0.0214	387.87	31	0.0398
			2.250	552.00	52	1.3983	558.87	41	0.0291	558.87	42	0.0531
		7.500	3.375	552.00	52	1.3898	558.87	41	0.0291	558.87	42	0.0512
			5.062	644.00	59	1.4905	651.52	47	0.0313	651.52	47	0.0617
		18.750	7.594	644.00	59	1.4914	651.52	47	0.0320	651.52	47	0.0604
			11.391	730.00	68	1.7817	737.92	57	0.0369	737.92	53	0.0716
1200	30	3.000	1.500	730.00	68	1.7391	737.92	57	0.0371	737.92	53	0.0695
			2.250	911.00	88	1.9749	919.78	70	0.0445	919.78	66	0.0932
		7.500	3.375	911.00	88	2.0066	919.78	70	0.0456	919.78	66	0.0898
			5.062	912.00	93	12.6370	921.18	72	0.0533	921.18	66	0.0917
		18.750	7.594	914.00	93	13.7083	923.18	73	0.0502	923.18	66	0.0963
			11.391	914.00	93	13.9706	923.18	73	0.0682	923.18	66	0.1091
	60	3.000	1.500	914.00	93	14.4392	923.18	73	0.0678	923.18	66	0.1079
			2.250	1007.00	106	8.4772	1017.38	79	0.0566	1017.38	74	0.1145
		7.500	3.375	1007.00	106	8.4629	1017.38	79	0.0558	1017.38	74	0.1117
			5.062	1120.00	129	8.4709	1132.28	89	0.0635	1132.28	80	0.1233
		18.750	7.594	1120.00	129	8.5921	1132.28	89	0.0625	1132.28	80	0.1214
			11.391	1150.00	131	8.6979	1162.88	91	0.0667	1162.88	81	0.1450
1300	30	3.000	1.500	1167.00	132	9.4377	1179.88	92	0.0738	1179.88	82	0.1393
			2.250	1276.00	149	9.8221	1289.68	102	0.0841	1289.68	93	0.1453
		7.500	3.375	1276.00	149	9.4149	1289.68	102	0.0730	1289.68	93	0.1419
			5.062	1371.00	164	10.2483	1385.28	119	0.0974	1385.28	106	0.1592
		18.750	7.594	1371.00	164	9.7240	1385.28	119	0.0816	1385.28	106	0.1582
			11.391	1401.00	168	9.7963	1415.88	125	0.0895	1415.88	112	0.1575
	60	3.000	1.500	1401.00	168	9.9129	1415.88	125	0.0910	1415.88	112	0.1748
			2.250	1549.00	206	10.8379	1565.02	143	0.0959	1565.02	126	0.1661
		7.500	3.375	1567.00	208	10.4934	1583.02	148	0.1041	1583.02	131	0.1716
			5.062	1660.00	229	10.7196	1677.22	160	0.1030	1677.22	140	0.1814
		18.750	7.594	1660.00	229	10.6213	1677.22	160	0.0995	1677.22	140	0.2010
			11.391	1672.00	233	10.9266	1689.82	161	0.1040	1689.82	141	0.1796

Chapter 7

Conclusion and Future Work

In this study, we examine simultaneous receive and transmit operations in wireless networks to provide an optimization methodology for the node selection process of the access point in unidirectional full duplex transmissions. Assessing the problem under both single and multiple concurrency assumption in terms of number of signal transmissions in single timeslot, we consider minimization of latest completion time of all transmissions and minimization of total tardiness objectives throughout this thesis.

Two opposite directions of signal transmissions are interpreted as supply and demand flow requirements of stations. In our solution methodology, we embrace scheduling and matching based formulations to find minimum latest completion time. From the scheduling perspective, assignment of supply and demands to the timeslots principle is followed where we only consider single concurrency assumption. Scheduling based MILP model is formulated and greedy heuristics that outputs initial latest completion time is developed. Result of the heuristics is then used as an upper bound parameter for the number of timeslots in the model. Under single concurrency assumption, matching based framework is considered where each matched edge between disjoint supply and demand sets represents simultaneous transmission between two non-interfering stations. Based on bipartite matching, LP model is formulated and it gives integer optimal solution due to totally unimodular structure of constraint matrix. Hopcroft-Karp algorithm for bipartite matching in the literature is applied to our problem to achieve minimum makespan and it leads optimal results.

In order to get the multiple concurrency involved in our problem, we developed flow based formulations. IP flow model for multiple concurrency is provided with the objective of the maximization of concurrent transmissions which brings out the minimum makespan. Analyzing the properties of LP relaxation of the flow model, we proved certain patterns for extreme points in the feasible region. As a result, we proved a valid inequality for the flow model and also found an optimality cut. Afterwards, we provided equivalent LP relaxation flow model which has totally unimodular structure finding integer optimal results. Flow based formulation is observed to be applicable to any maximum concurrency size in $n \in \mathbb{Z}^+$.

For the minimum total tardiness objective with single concurrency assumption, we introduce the deadline restriction to complete transmissions for each station in the network. We use scheduling based formulation and prove that minimum tardiness problem with single concurrency assumption is NP-complete. In addition, we propose an algorithm for this objective which is based on priority criteria that sorts the stations from first to last to be served by the access point.

In our computational study, we provide detailed comparison between all the algorithms and formulations in terms of their CPU times required for solution and also the objective values they yielded. Experiments indicate that flow based formulation outperforms other formulations in terms of its multiple concurrency extension and smaller CPU times. In addition, we conduct sensitivity analysis on flow based formulations when maximum concurrency allowance is 5. These analyses enable us to understand the effect of each parameter on the resulting objective values and CPU times of solution methodologies.

Future research of this study could be the analysis for the performance of the formulations when the data points for the station locations are not uniformly distributed different than our assumption in this study. In addition, more efficient algorithms for minimum tardiness objective that can yield better results within shorter computation times could be developed for this problem.

Bibliography

[1] X. Xia, K. Xu, Y. Wang, and Y. Xu, "A 5g-enabling technology: benefits, feasibility, and limitations of in-band full-duplex mmimo," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 81–90, 2018.

[2] B. Yin, M. Wu, C. Studer, J. R. Cavallaro, and J. Lilleberg, "Full-duplex in large-scale wireless systems," in *2013 Asilomar Conference on Signals, Systems and Computers*, pp. 1623–1627, IEEE, 2013.

[3] A. Tang and X. Wang, "A-duplex: Medium access control for efficient coexistence between full-duplex and half-duplex communications," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5871–5885, 2015.

[4] J. Lee and T. Q. Quek, "Hybrid full-/half-duplex system analysis in heterogeneous wireless networks," *IEEE transactions on wireless communications*, vol. 14, no. 5, pp. 2883–2895, 2015.

[5] Z. Tong and M. Haenggi, "Throughput analysis for full-duplex wireless networks with imperfect self-interference cancellation," *IEEE Transactions on communications*, vol. 63, no. 11, pp. 4490–4500, 2015.

[6] Z. Zhang, X. Chai, K. Long, A. V. Vasilakos, and L. Hanzo, "Full duplex techniques for 5g networks: self-interference cancellation, protocol design, and relay selection," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 128–137, 2015.

[7] W. Ye and J. Heidemann, "Medium access control in wireless sensor networks," *Wireless sensor networks*, pp. 73–91, 2004.

[8] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "Ieee 802.11 ax: Challenges and requirements for future high efficiency wifi," *IEEE wireless communications*, vol. 24, no. 3, pp. 130–137, 2016.

[9] S. Goyal, P. Liu, O. Gurbuz, E. Erkip, and S. Panwar, "A distributed mac protocol for full duplex radio," in *2013 Asilomar conference on signals, systems and computers*, pp. 788–792, IEEE, 2013.

[10] A. Aijaz and P. Kulkarni, "Simultaneous transmit and receive operation in next generation ieee 802.11 wlans: a mac protocol design approach," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 128–135, 2017.

[11] M. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 5, pp. 885–897, 1983.

[12] I. Chlamtac and S. Kutten, "On broadcasting in radio networks-problem analysis and protocol design," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240–1246, 1985.

[13] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on communications*, vol. 38, no. 4, pp. 456–460, 1990.

[14] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Transactions on networking*, vol. 1, no. 2, pp. 166–177, 1993.

[15] N. Funabiki and Y. Takefuji, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 828–831, 1993.

[16] M. L. Huson and A. Sen, "Broadcast scheduling algorithms for radio networks," in *Proceedings of MILCOM'95*, vol. 2, pp. 647–651, IEEE, 1995.

[17] C.-J. Su, L. Tassiulas, and V. J. Tsotras, "Broadcast scheduling for information distribution," *Wireless Networks*, vol. 5, no. 2, pp. 137–147, 1999.

[18] P.-K. Hung, J.-P. Sheu, and C.-S. Hsu, "Scheduling of broadcasts in multihop wireless networks," in *European Wireless*, 2002.

[19] B. d. Reyck and Z. Degraeve, "Broadcast scheduling for mobile advertising," *Operations Research*, vol. 51, no. 4, pp. 509–517, 2003.

[20] S. Salcedo-Sanz, C. Bousoño-Calzón, and A. R. Figueiras-Vidal, "A mixed neural-genetic algorithm for the broadcast scheduling problem," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 277–283, 2003.

[21] A. Behzad and I. Rubin, "On the performance of graph-based scheduling algorithms for packet radio networks," in *GLOBECOM'03. IEEE Global Telecommunications Conference*, vol. 6, pp. 3432–3436, IEEE, 2003.

[22] J.-C. Chen, Y.-C. Wang, and J.-T. Chen, "A novel broadcast scheduling strategy using factor graphs and the sum-product algorithm," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1241–1249, 2006.

[23] Z. Chen, C. Qiao, J. Xu, and T. Lee, "A constant approximation algorithm for interference aware broadcast in wireless networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pp. 740–748, IEEE, 2007.

[24] Y. Wang and I. Henning, "A deterministic distributed tdma scheduling algorithm for wireless sensor networks," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2759–2762, IEEE, 2007.

