HEURISTIC ALGORITHMS FOR SCHEDULING IN CENTRALIZED COGNITIVE RADIO NETWORKS

by

Başak Eraslan

B.S., Computer Engineering, Middle East Technical University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2011

HEURISTIC ALGORITHMS FOR SCHEDULING IN CENTRALIZED COGNITIVE RADIO NETWORKS

APPROVED BY:

Assoc. Prof. Fatih Alagöz .....
(Thesis Supervisor)

Assoc. Prof. Tuna Tuğcu .....

Assoc. Prof. Mehmet Akar .....

DATE OF APPROVAL: 25.05.2011

ACKNOWLEDGEMENTS

I would like to thank Didem Gözüpek and my advisor Assoc. Prof. Fatih Alagöz for their valuable assistance and contributions to my thesis.

Special thanks to my husband Gökçen Eraslan for his endless support.

ABSTRACT

HEURISTIC ALGORITHMS FOR SCHEDULING IN CENTRALIZED COGNITIVE RADIO NETWORKS

In this thesis we propose heuristic scheduling algorithms for the centralized, time slotted cognitive radio network (CRN) model in [1]. In the considered CRN scheme, a frequency f and time slot t pair constitutes a resource r and a centralized cognitive base station (CBS) coordinates the assignment of the resources to the secondary users (SUs) possibly with multiple antennas. The proposed algorithms differ in their problem objectives, but they all make sure that none of the primary users in the service area of the CBS is disturbed, no collisions occur among the SUs and each SU gets at least one time slot per scheduling period. For the throughput maximizing scheduling (TMS) problem, an auction theory based algorithm is proposed whereas Best First Resource Assignment (BFRA) and Resource Assignment With Partial Backtracking (RAPB) algorithms are proposed for the maximizing the number of satisfied users (MNSU) problem whose objective is to increase the number of satisfied SUs where an SU is satisfied only if its minimum data rate requirement is met.

ÖZET

MERKEZİ BİLİŞSEL RADYO AĞLARINDA ÇİZELGELEME İÇİN BULUŞSAL ALGORİTMALAR

Bu tezde [1]'de tanıtılmış olan bilişsel radyo ağı (BRA) modelinde çizelgeleme için buluşsal algoritmalar önerilmiştir. Kullanılan BRA planında, merkezi bir baz istasyonu frekans zaman dilimi kaynak çiftlerini birden fazla anteni olabilecek ikincil kullanıcılara (İK) atamaktadır. Önerilen algoritmalar servis alanındaki birincil kullanıcıların rahatsız olmamasını, İK' ler arasında çarpışma yaşanmamasını, her İK' nin her çizelgeleme döneminde en az bir zaman dilimi almasını ve İK' lerin baz istasyonuyla güvenilir bir biçimde iletişim kurmasını garanti etmektedir. İlk algoritmanın amacı ağ üretilen işini arttırmaktır ve bunu yaparken probleme özgü geliştirilmiş açık arttırma yöntemleri kullanılmıştır. Diğer algoritmalarda ise istediği veri hızını elde etmiş memnun İK sayısını arttırmak hedef alınmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xi
LIST OF SYMBOLS xii
LIST OF ABBREVIATIONS xiii
1. INTRODUCTION 1
1.1. Thesis Organization 3
2. BACKGROUND 5
3. THROUGHPUT MAXIMIZING SCHEDULING IN CENTRALIZED COGNITIVE RADIO NETWORKS 8
3.1. Problem Formulation 8
3.2. Proposed Scheduler 10
3.2.1. Auction Theory Based Scheduling Algorithm 10
3.2.2. Computational Complexity Analysis of Our Proposed Algorithm . . 13
3.2.3. Simulation Results 14
4. MAXIMIZING THE NUMBER OF SATISFIED USERS IN CENTRALIZED COGNITIVE RADIO NETWORKS 19
4.1. Problem Formulation 19
4.1.1. Strongly NP-Completeness Proof of the MNSU Problem 22
4.2. Proposed Schedulers 24
4.2.1. Best First Resource Assignment Algorithm 25
4.2.2. Resource Assignment With Partial Backtracking Algorithm 27
4.2.3. Computational Complexity Analysis of the Proposed Algorithms . . 29
4.2.4. Simulation Results 30
5. CONCLUSIONS 49

REFERENCES 50

LIST OF FIGURES

Figure 1.1. DSA with the spectrum hole concept [2]. 1

Figure 1.2. Considered CRN architecture [1]. 2

Figure 3.1. Action flow diagram of auction theory based scheduling algorithm. . . 13