[25] S. C.-H. Huang, P.-J. Wan, J. Deng, and Y. S. Han, "Broadcast scheduling in interference environment," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1338–1348, 2008.

[26] S. Menon and R. Gupta, "Optimal broadcast scheduling in packet radio networks via branch and price," *INFORMS Journal on Computing*, vol. 20, no. 3, pp. 391–399, 2008.

[27] R. Mahjourian, F. Chen, R. Tiwari, M. Thai, H. Zhai, and Y. Fang, "An approximation algorithm for conflict-aware broadcast scheduling in wireless

ad hoc networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pp. 331–340, 2008.

[28] C. Zhan and Y. Xu, "Broadcast scheduling based on network coding in time critical wireless networks," in *2010 IEEE International Symposium on Network Coding (NetCod)*, pp. 1–6, IEEE, 2010.

[29] R. Tiwari, T. N. Dinh, and M. T. Thai, "On centralized and localized approximation algorithms for interference-aware broadcast scheduling," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 233–247, 2011.

[30] D. Arivudainambi and D. Rekha, "Broadcast scheduling problem for tdma ad-hoc networks," in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief*, pp. 461–464, 2011.

[31] M. Cheng and Q. Ye, "Transmission scheduling based on a new conflict graph model for multicast in multihop wireless networks," in *2012 IEEE Global Communications Conference*, pp. 5717–5722, IEEE, 2012.

[32] S. Ji, R. Beyah, and Z. Cai, "Minimum-latency broadcast scheduling for cognitive radio networks," in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pp. 389–397, IEEE, 2013.

[33] E. L. Lawler, "A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness," in *Annals of discrete Mathematics*, vol. 1, pp. 331–342, Elsevier, 1977.

[34] J. Du and J. Y.-T. Leung, "Minimizing total tardiness on one machine is np-hard," *Mathematics of Operations Research*, vol. 15, no. 3, pp. 483–495, 1990.

[35] S. V. Mehta and R. Uzsoy, "Minimizing total tardiness on a batch processing machine with incompatible job families," *IIE Transactions*, vol. 30, no. 2, p. 165–178, 1998.

[36] M. Demangee, D. D. Werra, J. Monnot, and V. T. Paschos, "Time slot scheduling of compatible jobs," *Journal of Scheduling*, vol. 10, no. 2, p. 111–127, 2007.

[37] G. A. Süer, X. Yang, O. I. Alhawari, J. Santos, and R. Vazquez, "A genetic algorithm approach for minimizing total tardiness in single machine scheduling," *International Journal of Industrial Engineering and Management (IJIEM)*, vol. 3, no. 3, pp. 163–171, 2012.

[38] J. M. Framinan and P. Perez-Gonzalez, "Order scheduling with tardiness objective: Improved approximate solutions," *European Journal of Operational Research*, vol. 266, no. 3, pp. 840–850, 2018.

[39] L. Mönch and S. Roob, "A matheuristic framework for batch machine scheduling problems with incompatible job families and regular sum objective," *Applied Soft Computing*, vol. 68, pp. 835–846, 2018.

[40] M.-C. Costa, "Persistency in maximum cardinality bipartite matchings," *Operations Research Letters*, vol. 15, no. 3, pp. 143–149, 1994.

[41] G. Steiner and J. S. Yeomans, "A linear time algorithm for maximum matchings in convex, bipartite graphs," *Computers & Mathematics with Applications*, vol. 31, no. 12, pp. 91–96, 1996.

[42] C. R. Coullard, A. Gamble, and P. Jones, "Matching problems in selective assembly operations," *Annals of Operations Research*, vol. 76, pp. 95–107, 1998.

[43] Y. Wang, F. Makedon, J. Ford, and H. Huang, "A bipartite graph matching framework for finding correspondences between structural elements in two proteins," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 2972–2975, IEEE, 2004.

[44] J. E. Hopcroft and R. M. Karp, "An nˆ5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing*, vol. 2, no. 4, pp. 225–231, 1973.

[45] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, pp. 399–404, 1956.

Appendix A

Detailed Computational Results of Random Instances

Table A.1: MIP Scheduling Model Results of 50 Instances

Parameter Values							Scheduling Model Results					
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Value	Max # of Timeslots	CPU Time (s)	MIP Gap (%)	Iteration Count	Best Bound	Obj. Value	Improved (%)
13	643	169.37	3.97	6.25	23.52	59	0.08	0.00	1350	35.0	35.0	40.68
15	694	102.08	3.18	3.90	23.49	75	0.18	0.00	3043	39.0	39.0	48.00
11	689	80.62	2.71	3.76	17.28	57	0.05	0.00	915	30.0	30.0	47.37
19	855	50.04	4.72	5.25	9.71	79	0.23	0.00	158	41.0	41.0	48.10
14	778	187.29	2.80	3.15	28.61	74	0.53	0.00	4710	42.0	42.0	43.24
20	525	156.77	4.09	9.09	20.38	96	0.24	0.00	31	50.0	50.0	47.92
20	722	137.35	3.78	4.21	8.13	102	0.28	0.00	31	51.0	51.0	50.00
19	742	114.56	3.19	6.56	15.31	98	0.23	0.00	30	50.0	50.0	48.98
20	716	140.03	2.22	4.17	5.12	101	0.38	0.00	760	51.0	51.0	49.50
18	795	50.60	3.71	8.81	13.12	86	0.17	0.00	29	47.0	47.0	45.35
13	967	164.32	3.45	6.18	9.87	68	0.08	0.00	1240	37.0	37.0	45.59
15	716	65.17	4.51	4.96	16.16	60	0.10	0.00	1392	31.0	31.0	48.33
11	781	56.74	1.24	6.45	8.34	52	0.07	0.00	877	31.0	31.0	40.38
11	702	61.09	4.54	3.60	6.83	57	0.12	0.00	1805	57.0	57.0	0.00
17	780	79.54	4.96	2.53	14.64	82	0.33	0.00	51	82.0	82.0	0.00
16	812	59.65	2.60	2.88	8.65	78	1.10	0.00	5867	40.0	40.0	48.72
17	978	119.21	1.44	8.51	12.60	94	0.16	0.00	27	50.0	50.0	46.81
19	564	116.57	4.52	1.58	29.44	100	0.52	0.00	1505	100.0	100.0	0.00
11	512	61.56	1.22	6.47	12.98	57	0.06	0.00	661	29.0	29.0	49.12
18	787	65.39	3.25	1.04	22.97	99	0.57	0.00	2948	99.0	99.0	0.00
16	600	165.74	1.37	1.05	8.87	88	0.29	0.00	53	88.0	88.0	0.00
11	729	135.85	3.23	5.07	8.75	53	0.05	0.00	835	28.0	28.0	47.17
10	637	92.86	2.98	3.33	8.31	46	0.10	0.00	1720	24.0	24.0	47.83
17	909	50.80	4.91	7.31	14.26	82	0.22	0.00	2571	41.0	41.0	50.00
17	998	55.95	2.53	7.38	14.63	90	0.15	0.00	30	50.0	50.0	44.44
16	545	171.37	4.91	6.86	20.41	76	0.12	0.00	1497	40.0	40.0	47.37
19	881	191.54	3.92	9.69	12.34	96	0.34	0.00	809	48.0	48.0	50.00
17	868	67.48	1.45	9.89	10.63	74	0.14	0.00	1868	39.0	39.0	47.30
17	617	123.71	4.37	7.16	24.16	83	0.14	0.00	28	44.0	44.0	46.99
14	976	199.30	1.85	4.33	23.84	66	0.08	0.00	1252	35.0	35.0	46.97
11	529	184.43	2.92	4.07	8.13	56	0.06	0.00	907	31.0	31.0	44.64
11	665	90.00	3.91	6.27	12.11	58	0.05	0.00	872	32.0	32.0	44.83
17	519	154.45	2.87	9.20	5.83	74	0.11	0.00	1704	40.0	40.0	45.95
18	604	94.52	3.43	7.40	27.98	100	0.16	0.00	30	51.0	51.0	49.00
20	935	128.00	3.53	3.51	6.55	99	10.27	0.00	37923	58.0	58.0	41.41
10	900	66.87	3.79	3.23	29.78	49	0.08	0.00	1290	49.0	49.0	0.00
15	619	107.69	1.95	8.54	17.15	82	0.17	0.00	1744	43.0	43.0	47.56
12	703	115.08	2.16	1.18	13.34	62	0.13	0.00	1726	62.0	62.0	0.00
11	976	198.70	3.61	9.45	14.87	45	0.05	0.00	622	23.0	23.0	48.89
13	599	103.14	4.57	8.44	13.65	57	0.07	0.00	1025	30.0	30.0	47.37
15	923	82.59	2.37	3.80	5.37	74	0.13	0.00	2005	38.0	38.0	48.65
20	730	112.85	4.09	6.11	20.34	99	0.19	0.00	33	50.0	50.0	49.49
13	686	128.23	3.94	7.44	13.69	55	0.06	0.00	1114	30.0	30.0	45.45
19	667	81.75	2.97	1.53	17.14	95	0.61	0.00	2965	95.0	95.0	0.00
15	898	178.92	1.86	6.37	16.88	66	0.09	0.00	1318	35.0	35.0	46.97
19	859	132.88	4.03	6.04	19.11	91	0.19	0.00	36	46.0	46.0	49.45
15	618	56.91	3.33	8.53	23.07	78	0.12	0.00	2047	42.0	42.0	46.15
20	689	184.22	3.34	9.17	11.57	95	0.18	0.00	2080	48.0	48.0	49.47
13	604	111.63	1.81	5.24	9.38	72	0.08	0.00	1476	39.0	39.0	45.83
10	991	52.26	3.75	8.07	14.11	47	0.04	0.00	667	25.0	25.0	46.81

Table A.2: Scheduling Model and Algorithm Comparison of 50 Instances

Parameter Values							Algorithm Results			Scheduling Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Value	Max Timeslot Number	Soln. Value	Improved From Max (%)	Soln. Time (s)	CPU Time (s)	Iter. Count	Obj. Value	Improved From Max (%)
8	1597	47.01	4.17	9.11	19.89	32	20	37.50	0.0001	0.02	354	20.0	37.50
8	949	39.52	3.11	2.68	26.35	37	37	0.00	0.0000	0.05	774	37.0	0.00
13	1197	61.87	4.31	5.33	13.91	69	39	43.48	0.0003	0.12	1241	37.0	46.38
14	998	57.60	4.35	1.36	27.97	78	78	0.00	0.0004	0.19	47	78.0	0.00
5	1690	43.33	4.37	8.36	28.52	26	18	30.77	0.0001	0.02	298	14.0	46.15
28	1350	41.57	2.45	9.46	25.19	143	78	45.45	0.0006	0.57	1481	78.0	45.45
22	1917	60.84	1.96	8.51	26.61	91	50	45.05	0.0004	0.24	304	47.0	48.35
18	1754	62.93	2.94	3.92	16.13	90	48	46.67	0.0002	0.16	28	47.0	47.78
12	710	33.94	1.17	1.93	29.51	61	32	47.54	0.0002	0.05	788	31.0	49.18
27	1879	33.02	3.23	4.51	18.77	135	74	45.19	0.0005	0.55	2216	72.0	46.67
10	1621	34.55	3.92	6.56	25.20	37	22	40.54	0.0002	0.04	634	20.0	45.95
21	943	58.64	3.10	8.52	14.71	101	55	45.54	0.0005	0.22	34	52.0	48.51
8	1501	64.50	2.99	2.38	20.98	43	39	9.30	0.0001	0.05	948	39.0	9.30
6	1805	50.07	2.45	1.84	5.06	36	36	0.00	0.0001	0.04	642	36.0	0.00
29	608	38.11	2.81	3.85	24.92	156	83	46.79	0.0006	0.89	3727	82.0	47.44
20	731	58.59	4.57	1.16	17.56	103	103	0.00	0.0001	0.68	2070	103.0	0.00
21	1008	37.44	4.96	5.29	11.35	92	56	39.13	0.0003	0.29	31	48.0	47.83
15	1879	49.10	2.45	9.08	14.94	75	42	44.00	0.0002	0.09	1303	42.0	44.00
26	621	68.99	1.90	9.34	15.42	129	69	46.51	0.0005	0.34	39	66.0	48.84
12	1831	38.27	1.32	3.17	6.34	61	37	39.34	0.0001	0.06	1062	35.0	42.62
24	957	51.36	2.03	6.72	9.77	111	60	45.95	0.0005	0.28	36	57.0	48.65
12	1511	67.43	1.13	2.87	28.79	65	38	41.54	0.0001	0.06	1032	35.0	46.15
6	1492	50.85	4.66	2.75	9.44	36	36	0.00	0.0001	0.03	623	36.0	0.00
27	1988	47.62	2.66	8.89	7.65	153	86	43.79	0.0005	0.66	1808	81.0	47.06
28	806	59.27	2.42	2.23	15.61	130	128	1.54	0.0003	1.81	2304	128.0	1.54
27	1715	45.55	2.78	9.38	24.25	143	77	46.15	0.0005	0.41	41	74.0	48.25
6	1376	42.80	1.98	1.54	28.13	29	29	0.00	0.0001	0.03	581	29.0	0.00
5	1401	43.22	1.80	1.63	23.83	30	30	0.00	0.0001	0.02	429	30.0	0.00
20	1683	66.54	3.85	3.46	21.71	101	95	5.94	0.0002	0.83	3240	95.0	5.94
5	1203	49.34	2.62	6.18	6.71	27	18	33.33	0.0001	0.02	273	14.0	48.15
18	1497	53.09	4.87	2.52	27.41	84	84	0.00	0.0002	0.36	47	84.0	0.00
24	1307	46.82	1.90	1.82	15.83	117	117	0.00	0.0003	1.36	2170	117.0	0.00
6	745	66.85	2.22	6.01	20.46	23	13	43.48	0.0001	0.01	147	13.0	43.48
14	1002	43.95	3.88	3.93	16.55	69	51	26.09	0.0002	0.72	4267	48.0	30.43
11	686	64.96	3.99	9.46	10.80	48	27	43.75	0.0005	0.06	706	27.0	43.75
26	1909	47.65	1.91	4.29	22.65	131	71	45.80	0.0009	0.47	38	69.0	47.33
27	1088	50.00	3.09	7.42	19.65	129	67	48.06	0.0007	0.48	41	67.0	48.06
16	1465	60.43	3.34	7.99	21.71	72	40	44.44	0.0003	0.14	1677	40.0	44.44
12	897	49.59	2.80	5.37	29.67	50	29	42.00	0.0004	0.06	898	29.0	42.00
9	1848	50.86	3.68	8.95	9.04	50	34	32.00	0.0001	0.08	845	26.0	48.00
15	1091	39.66	3.78	1.79	6.40	73	73	0.00	0.0002	0.24	46	73.0	0.00
19	1973	60.01	3.80	9.99	9.74	94	49	47.87	0.0003	0.18	2010	47.0	50.00
10	1631	69.08	4.23	5.24	9.62	47	26	44.68	0.0002	0.05	696	26.0	44.68
29	1924	59.79	3.44	2.88	21.77	154	154	0.00	0.0003	1.78	1069	154.0	0.00
9	1712	64.05	4.57	9.43	12.68	49	27	44.90	0.0001	0.04	733	27.0	44.90
9	511	57.16	2.07	9.54	23.51	61	38	37.70	0.0001	0.06	725	31.0	49.18
29	970	66.63	2.07	5.33	14.47	150	78	48.00	0.0008	0.84	2113	77.0	48.67
12	1548	65.65	3.31	3.22	9.26	58	49	15.52	0.0002	0.31	2365	45.0	22.41
21	1074	45.68	2.90	9.68	28.64	108	56	48.15	0.0005	0.45	1202	54.0	50.00
9	906	34.00	3.41	5.58	25.36	48	31	35.42	0.0002	0.07	653	26.0	45.83

Table A.3: LP Matching Model Results of 50 Instances

Parameter Values										Matching Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Value	Sup. Nodes	Dem. Nodes	Feasible Arcs	Arc Ratio (%)	CPU Time (s)	Matched Pairs	Not Matched	Obj. Value
78	837	49.67	3.34	6.65	20.27	190	189	35456	98.74	0.0681	189.0	1.0	190.0
39	772	61.90	4.73	7.01	22.92	107	105	10880	96.84	0.0195	105.0	2.0	107.0
80	721	34.92	1.61	1.24	27.40	188	193	0	0.00	0.0006	0	381	381.0
81	643	33.66	3.99	9.38	17.11	230	198	44957	98.72	0.0828	198.0	32.0	230.0
22	1802	38.39	2.66	6.39	17.79	50	51	2429	95.25	0.0061	50.0	1.0	51.0
38	1327	47.41	1.97	7.75	19.79	87	94	7963	97.37	0.0163	87.0	7.0	94.0
37	1648	33.54	1.65	3.46	22.24	94	80	7312	97.23	0.0171	80.0	14.0	94.0
40	1910	48.44	2.54	4.74	22.86	110	102	10926	97.38	0.0230	102.0	8.0	110.0
89	1753	60.78	2.88	2.21	14.01	233	227	310	0.59	0.0013	2.0	456.0	458.0
96	930	48.56	3.79	5.82	28.60	238	241	56062	97.74	0.1030	238.0	3.0	241.0
63	1646	51.35	3.81	9.47	10.14	156	159	24415	98.43	0.0409	156.0	3.0	159.0
91	1236	53.87	4.37	6.05	5.96	242	225	52854	97.07	0.1012	225.0	17.0	242.0
70	877	54.17	3.08	5.83	6.15	189	186	34667	98.61	0.0573	186.0	3.0	189.0
15	940	45.80	3.34	3.98	6.96	36	34	1040	84.97	0.0019	34.0	2.0	36.0
47	674	39.10	3.79	5.31	6.99	115	132	14514	95.61	0.0288	115.0	17.0	132.0
97	1028	30.57	1.61	9.86	12.59	251	242	60107	98.95	0.1083	242.0	9.0	251.0
98	512	53.03	1.19	5.02	7.07	251	249	61857	98.97	0.1093	249.0	2.0	251.0
11	1945	59.63	4.90	1.69	29.30	36	27	0	0.00	0.0009	0	63	63.0
37	1378	46.76	1.39	2.58	21.38	90	92	7986	96.45	0.0150	90.0	2.0	92.0
87	775	37.35	4.53	6.79	12.60	229	197	44196	97.97	0.0749	197.0	32.0	229.0
17	1794	38.83	1.36	7.48	9.85	44	46	1905	94.12	0.0046	44.0	2.0	46.0
23	1937	61.58	3.88	7.40	25.59	58	58	3228	95.96	0.0085	58.0	0.0	58.0
26	947	55.62	1.89	4.81	6.25	63	60	3635	96.16	0.0080	60.0	3.0	63.0
26	1214	55.42	3.61	5.27	16.02	63	64	3833	95.06	0.0078	63.0	1.0	64.0
70	1439	48.93	4.86	4.89	27.84	185	170	8021	25.50	0.0151	158.0	39.0	197.0
31	722	68.11	2.60	9.22	10.44	87	79	6666	96.99	0.0138	79.0	8.0	87.0
16	1995	68.66	2.32	3.66	21.21	35	44	1397	90.71	0.0039	35.0	9.0	44.0
76	1282	44.62	2.86	5.74	6.41	185	170	31019	98.63	0.0527	170.0	15.0	185.0
44	1486	65.38	3.94	5.79	18.63	100	120	11615	96.79	0.0217	100.0	20.0	120.0
70	1882	47.63	1.31	5.55	19.91	175	182	31399	98.58	0.0576	175.0	7.0	182.0
14	1602	62.17	2.85	6.90	26.56	35	36	1170	92.86	0.0019	35.0	1.0	36.0
78	1272	53.38	2.54	5.07	18.17	191	197	37142	98.71	0.0850	191.0	6.0	197.0
40	1797	31.14	4.43	8.74	9.03	98	101	9632	97.31	0.0180	98.0	3.0	101.0
32	1837	39.55	1.33	3.20	29.76	85	92	7577	96.89	0.0146	85.0	7.0	92.0
63	1873	65.85	3.80	9.63	6.51	167	152	24974	98.38	0.0397	152.0	15.0	167.0
71	620	52.64	4.81	8.94	14.14	185	180	32814	98.54	0.0661	180.0	5.0	185.0
91	1327	51.78	1.91	8.59	13.69	225	225	50042	98.85	0.0839	225.0	0.0	225.0
36	1272	41.36	1.26	7.28	19.83	88	82	7018	97.26	0.0154	82.0	6.0	88.0
41	1490	66.49	2.76	7.68	27.14	109	103	10946	97.50	0.0199	103.0	6.0	109.0
72	1758	58.65	3.15	6.74	25.96	182	171	30685	98.60	0.0534	171.0	11.0	182.0
66	738	40.52	1.68	8.00	5.05	159	167	26163	98.53	0.0463	159.0	8.0	167.0
93	578	61.99	1.74	9.87	21.81	250	237	58596	98.90	0.1035	237.0	13.0	250.0
62	893	44.42	4.54	8.02	12.78	149	146	21395	98.35	0.0367	146.0	3.0	149.0
49	741	60.84	3.69	6.95	24.09	123	113	13621	98.00	0.0226	113.0	10.0	123.0
35	942	51.64	1.15	1.13	12.65	84	86	20	0.28	0.0008	4.0	162.0	166.0
75	1765	66.46	2.01	9.10	15.63	185	187	34138	98.68	0.0632	185.0	2.0	187.0
13	1910	62.90	3.18	7.82	14.43	35	36	1162	92.22	0.0022	35.0	1.0	36.0
12	1158	34.42	1.82	1.49	29.31	31	25	0	0.00	0.0009	0	56	56.0
56	1356	32.96	3.16	1.31	29.20	151	124	0	0.00	0.0006	0	275	275.0
40	1215	64.60	2.74	5.80	23.72	107	100	10435	97.52	0.0208	100.0	7.0	107.0