Figure 3.2. Auction algorithm results compared with CPLEX results for varying N and F. 16

Figure 3.3. Auction algorithm results compared with CPLEX results for varying F and a_i. 16

Figure 3.4. Auction algorithm results compared with CPLEX results for varying a_i and F. 17

Figure 4.1. Action flow diagram of Best First Resource Assignment Algorithm. . . 26

Figure 4.2. Action flow diagram of Resource Assignment With Partial Backtracking Algorithm. 29

Figure 4.3. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying N. 34

Figure 4.4. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying F. 34

Figure 4.5. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying a_i. 35

Figure 4.6. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ when the PN_i is different for each SU i. 36

Figure 4.7. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying N. 37

Figure 4.8. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying F. 38

Figure 4.9. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying a_i. 38

Figure 4.10. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ when the PN value differs for each SU. 39

Figure 4.11. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying N. 40

Figure 4.12. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying F. 41

Figure 4.13. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying a_i. 41

Figure 4.14. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ when the PN value differs for each SU. 43

Figure 4.15. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying N. 44

Figure 4.16. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying F. 45

Figure 4.17. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying a_i. 45

Figure 4.18. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying N. 46

Figure 4.19. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying F. 47

Figure 4.20. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying a_i. 47

LIST OF TABLES

Table 4.1. Simulation parameters. 32

LIST OF SYMBOLS

a_i	Number of antennas of secondary user i
D_{ift}	Absolute need weight difference for secondary user i and $FTRP_{if}$
F	Total number of frequencies
N	Total number of secondary users
N_s	Number of satisfied secondary users
T	Number of time slots in a scheduling period
U_{if}	Maximum number of packets that can be transmitted by user i using frequency f
W_{ift}	Weight of $FTRP_{if}$ for secondary user i
Ω_i	Throughput of secondary user i in the current scheduling period

LIST OF ABBREVIATIONS

BFRA	Best first resource assignment
$BFRA_{FFO}$	Best first resource assignment with frequency first ordering
$BFRA_{TFO}$	Best first resource assignment with time slot first ordering
BILP	Binary integer linear programming
CBS	Cognitive base station
CRN	Cognitive radio network
DSA	Dynamic spectrum access
ERQ	Excess resources queue
FIFO	First in first out
$FTRP_{ft}$	Frequency time slot resource pair for frequency f and time slot t
MNSU	Maximum number of satisfied users
PU	Primary user
PN_i	Packet need of secondary user i for the current scheduling period
RAPB	Resource Assignment with partial backtracking
$RAPB_{FFO}$	Resource assignment with partial backtracking with frequency first ordering
$RAPB_{TFO}$	Resource assignment with partial backtracking with time slot first ordering
RQ	Resources queue
SU	Secondary user
TMS	Throughput maximizing scheduling

1. INTRODUCTION

Effective management of the limited spectrum has become a critical issue due to the dramatic increase in the bandwidth usage. However, current fixed spectrum assignment policy leads to suffering of some particular frequency bands from service quality due to heavy load while large portions of the licensed bands are underutilized [2]. In order to overcome this inefficiency in spectrum usage, dynamic spectrum access (DSA) has been introduced in the literature. DSA enables the devices to opportunistically access the licensed frequency bands without causing significant interference to the licensed users. In other words, DSA provides the usage of the temporally unused spectrum portions, which are called *"white spaces"* or *"spectrum holes"*. DSA with the spectrum hole concept is shown in Figure 1.1.

DSA can be realized by cognitive radio (CR) technology, initially proposed by [3].

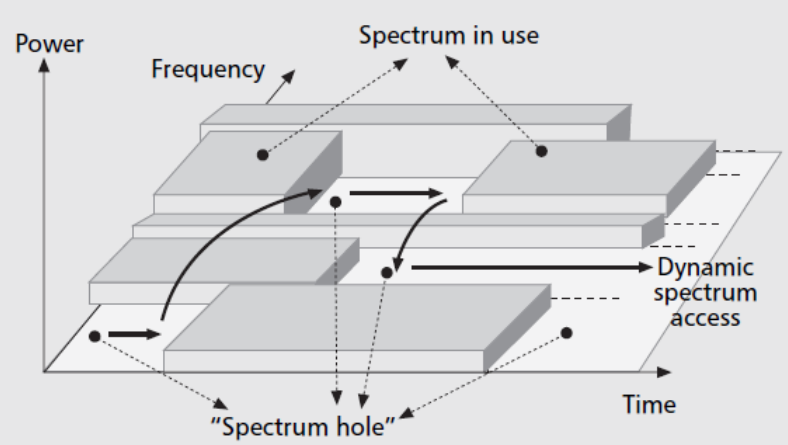


Figure 1.1. DSA with the spectrum hole concept [2].