Table A.4: Matching Model and Hopcroft-Karp Algorithm Comparison of 50 Instances

Parameter Values										Model & Hopcroft-Karp Algorithm Results					
# of STAs	Circle Radius	Min Dist.	δ Val.	β Val.	Ω Val.	Sup. Nodes	Dem. Nodes	Feas. Arcs	Arc Ratio (%)	Matched Pairs	Not Matched	Imp. (%)	HK Time (s)	CPU Time (s)	Obj. Val.
38	711	42.08	3.67	8.43	15.56	94	102	9336	97.37	94.0	8.0	47.96	0.0021	0.0196	102.0
93	1051	69.07	1.07	3.51	25.29	226	224	50100	98.96	224.0	2.0	49.78	0.0066	0.0923	226.0
84	1538	30.32	4.04	8.48	21.31	184	218	39642	98.83	184.0	34.0	45.77	0.0081	0.0641	218.0
65	1581	34.21	1.58	8.19	29.39	156	181	27804	98.47	156.0	25.0	46.29	0.0032	0.0434	181.0
20	1066	47.17	4.10	3.51	24.22	48	47	152	6.74	4.0	87.0	4.21	0.0007	0.0011	91.0
45	1679	56.27	4.06	6.57	25.58	123	117	14034	97.52	117.0	6.0	48.75	0.0027	0.0304	123.0
72	567	62.95	4.34	1.48	8.22	180	179	0	0.00	0	359	0.00	0.0032	0.0006	359.0
26	1865	35.56	2.54	9.34	24.78	79	68	5164	96.13	68.0	11.0	46.26	0.0017	0.0123	79.0
35	1275	33.88	1.58	7.28	20.83	84	84	6858	97.19	84.0	0.0	50.00	0.0019	0.0130	84.0
32	1704	39.70	1.87	3.87	22.49	86	76	6326	96.79	76.0	10.0	46.91	0.0018	0.0129	86.0
48	1224	61.12	3.80	8.39	7.99	114	132	14740	97.95	114.0	18.0	46.34	0.0027	0.0350	132.0
10	788	52.09	1.37	3.43	14.05	17	28	429	90.13	17.0	11.0	37.78	0.0008	0.0016	28.0
47	881	42.89	1.77	6.51	19.14	122	130	15529	97.91	122.0	8.0	48.41	0.0029	0.0280	130.0
41	847	42.49	4.35	3.42	18.53	110	106	362	3.10	7.0	202.0	3.24	0.0016	0.0013	209.0
74	1541	33.51	3.29	2.09	17.52	200	188	0	0.00	0	388	0.00	0.0052	0.0008	388.0
74	887	32.82	2.75	1.74	23.89	180	178	0	0.00	0	358	0.00	0.0035	0.0006	358.0
71	1445	43.98	1.95	9.83	9.94	182	168	30149	98.60	168.0	14.0	48.00	0.0045	0.0542	182.0
89	1852	30.81	2.22	9.31	15.58	224	226	50059	98.88	224.0	2.0	49.78	0.0061	0.1131	226.0
30	1163	35.26	1.76	2.64	18.34	74	67	4509	90.94	67.0	7.0	47.52	0.0014	0.0089	74.0
88	1620	47.43	1.72	2.18	20.21	201	220	32642	73.82	201.0	19.0	47.74	0.0068	0.0642	220.0
95	525	61.69	2.41	5.12	5.12	247	245	59868	98.93	245.0	2.0	49.80	0.0071	0.1061	247.0
72	897	54.20	2.64	4.48	20.54	177	164	28479	98.11	164.0	13.0	48.09	0.0121	0.0528	177.0
76	1377	34.88	3.43	6.58	5.82	197	190	36915	98.62	190.0	7.0	49.10	0.0053	0.0709	197.0
10	874	30.23	4.11	3.04	18.11	23	23	0	0.00	0	46	0.00	0.0004	0.0005	46.0
51	1488	60.65	1.50	1.20	20.83	117	125	0	0.00	0	242	0.00	0.0019	0.0005	242.0
93	625	52.68	4.35	9.47	6.57	237	220	51568	98.90	220.0	17.0	48.14	0.0072	0.0904	237.0
65	1211	42.05	4.61	3.69	13.78	159	163	606	2.34	6.0	310.0	1.86	0.0024	0.0026	316.0
11	744	32.09	2.72	7.05	22.62	24	26	576	92.31	24.0	2.0	48.00	0.0007	0.0014	26.0
20	1895	55.10	3.64	2.36	22.72	39	58	0	0.00	0	97	0.00	0.0011	0.0007	97.0
90	1577	31.32	3.74	5.09	14.38	244	231	54206	96.17	231.0	13.0	48.63	0.0074	0.0989	244.0
67	1646	53.48	1.04	1.02	25.00	159	174	0	0.00	0	333	0.00	0.0023	0.0007	333.0
94	628	57.67	1.55	9.66	12.18	245	245	59377	98.92	245.0	0.0	50.00	0.0081	0.1108	245.0
87	899	42.20	2.52	1.90	29.39	224	230	10	0.02	2.0	450.0	0.44	0.0040	0.0008	452.0
98	1042	54.05	2.27	3.91	10.87	235	239	55308	98.47	235.0	4.0	49.58	0.0071	0.0891	239.0
16	1744	57.12	1.81	5.24	15.89	38	38	1354	93.77	38.0	0.0	50.00	0.0007	0.0024	38.0
76	1255	56.56	2.29	9.45	7.46	194	201	38476	98.67	194.0	7.0	49.11	0.0053	0.0965	201.0
94	717	41.48	4.82	5.71	25.94	225	219	41360	83.94	219.0	6.0	49.32	0.0054	0.1428	225.0
22	1648	59.25	2.65	9.51	5.70	57	55	2992	95.44	55.0	2.0	49.11	0.0013	0.0045	57.0
98	1972	38.41	1.27	2.60	17.07	260	255	65496	98.79	255.0	5.0	49.51	0.0100	0.1367	260.0
62	1884	39.47	3.18	3.14	22.39	155	170	1690	6.41	52.0	221.0	16.00	0.0060	0.0027	273.0
25	884	44.89	4.81	8.09	29.89	66	58	3664	95.72	58.0	8.0	46.77	0.0012	0.0158	66.0
80	1974	48.16	4.36	3.36	10.65	211	228	0	0.00	0	439	0.00	0.0047	0.0008	439.0
53	1072	41.51	4.14	4.39	14.60	120	134	9130	56.78	120.0	14.0	47.24	0.0027	0.0163	134.0
19	1031	39.18	4.13	5.35	19.57	53	47	2053	82.42	47.0	6.0	47.00	0.0017	0.0031	53.0
56	1868	55.55	1.17	9.46	8.32	132	140	18154	98.24	132.0	8.0	48.53	0.0030	0.0413	140.0
98	1433	45.64	1.27	1.49	9.70	233	261	24234	39.85	223.0	48.0	45.14	0.0061	0.0522	271.0
14	1203	56.51	1.53	8.70	17.67	44	30	1222	92.58	30.0	14.0	40.54	0.0009	0.0024	44.0
50	841	51.04	1.08	3.66	11.34	130	123	15675	98.03	123.0	7.0	48.62	0.0026	0.0428	130.0
57	994	64.93	2.47	5.00	23.32	151	139	20628	98.28	139.0	12.0	47.93	0.0060	0.0464	151.0
60	1093	65.42	1.50	2.33	9.18	153	164	24304	96.86	153.0	11.0	48.26	0.0040	0.0605	164.0

Table A.5: MIP Deadline Model Results of 50 Instances

Parameter Values						Deadline Model Results					
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Value	Minimum Makespan	CPU Time (s)	MIP Gap (%)	Iter. Count	# of Tardy STAs	Minimum Tardiness
3	1767	36.52	3.12	1.86	28.39	16	0.08	0.00	188	2	3.0
4	684	54.33	4.08	3.84	13.21	17	0.09	0.00	349	2	8.0
8	606	38.36	1.57	7.13	8.38	20	0.13	0.00	997	4	7.0
7	608	59.04	4.74	3.00	11.05	29	0.02	0.00	255	0	0.0
7	1744	50.83	1.45	6.54	20.27	19	0.01	0.00	221	1	1.0
3	1438	69.02	1.98	7.17	22.35	8	0.01	0.00	52	2	2.0
3	1804	56.71	3.76	8.00	8.25	10	0.01	0.00	48	1	1.0
6	628	45.24	2.21	7.58	9.77	16	0.02	0.00	202	2	2.0
4	1773	64.21	4.88	9.04	14.16	16	0.01	0.00	95	1	3.0
4	1290	67.61	1.86	4.16	21.86	11	0.01	0.00	0	0	0.0
8	1094	33.69	4.04	4.72	23.62	22	0.03	0.00	271	0	0.0
7	1625	37.04	4.21	9.45	5.31	19	0.03	0.00	224	2	3.0
6	675	52.51	2.28	8.83	23.50	16	0.03	0.00	303	3	4.0
6	618	54.61	1.15	5.58	14.03	19	0.01	0.00	167	0	0.0
4	951	50.10	4.59	4.44	7.04	15	0.03	0.00	372	1	4.0
7	993	38.07	1.03	8.88	20.74	20	0.03	0.00	295	1	1.0
6	1604	42.50	1.17	2.95	6.32	20	0.04	0.00	301	2	2.0
5	1399	64.46	2.55	2.01	28.06	26	0.16	0.00	1554	2	12.0
4	1259	42.65	1.19	5.57	16.92	9	0.02	0.00	268	2	4.0
9	1967	53.23	4.18	8.85	21.53	24	0.21	0.00	909	3	3.0
3	1520	57.56	1.91	2.38	6.55	9	0.01	0.00	35	1	1.0
6	1805	44.46	3.81	4.62	20.96	18	0.03	0.00	295	2	3.0
5	1925	30.97	2.26	5.57	10.10	14	0.01	0.00	51	0	0.0
4	1379	46.04	4.84	1.95	6.74	21	0.07	0.00	448	2	7.0
5	1094	41.27	1.45	1.24	11.94	28	0.39	0.00	21792	3	16.0
4	979	59.08	3.71	6.40	23.83	12	0.01	0.00	0	0	0.0
2	695	66.84	2.89	6.50	17.92	5	0.00	0.00	0	0	0.0
4	912	31.57	3.66	8.00	13.11	9	0.00	0.00	0	0	0.0
8	1244	65.89	2.00	2.09	17.63	45	0.33	0.00	1578	3	9.0
2	1346	63.96	2.96	4.94	7.51	5	0.00	0.00	0	0	0.0
4	840	51.35	1.20	7.92	19.13	12	0.01	0.00	64	0	0.0
9	614	40.84	4.04	5.70	29.51	22	0.21	0.00	1267	3	6.0
7	621	35.07	1.29	9.26	9.97	22	0.05	0.00	551	2	3.0
1	1150	60.38	2.33	4.13	10.13	5	0.00	0.00	0	0	0.0
8	557	54.88	1.16	5.18	6.43	23	0.04	0.00	487	2	3.0
8	520	58.84	1.01	9.17	9.31	26	0.66	0.00	6110	2	9.0
9	1801	51.10	1.15	7.32	20.71	21	0.06	0.00	531	1	3.0
6	832	46.47	3.06	6.99	17.08	16	0.01	0.00	129	1	1.0
3	846	55.95	1.95	7.81	5.43	9	0.00	0.00	0	0	0.0
4	1171	65.26	1.58	4.77	14.61	13	0.02	0.00	186	2	2.0
1	1763	35.55	2.24	3.27	23.06	3	0.00	0.00	0	0	0.0
6	729	62.89	1.31	9.67	21.67	14	0.01	0.00	220	1	1.0
5	1763	52.66	2.01	6.18	19.67	14	0.09	0.00	1195	2	6.0
8	1924	45.16	2.44	1.40	13.47	44	0.16	0.00	1004	2	8.0
3	611	54.29	1.46	1.92	8.49	9	0.01	0.00	39	1	2.0
3	1934	55.51	1.09	4.18	19.76	10	0.01	0.00	73	1	2.0
1	1959	33.43	2.90	5.43	10.41	3	0.00	0.00	0	0	0.0
1	860	43.93	1.72	3.68	9.36	5	0.00	0.00	0	0	0.0
1	1602	38.75	3.81	4.11	14.46	4	0.00	0.00	0	0	0.0
4	1397	51.09	3.69	1.07	29.56	22	0.02	0.00	111	1	2.0

Table A.6: MIP Deadline Model and Algorithm Comparison of 50 Instances

Parameter Values						Algorithm Results		Deadline Model Results				
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Value	Soln. Value	Soln. Time (s)	CPU Time (s)	MIP Gap (%)	Iter. Count	Obj. Value	Improved (%)
6	1173	68.89	1.98	4.12	17.60	14.0	0.0027	0.17	0.00	1655	11.0	21.00
8	1377	47.66	4.59	8.38	6.08	6.0	0.0031	0.11	0.00	1194	4.0	33.00
5	1101	37.25	1.57	7.17	24.11	7.0	0.0008	0.03	0.00	390	4.0	43.00
8	1682	50.00	4.19	6.15	23.28	2.0	0.0020	0.03	0.00	434	1.0	50.00
3	1482	66.12	2.88	4.71	23.96	3.0	0.0007	0.02	0.00	178	3.0	0.00
4	1116	38.82	1.76	3.42	25.88	4.0	0.0010	0.01	0.00	169	3.0	25.00
6	1476	54.26	2.05	2.47	13.60	8.0	0.0017	0.09	0.00	1133	5.0	38.00
4	1904	47.07	1.23	5.25	22.78	1.0	0.0007	0.01	0.00	71	0.0	100.00
4	1222	52.21	2.81	3.06	21.36	3.0	0.0006	0.01	0.00	89	0.0	100.00
7	531	56.29	4.85	4.00	22.63	3.0	0.0007	0.04	0.00	394	3.0	0.00
7	562	30.43	1.67	9.91	18.36	9.0	0.0023	0.12	0.00	1548	6.0	33.00
8	1902	60.20	4.16	7.22	22.29	13.0	0.0039	0.17	0.00	1298	10.0	23.00
8	958	46.43	4.17	9.77	10.08	12.0	0.0037	0.09	0.00	894	6.0	50.00
8	646	61.06	2.99	7.80	20.33	16.0	0.0034	0.34	0.00	7412	11.0	31.00
8	1713	35.49	4.40	4.78	5.42	6.0	0.0033	0.01	0.00	307	0.0	100.00
7	1246	59.34	3.85	3.38	5.66	33.0	0.0007	1.23	0.00	73086	21.0	39.00
7	1624	31.93	3.97	4.59	8.37	5.0	0.0018	0.06	0.00	796	3.0	40.00
5	1348	42.24	4.10	3.26	15.80	5.0	0.0005	0.05	0.00	386	4.0	20.00
7	1118	33.04	4.35	5.49	6.72	0.0	0.0023	0.01	0.00	193	0.0	0.00
5	754	57.24	4.13	8.44	18.00	3.0	0.0009	0.01	0.00	145	3.0	0.00
3	1548	66.10	1.76	9.23	6.20	6.0	0.0005	0.02	0.00	136	3.0	50.00
3	1148	36.71	3.95	8.99	24.62	3.0	0.0004	0.01	0.00	40	1.0	67.00
5	1980	56.77	3.62	8.68	26.87	1.0	0.0014	0.01	0.00	134	1.0	0.00
3	1879	52.46	1.39	5.87	12.36	2.0	0.0006	0.01	0.00	40	1.0	50.00
6	1689	46.59	1.51	6.93	20.77	5.0	0.0019	0.03	0.00	274	3.0	40.00
4	1456	30.80	4.35	8.61	19.25	10.0	0.0010	0.05	0.00	760	10.0	0.00
5	696	57.41	4.03	3.45	10.82	3.0	0.0004	0.01	0.00	169	1.0	67.00
5	992	47.02	4.14	1.49	14.32	1.0	0.0004	0.01	0.00	122	1.0	0.00
6	1984	67.46	2.52	6.01	30.00	3.0	0.0011	0.01	0.00	114	1.0	67.00
5	1211	35.17	2.41	8.20	24.84	10.0	0.0010	0.02	0.00	270	4.0	60.00
6	1255	68.57	1.39	3.57	20.94	4.0	0.0014	0.02	0.00	262	3.0	25.00
4	634	39.74	2.78	5.34	28.25	3.0	0.0009	0.01	0.00	113	0.0	100.00
3	1765	53.68	1.81	9.37	26.69	0.0	0.0006	0.00	0.00	0	0.0	0.00
8	1411	59.02	2.62	5.60	15.46	2.0	0.0044	0.04	0.00	521	1.0	50.00
3	1953	55.79	2.54	8.85	19.94	3.0	0.0004	0.01	0.00	95	2.0	33.00
7	640	40.83	2.66	5.59	25.04	5.0	0.0023	0.02	0.00	267	2.0	60.00
6	849	59.35	3.67	3.25	8.77	1.0	0.0005	0.01	0.00	229	1.0	0.00
3	844	36.87	4.41	8.11	22.82	1.0	0.0003	0.00	0.00	32	0.0	100.00
8	1091	55.11	3.85	6.89	29.57	3.0	0.0037	0.02	0.00	318	1.0	67.00
6	1360	31.62	1.21	1.75	17.76	4.0	0.0014	0.02	0.00	212	3.0	25.00
6	1337	49.24	1.56	8.19	14.06	0.0	0.0013	0.01	0.00	88	0.0	0.00
8	1004	69.56	3.35	8.16	14.46	11.0	0.0022	0.11	0.00	1133	7.0	36.00
5	1410	59.98	4.92	3.20	6.39	1.0	0.0004	0.01	0.00	148	1.0	0.00
6	869	33.84	2.85	1.49	27.96	4.0	0.0006	0.04	0.00	436	3.0	25.00
7	1889	68.97	4.17	9.33	7.25	3.0	0.0023	0.03	0.00	401	3.0	0.00
5	841	37.54	1.19	2.27	9.45	6.0	0.0012	0.02	0.00	310	4.0	33.00
5	1451	54.03	3.09	9.96	19.57	2.0	0.0007	0.01	0.00	84	1.0	50.00
3	1380	49.07	2.39	2.23	17.66	6.0	0.0002	0.02	0.00	264	6.0	0.00
4	1277	43.43	2.06	9.25	5.11	0.0	0.0008	0.01	0.00	66	0.0	0.00
4	1695	52.90	4.85	1.78	14.03	22.0	0.0003	0.22	0.00	25987	17.0	23.00