CRs achieve DSA by being aware of their environment and changing their transmission and reception parameters according to the network state and user demands. More specifically, they sense the spectrum, select the best available channel, coordinate access to this channel with other CRs and vacate the channel when a licensed user is detected [4].

A cognitive radio network (CRN) consists of primary users (PU) and secondary users (SU). PUs have a license to operate in a certain spectrum band and always have privilege over the SUs in spectrum access. If there is a network infrastructure between the PUs, PU activities may be controlled through primary base stations. On the other hand, SUs (cognitive users) are the cognitive radios who opportunistically use the spectrum holes for transmission and must make sure that they never disturb any PU in the network. SUs may have a single-hop connection to a CR base station which helps the SUs to share the available spectrum bands [2].

In this thesis we focus on the time-slotted and centralized CRN architecture provided in [1]. In that architecture a cognitive base station (CBS) controls and guides the cognitive nodes which may have several antennas. That is to say, a scheduler at the CBS determines how many packets and with which frequency each SU will transmit in each time slot [1]. The considered CRN architecture is shown in Figure 1.2. In that CRN

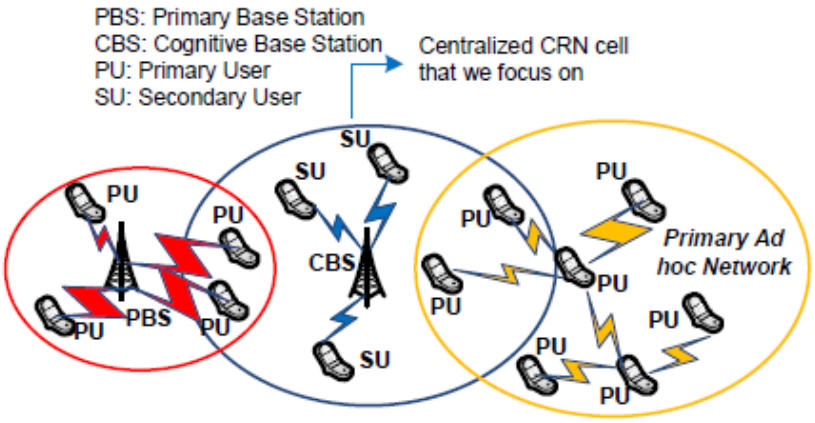


Figure 1.2. Considered CRN architecture [1].

model, an active PU may get disturbed if the interference power generated by the SUs operating on the same frequency is above its maximum tolerable limit. Addressing to this problem, in [1] the binding transmission power and consequently the data rate of each SU for each frequency and time slot is formulated to make sure that the PUs that are actively communicating in that particular frequency are not disturbed.

In this thesis we propose effective scheduling methods for the stated CRN model in which the scheduler at the CBS determines the data rates of the SUs for each frequency and time slot as formulated in [1]. We have two different objectives according to which our scheduling algorithms differ. One of our objectives is to maximize the total network throughput of the CRN. To this end, the authors in [1] have introduced a binary integer linear programming (BILP) model and solved the problem using the optimization software CPLEX [5]. However, this BILP problem is proved to have $O(N^4)$ complexity [1] and a commercial optimization software may not always be available in real life situations like in the premises of the CBS operator. Furthermore, the running times of an optimization software may also be high. We propose an auction based computationally efficient heuristic scheduling algorithm to address this BILP problem.

Our second objective is to maximize the number of the SUs who are satisfied in terms of their data rates. The inspiring idea behind this objective is the case that CRNs may be composed of SUs executing applications with various data rate requirements. Unless their minimum data rate requirements are met, the allocated resources may be totally useless for them. This problem is addressed and formulated in [1]. The problem is proved to be strongly NP-complete by the authors in [1]. In this thesis, we propose two heuristic algorithms with low complexities for a good approximate solution.

1.1. Thesis Organization

The rest of this thesis is organized as follows:

Chapter 2 presents the literature survey about studies on utilizing spectrum via dynamic spectrum access using auction mechanism for CRNs.