Table A.7: IP Flow Model Results of 50 Instances

Parameter Values							Flow Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Array	Max Conc.	CPU Time (s)	Iter. Count	Obj. Value	Makespan
82	1733	45.29	4.61	5.27	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.9902	455	322.0	90.0
93	1469	42.38	1.74	4.96	[5. 6. 7. 8. 9. 10.]	6	1.7437	524	345.0	121.0
47	659	69.71	3.52	6.21	[5. 10.]	2	0.2836	588	149.0	71.0
11	1146	42.71	1.31	4.20	[5. 6.67 8.33 10.]	4	0.0272	65	38.0	13.0
73	777	60.85	4.00	2.26	[5. 10.]	2	0.0086	0	0.0	349.0
75	899	39.94	2.81	8.34	[5. 6.25 7.5 8.75 10.]	5	0.9281	423	289.0	97.0
92	1550	36.50	1.07	9.52	[5.]	1	0.1929	430	228.0	259.0
92	1741	57.20	4.05	5.34	[5. 10.]	2	1.0528	1050	328.0	152.0
83	1617	68.15	4.08	6.71	[5.]	1	0.1503	425	206.0	214.0
23	831	34.47	4.26	2.65	[5. 6.67 8.33 10.]	4	0.0005	0	0.0	120.0
53	520	40.73	1.65	4.21	[5. 6.25 7.5 8.75 10.]	5	0.2839	288	187.0	75.0
18	605	68.91	2.64	8.14	[5. 6.25 7.5 8.75 10.]	5	0.0336	104	60.0	22.0
51	1691	69.44	4.96	1.12	[5. 6.67 8.33 10.]	4	0.0006	0	0.0	270.0
41	1200	69.10	4.67	8.57	[5. 7.5 10.]	3	0.2242	612	155.0	52.0
17	1313	61.57	3.09	9.95	[5.]	1	0.0080	90	35.0	39.0
37	613	59.32	3.66	6.48	[5. 7.5 10.]	3	0.1811	334	122.0	56.0
70	1616	39.26	4.02	8.61	[5. 6.67 8.33 10.]	4	1.3080	988	292.0	72.0
76	613	52.85	1.14	8.20	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	1.1384	936	269.0	90.0
86	1041	46.51	2.69	3.76	[5.]	1	0.1443	416	217.0	237.0
52	1590	52.24	4.45	4.14	[5. 6.67 8.33 10.]	4	0.0309	141	72.0	183.0
11	1188	48.73	2.72	3.87	[5. 7.5 10.]	3	0.0200	84	47.0	20.0
67	834	37.57	4.07	8.68	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	1.2598	909	280.0	70.0
36	1543	69.75	1.03	6.61	[5. 7.5 10.]	3	0.2163	460	129.0	46.0
33	1951	69.85	1.45	2.08	[5. 10.]	2	0.1372	399	120.0	63.0
67	1691	42.19	1.90	1.27	[5. 7.5 10.]	3	0.0010	0	0.0	326.0
69	1931	34.27	3.84	2.54	[5. 6.67 8.33 10.]	4	0.0006	0	0.0	338.0
28	1336	66.77	1.71	7.81	[5. 7.5 10.]	3	0.1230	311	105.0	55.0
54	1495	61.89	4.95	5.61	[5. 6.67 8.33 10.]	4	0.2784	348	223.0	59.0
79	1112	44.50	3.74	1.20	[5. 10.]	2	0.0005	0	0.0	372.0
38	1177	68.55	3.28	2.16	[5. 6.67 8.33 10.]	4	0.0009	0	0.0	206.0
43	505	36.11	1.97	2.99	[5. 6. 7. 8. 9. 10.]	6	2.1691	41463	168.0	47.0
43	1405	63.06	4.49	1.70	[5. 6.25 7.5 8.75 10.]	5	0.0009	0	0.0	214.0
29	531	52.33	4.44	2.52	[5. 7.5 10.]	3	0.0011	0	0.0	135.0
42	568	62.85	4.57	5.36	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.2717	384	146.0	47.0
84	798	62.11	1.67	6.49	[5. 6.25 7.5 8.75 10.]	5	1.1845	431	324.0	104.0
23	864	46.31	3.57	2.20	[5. 6.67 8.33 10.]	4	0.0009	0	0.0	114.0
15	1265	53.56	2.79	9.65	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0236	43	55.0	24.0
59	1287	56.36	2.47	1.24	[5. 6.67 8.33 10.]	4	0.0005	0	0.0	287.0
26	588	52.87	2.46	4.69	[5. 10.]	2	0.0833	166	77.0	47.0
69	1629	31.90	1.54	5.08	[5. 7.5 10.]	3	0.7653	997	252.0	90.0
70	1028	36.07	3.30	5.21	[5. 6. 7. 8. 9. 10.]	6	0.5625	397	252.0	90.0
51	987	42.17	2.22	5.91	[5.]	1	0.0434	297	130.0	137.0
50	769	55.98	1.06	1.28	[5. 10.]	2	0.0861	438	150.0	98.0
85	503	32.72	4.91	2.05	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0006	0	0.0	415.0
89	863	51.15	1.34	3.10	[5. 10.]	2	1.0276	981	295.0	150.0
56	1969	41.99	1.10	7.79	[5. 6.67 8.33 10.]	4	0.3324	390	203.0	59.0
10	1507	39.81	3.95	5.97	[5. 7.5 10.]	3	0.0557	51	31.0	12.0
42	1521	38.13	3.97	8.68	[5. 7.5 10.]	3	0.2525	452	152.0	66.0
91	600	36.14	2.75	2.27	[5. 7.5 10.]	3	0.0147	73	18.0	462.0
41	687	35.71	1.07	2.08	[5. 6.25 7.5 8.75 10.]	5	0.1404	259	151.0	47.0

Table A.8: LP Flow Model Results of 50 Instances

Parameter Values							Flow Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Array	Max Conc.	CPU Time (s)	Iter. Count	Obj. Value	Makespan
92	522	63.38	3.96	7.67	[5. 6.25 7.5 8.75 10.]	5	0.0350	49	404.0	63.00
87	1993	64.91	2.71	7.10	[5. 6. 7. 8. 9. 10.]	6	0.0322	37	381.0	67.00
91	1237	60.90	3.16	2.29	[5. 6. 7. 8. 9. 10.]	6	0.0007	0	0.0	460.00
86	698	34.78	3.39	4.96	[5.]	1	0.0202	301	209.0	217.00
66	1958	52.02	1.13	9.91	[5. 7.5 10.]	3	0.0173	49	266.7	56.33
63	955	44.98	1.12	1.09	[5. 6. 7. 8. 9. 10.]	6	0.0013	3	9.8	316.25
79	1932	59.54	3.27	1.43	[5. 6.25 7.5 8.75 10.]	5	0.0006	0	0.0	399.00
27	848	52.58	2.29	2.20	[5. 6.25 7.5 8.75 10.]	5	0.0014	9	19.8	117.25
82	1135	34.26	1.34	7.58	[5. 6. 7. 8. 9. 10.]	6	0.0287	49	341.3	61.67
43	1488	54.97	4.13	9.33	[5. 6. 7. 8. 9. 10.]	6	0.0091	42	195.7	32.33
27	511	66.19	4.85	7.68	[5. 7.5 10.]	3	0.0024	44	91.7	35.33
77	1916	58.76	3.89	1.47	[5. 6. 7. 8. 9. 10.]	6	0.0006	0	0.0	388.00
39	832	34.61	3.11	1.46	[5. 10.]	2	0.0005	0	0.0	203.00
61	1591	67.49	4.92	4.70	[5. 10.]	2	0.0026	179	149.0	173.00
74	1517	60.21	2.07	7.17	[5. 10.]	2	0.0162	47	262.5	111.50
42	1435	34.57	1.72	7.39	[5. 10.]	2	0.0053	37	168.0	63.00
84	1880	50.83	4.82	8.52	[5.]	1	0.0155	90	198.0	199.00
30	1918	42.69	3.49	1.92	[5. 6.25 7.5 8.75 10.]	5	0.0007	0	0.0	159.00
20	521	43.89	4.65	8.05	[5. 7.5 10.]	3	0.0017	53	78.3	19.67
18	1912	46.15	4.63	6.63	[5.]	1	0.0011	36	46.0	51.00
18	912	44.09	4.49	7.86	[5.]	1	0.0012	48	35.0	53.00
51	876	43.38	3.94	4.24	[5. 10.]	2	0.0088	265	201.0	67.00
52	1589	58.92	2.00	4.31	[5. 10.]	2	0.0098	58	184.5	75.50
97	890	31.63	4.45	3.11	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0008	0	5.2	453.75
43	1721	53.83	1.02	2.71	[5. 6.25 7.5 8.75 10.]	5	0.0080	36	174.3	45.67
77	1796	61.81	4.89	9.09	[5. 7.5 10.]	3	0.0246	54	315.0	63.00
49	1024	38.45	3.45	5.77	[5. 6. 7. 8. 9. 10.]	6	0.0111	79	188.0	35.00
45	1131	31.96	1.78	9.67	[5. 6. 7. 8. 9. 10.]	6	0.0132	44	171.3	38.67
67	1116	46.36	2.68	8.34	[5. 6. 7. 8. 9. 10.]	6	0.0200	56	291.0	49.00
44	813	52.61	3.37	6.79	[5. 6. 7. 8. 9. 10.]	6	0.0101	50	186.0	31.00
99	1405	56.44	3.75	5.42	[5. 10.]	2	0.0318	370	364.5	121.50
23	1204	35.36	2.17	8.70	[5. 10.]	2	0.0016	36	84.0	35.00
38	549	53.53	2.39	9.33	[5. 6.25 7.5 8.75 10.]	5	0.0069	52	156.3	28.67
82	1553	41.62	1.10	8.80	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0262	58	349.3	56.67
67	1029	48.17	2.90	4.67	[5. 6.25 7.5 8.75 10.]	5	0.0183	132	268.0	57.00
66	1387	39.34	4.84	1.18	[5.]	1	0.0010	0	0.0	304.00
96	1549	53.19	3.85	5.81	[5.]	1	0.0263	231	245.0	256.00
67	791	40.62	2.98	6.89	[5. 6. 7. 8. 9. 10.]	6	0.0304	47	284.7	58.33
75	1053	42.07	2.28	2.61	[5. 6. 7. 8. 9. 10.]	6	0.0282	361	322.0	59.00
76	1639	62.86	4.57	2.57	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0197	0	0.0	376.00
71	843	62.73	1.08	1.61	[5. 6.25 7.5 8.75 10.]	5	0.0390	382	316.0	51.00
77	1709	66.56	3.94	3.84	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0115	339	283.8	84.17
79	1277	36.31	1.44	3.74	[5. 7.5 10.]	3	0.0361	45	328.3	71.67
47	1611	36.94	3.37	7.17	[5. 6.67 8.33 10.]	4	0.0111	44	188.3	38.67
56	1897	50.09	2.65	5.02	[5. 7.5 10.]	3	0.0183	61	223.3	44.67
85	1274	56.73	4.15	3.29	[5. 6.25 7.5 8.75 10.]	5	0.0014	0	0.0	411.00
98	1668	40.36	4.95	5.49	[5. 10.]	2	0.0366	367	343.5	133.50
23	1923	39.06	4.86	1.99	[5. 10.]	2	0.0006	0	0.0	111.00
97	649	63.64	4.46	6.98	[5. 6. 7. 8. 9. 10.]	6	0.0483	155	413.0	76.00
76	823	44.15	4.55	7.97	[5.]	1	0.0160	46	187.0	204.00