Chapter 3 explains the details of the TMS problem in centralized CRNs, presents our auction theory based scheduling algorithm, analyses the complexity of our proposed

algorithm, and compares the simulation results with the solutions obtained by using optimization software CPLEX [5].

Chapter 4 presents the maximizing the number of satisfied users (MNSU) problem in centralized CRNs, explains the problem formulation and the strongly NP-complete structure of the problem. The chapter then presents our proposed schedulers, and discusses their computational complexity analyses. Finally, it presents the simulation results, compares the algorithm results both with each other and with CPLEX results.

Chapter 5 concludes the thesis, summarizes our observations and discusses possible future work.

2. BACKGROUND

In addition to the usage of auction theory in real world auctions like airport time slots, railroad segments and delivery routes, some research studies on utilizing spectrum via dynamic spectrum access using auction mechanism for CRNs have been conducted in [6–14]. In [6], authors introduce a spectrum management policy that simultaneously satisfies the spectrum regulator (i.e., general public interests), the service provider (i.e., operator), licensed users, and unlicensed users. In [7], the authors discuss the problem from these four parties' point of view but they do not guarantee the quality of service of the users. In [8], spectrum sharing among a PU and multiple SUs is considered. Authors formulate the problem of sharing the spectrum as an oligopoly market competition and use a noncooperative game to which Nash equilibrium is considered to be the solution in order to obtain the spectrum allocation for SUs. In [9], the authors model the spectrum allocation in wireless networks with multiple selfish legacy spectrum holders and SUs as multi-stage dynamic games. In [10], only the interaction among SUs is considered via a non-cooperative model. It uses a power/channel allocation scheme that uses a distributed pricing strategy to improve the network's performance. In [11], the problem is formulated as a bandwidth auction in which each SU makes a bid for the amount of spectrum assigned by the PU which takes into account that the assignment does not decrease its own performance. Authors in [11] show that the auction is a non-cooperative game and again Nash equilibrium is considered as the solution of this game. In [12] authors model the problem of spectrum sharing among a few PUs and multiple SUs through second-price and Vickrey type of auctions. In [13] power allocation subject to a constraint on the interference temperature at a measurement point is addressed. However, the interaction between primary users and secondary users is not considered and the bid of one secondary user is unbounded which is unrealistic. The authors in [14] formulate the problem of spectrum sharing among primary and secondary users based on bandwidth auctions in which the primary users are the auctioneers and the secondary users are the bidders. Here the primary users get information form the secondary users and assign the spectrum

bandwidth among them.

The utility theory which has its origins in economics [15], define the utility functions to quantify the benefit of using certain resources. Similarly, utility theory can be used in communications networks to evaluate the degree to which a network satisfies the requirements of the users' applications, rather than in terms of system centric quantities like throughput, packet drop rate and power [16]. Examples of different utility functions for wireless networks are available in [17, 18]. In [17] maintaining both long and short-term fairness among users is concerned and a framework for opportunistic scheduling over multiple wireless channels is developed. Authors in [18] consider improving the wireless network efficiency by exploiting time-varying channel conditions while controlling the level of fairness among users.

In [19–22], the authors proposed scheduling algorithms which primarily objective is to increase the number of satisfied users in wireless networks. In [19] a low complexity scheduling algorithm is proposed for maximizing the number of satisfied users in the wireless networks which have delay constraint traffic. In that study, the satisfied users are defined as the users with a packet delay below a threshold and the main idea of the proposed algorithm is to find the optimum data transmission rate for each user based on their the channel variations and required quality of service. The authors in [20] aim to satisfy the minimum slot requirements of the users for delay sensitive applications and propose a scheduling algorithm based on two-phase combinatorial reverse auction model. In [21] the problem of scheduling transmissions of multiple data users sharing the same wireless channel is considered and the main goal is to satisfy delay or throughput constraints of as many as possible users. In that problem definition, the capacity of the channel varies with time randomly and asynchronously for different users. Even the modified largest weighted delay first algorithm proposed in [21] tends to maximize the number of satisfied users, it does not take the maximum advantage of channel variability. In [22] the authors use the fairness criteria of General Processor Sharing (GPS), which guarantees a number of time slots according to the priority of the users.