Table A.9: LP Flow Model with Valid Inequality Results of 50 Instances

Parameter Values							Flow Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Array	Max Conc.	CPU Time (s)	Iter. Count	Obj. Value	Makespan
71	1461	43.25	4.67	4.15	[5. 6. 7. 8. 9. 10.]	6	0.0022	49	53.17	288.83
62	1461	69.15	3.34	9.31	[5. 6.25 7.5 8.75 10.]	5	0.0271	48	253.00	47.00
92	1809	68.87	1.38	6.08	[5. 6.25 7.5 8.75 10.]	5	0.0602	54	385.33	67.67
17	1315	53.13	1.28	7.17	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0027	56	67.67	14.33
75	705	54.85	1.38	5.79	[5. 6.67 8.33 10.]	4	0.0405	61	322.67	52.33
36	854	31.41	3.15	7.86	[5. 7.5 10.]	3	0.0114	77	150.00	43.00
24	1763	41.62	1.03	1.99	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0035	65	106.00	16.00
61	884	62.10	4.30	8.22	[5. 6.25 7.5 8.75 10.]	5	0.0302	42	268.00	47.00
89	766	49.52	2.04	8.47	[5.]	1	0.0207	71	216.00	232.00
70	1147	39.31	3.52	1.15	[5. 6.25 7.5 8.75 10.]	5	0.0008	0	0.00	324.00
90	1794	32.15	3.54	5.95	[5. 6. 7. 8. 9. 10.]	6	0.0612	112	371.33	84.67
68	994	46.75	4.72	7.80	[5. 7.5 10.]	3	0.0317	72	273.33	67.67
98	1088	38.07	3.90	5.39	[5. 6. 7. 8. 9. 10.]	6	0.0803	478	399.33	91.67
20	718	58.57	4.91	5.66	[5. 7.5 10.]	3	0.0031	250	85.00	25.00
96	746	44.06	4.10	2.40	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0005	0	0.00	496.00
44	645	63.75	1.53	6.48	[5. 7.5 10.]	3	0.0138	113	196.67	39.33
68	949	41.78	4.17	8.55	[5. 7.5 10.]	3	0.0441	45	271.67	91.33
77	1375	59.49	3.03	1.75	[5. 10.]	2	0.0006	0	0.00	409.00
22	1542	42.56	4.96	3.33	[5. 7.5 10.]	3	0.0009	0	0.00	106.00
49	1217	40.56	4.38	8.08	[5. 6.25 7.5 8.75 10.]	5	0.0194	64	190.67	40.33
61	1372	66.62	3.03	8.47	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0279	52	271.33	49.67
88	1450	65.58	3.05	4.88	[5.]	1	0.0242	153	219.00	227.00
48	1177	31.93	3.37	3.69	[5.]	1	0.0054	282	115.00	116.00
62	1113	68.09	3.48	2.04	[5. 6. 7. 8. 9. 10.]	6	0.0010	0	0.00	301.00
92	1894	30.17	3.59	2.59	[5.]	1	0.0006	0	0.00	470.00
27	739	30.89	1.62	9.83	[5. 6.67 8.33 10.]	4	0.0038	37	99.00	25.00
73	701	52.37	3.57	1.49	[5. 6.67 8.33 10.]	4	0.0007	0	0.00	360.00
80	1533	36.17	4.59	6.24	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0505	350	357.33	68.67
50	1925	35.04	4.68	1.00	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0005	0	0.00	267.00
79	725	66.51	2.59	7.99	[5.]	1	0.0171	46	199.00	208.00
15	1294	52.14	3.80	4.76	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0017	94	60.33	11.67
34	1782	35.51	3.84	5.14	[5. 6.67 8.33 10.]	4	0.0118	363	126.33	32.67
74	1853	42.67	1.11	8.59	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0433	52	322.33	62.67
23	1192	32.76	3.10	7.23	[5. 6.25 7.5 8.75 10.]	5	0.0041	47	93.33	26.67
25	1801	66.65	2.79	7.82	[5. 10.]	2	0.0027	107	94.50	35.50
12	667	39.58	2.09	1.04	[5. 6.67 8.33 10.]	4	0.0006	0	0.00	66.00
57	1859	51.18	3.71	6.16	[5. 6.67 8.33 10.]	4	0.0253	58	246.67	56.33
51	1386	45.55	3.92	6.19	[5. 6. 7. 8. 9. 10.]	6	0.0273	344	212.67	50.33
11	784	63.03	1.44	7.33	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0013	25	45.67	9.33
58	610	65.43	1.34	3.45	[5. 6.25 7.5 8.75 10.]	5	0.0288	51	244.67	48.33
16	905	66.03	2.65	9.19	[5. 6.25 7.5 8.75 10.]	5	0.0019	42	71.00	17.00
33	1891	56.18	1.24	9.18	[5. 7.5 10.]	3	0.0051	118	121.67	31.33
22	1262	53.31	4.76	3.08	[5. 6.67 8.33 10.]	4	0.0007	0	0.00	116.00
42	1421	37.58	1.93	2.41	[5. 10.]	2	0.0205	757	156.00	55.00
68	1086	59.14	2.45	2.21	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0026	147	27.25	315.75
100	1748	35.69	4.42	9.36	[5. 6.25 7.5 8.75 10.]	5	0.0660	65	419.33	68.67
93	655	58.08	3.50	7.78	[5. 10.]	2	0.0418	50	345.00	125.00
88	757	60.26	2.00	9.82	[5. 7.5 10.]	3	0.0593	57	390.00	84.00
61	1483	61.53	4.59	8.02	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0277	40	252.00	42.00
60	676	56.80	2.24	5.54	[5.]	1	0.0101	82	139.00	142.00