Different utility based scheduling problems in cognitive radio networks has been studied in [23–26]. In [23] the objective is to maximize the throughput utility of the secondary users subject to maximum collision constraints with the primary users. In that study the cognitive network consists of static primary users and potentially mobile secondary users. In [24] the problem of resource allocation for multiaccess channel of OFDMA-based cognitive radio networks is studied where the main objective is to balance the efficiency and fairness of wireless resources. Based on their theoretical framework, the authors have applied ellipsoid method and FrankWolfe method to develop algorithms for practical sub-carrier assignment and power allocation. In [25] a resource allocation framework for spectrum underlay in cognitive radio networks using code-division multiple access is proposed. Here, an interference limit corresponding to an interference temperature level is maintained at receiving points of the primary network. In that study, both the interference constraints for primary users and quality of service constraints for secondary users are considered. For the quality of service insurance, signal to interference plus noise ratio of each secondary user is maintained higher than a desired level. Finally in [26] throughput utility maximization in cognitive radio networks with dynamic user arrivals and departures is studied. In that study, the considered cognitive radio network consists of a number of heterogeneous sub-networks which may be power-constrained. That is, the sub-networks are required to operate in such a way that the average total interference received on primary channels are kept below given thresholds.

3. THROUGHPUT MAXIMIZING SCHEDULING IN CENTRALIZED COGNITIVE RADIO NETWORKS

3.1. Problem Formulation

In the time-slotted, centralized CRN model introduced in [1], a CBS determines how many packets and with which frequency each SU in the service area of the CBS will transmit in each time slot. Since the main goal of the CBS is to maximize the total network throughput while assigning the frequency time slot resource pairs (FTRPs) to the SUs, the scheduling problem is named as throughput maximizing scheduling (TMS) problem. In the model, in addition to ensuring that the interference power values perceived by the PUs due to the transmissions of the SUs are within the tolerable limits, it is guaranteed that no collisions occur among the SUs, reliable communication of the SUs with the CBS is maintained, and each SU is assigned at least one FTRP in each scheduling period. In the scheduling model in [1], first the CBS computes the maximum number of packets that the SUs can transmit for every frequency and constructs a matrix called $\mathrm{U} = [U_{if}]$ where U_{if} is the maximum number of packets that can be transmitted by user i using frequency f in a single time slot ensuring that the PU that is actively communicating in that particular f is not disturbed. The number of packets an SU can send by using a frequency is the same for all time slots of a scheduling period. In the second step, the scheduling problem that maximizes the considered CRN network throughput is represented as a BILP with the following formulation [1]:

$$max(\sum_{i=1}^{N}\sum_{f=1}^{F}\sum_{t=1}^{T}\frac{U_{if}X_{ift}}{T}) \tag{3.1}$$

$$s.t.$$

$$\sum_{f=1}^{F}\sum_{t=1}^{T}X_{ift} \geq 1; \forall i \in N \tag{3.2}$$

$$\sum_{i=1}^{N}X_{ift} \leq 1; \forall f \in F, \forall t \in T \tag{3.3}$$

$$\sum_{f=1}^{F}X_{ift} \leq a_i; \forall i \in N, \forall t \in T \tag{3.4}$$

In this formulation, N is the total number of SUs, F is the total number of frequencies, T is the total number of time slots, X_{ift} is a binary decision variable such that $X_{ift} = 1$ if SU i transmits with frequency f in time slot t and 0 otherwise and a_i is the number of transceivers (antennas) of SU i. In the above formulation, (3.2) ensures that each SU is assigned at least one time slot, whereas (3.3) guarantees that at most one SU can transmit in a certain time slot and frequency pair, thereby avoiding collisions among the SUs. Since each transceiver can tune to at most one frequency at a time, (3.4) stands for the fact that an SU i cannot transmit at the same time using frequencies more than the number of its transceivers, (a_i).

In the simulations part of [1], this BILP is solved using the optimization software CPLEX. However, this BILP has $O(N^4F^4T)$ complexity [1] and the running times of an optimization software may be high. In addition, a commercial optimization software may not always be available in real life situations. Taking these facts into account, we have developed a computationally efficient heuristic algorithm for this TMS problem. In the next section, we explain our proposed algorithm, analyse its computational complexity, and present the simulation results.

3.2. Proposed Scheduler

3.2.1. Auction Theory Based Scheduling Algorithm

Auction theory is the branch of game theory dealing with incomplete information, how people behave in these auction markets, and aims to find out their game theoretical properties. Auction theory mainly deals with ascending-bid (open, oral, or English), the descending-bid, the first-price sealed-bid, and the second-price sealed-bid (Vickrey) auction types in different application areas. Besides providing mathematical models of these types by examining the obtained data from previous auctions (including other combinations of these four types), it explores new auction procedures for specific applications [27].