Table A.10: Totally Unimodular LP Flow Model Results of 50 Instances

Parameter Values							Flow Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Array	Max Conc.	CPU Time (s)	Iter. Count	Obj. Value	Makespan
85	1148	66.63	2.78	4.57	[5.]	1	0.1849	456	206.00	219.00
20	1149	63.83	3.41	6.91	[5. 6.25 7.5 8.75 10.]	5	0.0099	77	74.33	14.67
100	848	50.75	3.89	4.58	[5.]	1	0.2106	538	233.00	246.00
76	1319	50.15	2.70	7.29	[5. 6.67 8.33 10.]	4	0.1726	348	292.33	72.67
62	841	51.22	3.59	7.23	[5. 7.5 10.]	3	0.0846	332	240.00	61.00
95	1305	49.81	3.20	8.02	[5. 6.25 7.5 8.75 10.]	5	0.3416	460	409.33	61.67
94	1773	45.91	4.30	7.28	[5.]	1	0.2666	478	223.00	231.00
62	1481	53.07	3.42	4.88	[5. 6.67 8.33 10.]	4	0.0703	436	261.00	43.00
66	1830	59.73	2.51	1.89	[5. 6.67 8.33 10.]	4	0.0006	0	0.00	350.00
51	1648	47.73	2.92	2.33	[5.]	1	0.0032	0	3.00	257.00
49	1227	31.85	4.02	4.62	[5. 7.5 10.]	3	0.0382	277	191.67	41.33
94	744	46.30	1.08	3.67	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.3197	519	398.33	75.67
98	951	65.35	1.62	1.22	[5. 7.5 10.]	3	0.0014	0	3.00	460.00
20	1395	56.81	4.14	8.33	[5. 7.5 10.]	3	0.0103	86	85.00	18.00
28	1671	62.15	4.21	7.84	[5.]	1	0.0228	143	64.00	71.00
62	1358	46.81	4.35	4.64	[5.]	1	0.0613	311	155.00	166.00
20	1453	43.26	4.19	7.89	[5.]	1	0.0084	118	46.00	54.00
86	1435	68.97	1.24	7.29	[5. 6.67 8.33 10.]	4	0.1910	460	374.33	56.67
45	1512	57.77	4.85	8.79	[5. 6.25 7.5 8.75 10.]	5	0.0368	267	192.33	28.67
24	1954	42.45	4.16	7.39	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0128	138	99.00	16.00
40	773	61.51	4.23	1.67	[5. 6.25 7.5 8.75 10.]	5	0.0008	0	0.00	191.00
94	735	61.06	2.82	4.50	[5. 10.]	2	0.2066	403	372.00	138.00
19	591	45.08	3.88	3.10	[5. 6.67 8.33 10.]	4	0.0012	0	1.75	101.25
32	817	51.05	1.97	6.92	[5. 10.]	2	0.0247	123	102.00	57.00
96	1080	59.95	3.27	8.71	[5. 6.25 7.5 8.75 10.]	5	0.2767	490	397.67	79.33
50	1204	34.09	3.39	6.55	[5. 10.]	2	0.0496	301	187.50	76.50
35	1255	51.70	2.81	9.35	[5.]	1	0.0229	204	73.00	94.00
86	622	30.08	1.66	4.93	[5. 10.]	2	0.1586	352	297.00	112.00
13	1889	62.17	1.09	8.43	[5.]	1	0.0072	56	31.00	32.00
33	1826	68.96	2.40	3.85	[5. 6.25 7.5 8.75 10.]	5	0.0212	240	144.33	23.67
22	619	41.32	2.36	8.34	[5. 10.]	2	0.0115	112	70.50	41.50
91	1765	51.67	1.95	2.33	[5. 7.5 10.]	3	0.1531	381	355.00	82.00
89	1838	65.56	4.78	7.63	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.1814	394	390.67	62.33
83	1680	63.10	3.39	7.32	[5. 6.25 7.5 8.75 10.]	5	0.1662	500	371.67	59.33
94	1630	52.27	4.17	3.31	[5. 6. 7. 8. 9. 10.]	6	0.0040	1	10.25	441.75
58	1619	39.84	4.33	3.71	[5. 6. 7. 8. 9. 10.]	6	0.0054	4	19.25	291.75
12	577	51.42	3.74	1.43	[5. 6. 7. 8. 9. 10.]	6	0.0006	0	0.00	60.00
100	551	32.96	3.45	9.58	[5. 6.25 7.5 8.75 10.]	5	0.2935	541	439.00	75.00
90	653	52.56	3.86	7.10	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.1888	409	376.00	73.00
29	821	62.65	2.37	4.33	[5. 6.67 8.33 10.]	4	0.0149	121	131.25	26.75
56	1679	50.00	2.38	1.76	[5. 6.25 7.5 8.75 10.]	5	0.0014	0	7.00	287.00
25	1944	65.39	1.24	8.03	[5.]	1	0.0116	125	61.00	71.00
43	1028	52.14	1.48	1.94	[5. 10.]	2	0.0315	250	151.50	66.50
57	947	44.49	2.29	7.85	[5. 7.5 10.]	3	0.0789	290	240.00	61.00
23	1964	38.80	3.32	3.89	[5. 6.25 7.5 8.75 10.]	5	0.0110	127	103.25	15.75
21	1843	55.51	4.39	5.17	[5. 10.]	2	0.0093	97	78.00	34.00
89	1402	47.14	2.22	9.48	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.2010	483	351.33	76.67
21	1698	69.36	4.70	1.35	[5. 7.5 10.]	3	0.0006	0	0.00	101.00
48	1293	35.60	1.53	2.36	[5.]	1	0.0349	291	117.00	117.00
90	1765	34.76	4.68	4.31	[5.]	1	0.0116	40	50.00	384.00

Table A.11: LP Flow Model with Optimality Cut Results of 50 Instances

Parameter Values							Flow Model Results			
# of STAs	Circle Radius	Min Dist.	δ Value	β Value	Ω Array	Max Conc.	CPU Time (s)	Iter. Count	Obj. Value	Makespan
87	943	46.76	4.56	9.51	[5. 6.25 7.5 8.75 10.]	5	0.0162	50	61.42	401.58
23	779	60.31	2.07	5.34	[5. 6.25 7.5 8.75 10.]	5	0.0015	34	16.25	96.75
69	793	46.60	2.56	8.10	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0108	50	45.92	313.08
14	799	35.56	4.97	6.44	[5. 6.25 7.5 8.75 10.]	5	0.0014	43	10.50	61.50
20	1470	37.80	2.41	1.89	[5. 6. 7. 8. 9. 10.]	6	0.0008	0	98.00	0.00
98	935	56.92	1.91	1.86	[5. 6. 7. 8. 9. 10.]	6	0.0018	130	270.75	222.25
58	1012	69.04	3.73	6.97	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0087	44	37.50	234.50
37	526	43.25	2.86	4.65	[5. 6.67 8.33 10.]	4	0.0025	59	24.25	162.75
56	781	31.49	2.97	3.77	[5. 10.]	2	0.0098	319	74.00	219.00
70	1064	67.48	1.68	1.62	[5.]	1	0.0010	37	329.00	14.00
57	1012	30.56	2.78	6.96	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0067	51	39.33	242.67
36	1171	31.68	3.50	1.29	[5.]	1	0.0010	0	165.00	0.00
52	1894	46.18	1.59	2.84	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0074	84	33.67	208.33
88	1359	45.79	1.45	3.56	[5. 6.25 7.5 8.75 10.]	5	0.0182	48	58.83	399.17
87	1762	66.51	4.56	8.37	[5. 6.67 8.33 10.]	4	0.0165	49	56.83	367.17
68	1733	39.03	3.30	1.34	[5. 6.25 7.5 8.75 10.]	5	0.0007	0	343.00	0.00
41	1381	30.42	3.84	5.86	[5. 6.67 8.33 10.]	4	0.0026	56	27.00	163.00
85	1301	41.39	3.11	3.50	[5. 6.67 8.33 10.]	4	0.0208	647	55.75	373.25
57	518	38.12	2.98	6.20	[5. 7.5 10.]	3	0.0163	39	48.33	227.67
87	644	42.63	3.80	4.87	[5. 7.5 10.]	3	0.0183	487	71.00	349.00
98	518	57.32	2.92	6.11	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0203	54	65.83	416.17
27	1102	48.46	3.99	7.89	[5. 7.5 10.]	3	0.0015	40	26.67	113.33
21	1780	48.21	4.97	1.31	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0005	0	96.00	0.00
67	1046	58.02	1.31	3.30	[5. 7.5 10.]	3	0.0098	37	59.67	289.33
14	570	31.96	4.46	9.11	[5. 6.67 8.33 10.]	4	0.0011	28	9.00	61.00
78	1530	43.12	1.43	6.83	[5.]	1	0.0127	63	190.00	186.00
99	1862	38.91	2.93	6.22	[5. 6.25 7.5 8.75 10.]	5	0.0231	49	67.08	401.92
68	637	46.44	1.36	3.19	[5. 6.67 8.33 10.]	4	0.0089	51	48.33	293.67
62	980	58.50	2.74	8.89	[5. 6.67 8.33 10.]	4	0.0080	42	39.75	267.25
52	616	31.44	3.93	2.55	[5. 10.]	2	0.0007	0	254.00	0.00
97	799	39.53	2.71	7.71	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0183	47	63.58	417.42
49	1615	41.73	4.01	6.37	[5. 6. 7. 8. 9. 10.]	6	0.0038	68	33.00	215.00
78	1441	68.86	4.93	1.93	[5. 6. 7. 8. 9. 10.]	6	0.0008	0	385.00	0.00
31	1601	43.24	3.15	5.30	[5. 7.5 10.]	3	0.0020	39	27.67	130.33
73	1568	42.19	1.55	6.60	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0116	40	52.67	323.33
51	1165	46.16	1.91	4.21	[5. 6. 7. 8. 9. 10.]	6	0.0037	44	34.00	234.00
54	659	39.73	3.96	4.28	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0076	496	38.75	246.25
34	1762	39.07	2.27	3.65	[5.]	1	0.0022	76	90.00	77.00
81	1795	45.60	4.07	7.04	[5. 6. 7. 8. 9. 10.]	6	0.0122	43	57.50	364.50
65	653	59.40	4.84	3.94	[5. 6.67 8.33 10.]	4	0.0014	16	186.00	122.00
71	1464	68.36	4.57	5.17	[5. 7.5 10.]	3	0.0140	465	62.33	304.67
63	1872	63.10	1.08	7.17	[5. 6.67 8.33 10.]	4	0.0101	51	39.83	256.17
97	570	56.78	3.63	2.13	[5. 6. 7. 8. 9. 10.]	6	0.0007	0	372.67	125.33
53	1478	56.27	2.82	7.39	[5. 10.]	2	0.0046	44	64.00	180.00
72	1219	67.60	3.82	1.14	[5. 6.25 7.5 8.75 10.]	5	0.0007	0	354.00	0.00
79	1731	68.36	2.67	8.80	[5.]	1	0.0158	58	197.00	180.00
57	1783	62.82	3.07	8.10	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0095	41	41.08	265.92
57	1779	39.08	4.20	2.95	[5. 7.5 10.]	3	0.0008	0	278.00	0.00
94	676	30.22	1.94	7.48	[5. 5.83 6.67 7.5 8.33 9.17 10.]	7	0.0168	44	65.58	405.42
36	807	38.92	1.09	2.80	[5. 6.25 7.5 8.75 10.]	5	0.0023	51	24.42	154.58