In the ascending auction, the price is successively raised until one bidder remains, winning the object at the final price. Conversely, in the descending auction the auctioneer starts at a very high price, and then lowers the price continuously. In the first-price sealed-bid auction, each bidder independently submits a single bid without seeing others' bids, and the object is sold to the bidder who makes the highest bid. Different from the first-price, in the second-price sealed-bid auction, the winning bidder pays the second highest bidders' bid [27].

In this thesis, we propose a first-price sealed-bid auction based scheduling method to address the TMS problem formulated in (3.1) - (3.4). Since our optimization problem has unique constraints such as ensuring that each SU is assigned at least one time slot and each SU possibly has multiple antennas, the existing auction mechanisms in the literature fail to address these types of constraints. Therefore, our proposed heuristic algorithm uses a novel auction mechanism tailored for our optimization problem.

In the optimization problem in (3.1) - (3.4) a frequency f and time slot t pair constitutes a resource r which we also refer to as an $FTRP$. Notice here that if we ignore constraint (3.2) and (3.4) in this problem, the optimal solution is achieved when each

resource r is assigned to the SU i that has the maximum U_{if} value for the frequency f of this resource r. In other words, all the time slots of a frequency f are assigned to SU i where $i = arg\,max_i(U_{if})$. In the rest of this thesis we use the term *"starving user"* for each SU that has not yet been assigned any FTRP during any stage of the algorithm. Since the optimization problem aims to assign at least one resource to each SU, our auction algorithm avoids having starving users at the end of the algorithm. However, note that the problem has a feasible solution if and only if $N \leq FT$ and the starving users can be avoided only if this situation holds.

Our proposed algorithm is explained through Step 1 to Step 4 below. Here, n_i denotes the number of frequencies assigned to SU i at the end of Step 1. Recall that *starving user* stands for SUs to whom no FTRP is assigned yet.

STEP 1: For each frequency, find the SU who transmits the maximum number of packets using that frequency and assign the frequency to that SU for all time slots of the scheduling period. In other words, assign frequency f to SU i where $i = arg\,max_i(U_{if})$.

STEP 2: If all SUs are assigned at least one FTRP and $n_i \leq a_i, \forall i$, return. Otherwise each SU that is assigned more than one frequency sorts its frequencies according to their U_{if} values. If any SU i has n_i greater than a_i, go to STEP 3, else go to STEP 4.

STEP 3: If there exists any starving user go to 3a, else go to 3b.

3a: Any SU i whose $n_i \geq a_i$ auctions all time slots of $n_i - a_i$ number of its frequency bands which have the smallest U_{if} values. Starving users bid to all available time slots of the auctioned frequency with which they send maximum number of packets. That is to say, a starving SU i bids to all unassigned timeslots of the frequency f for whom U_{if} value is the greatest one among all simultaneously auctioned frequencies. As soon as a starving user gets an FTRP, it does not participate in the subsequent auctions anymore. The auctions continue until either all of the starving users get at least one FTRP or no

FTRP remains. At the end of the auctioning procedure if all resources are assigned and there still exists some starving users go to Step 4. Otherwise if there still exists FTRPs that are not assigned to any SU, they are auctioned to the bidders who have available antennas and sold to the one with the maximum bid.

3b: Any SU i whose $n_i \geq a_i$ takes a_i of its frequency bands which have the largest U_{if} values. The FTRPs which belong to the remaining $n_i - a_i$ frequencies are auctioned to the bidders who have available antenna and each is assigned to the SU with the maximum bid.

STEP 4: Any SU i who is assigned more than one frequency auctions T time slots of $n_i - 1$ number of its frequencies with the smallest U_{if} values until either no starving SU remains or the auctioned FTRPs run out. When all SUs have at most a single frequency, if any starving SU exists, they auction $T - 1$ time slots of their remaining frequencies.

U_{if} values of any SU i are independent from other SUs' values, and each SU knows only its own U_{if}, namely bid values. Since the bids do not affect each other, the auctions are held in a sealed bid fashion. Moreover, in order to maximize the network throughput, first-price auctions are used in the algorithm steps and an auctioned frequency is assigned to an SU whose bid is greater than all other bids.

The FTRPs which belong to the same time slot are auctioned simultaneously and we can name a group of such auctions as a *"simultaneous auction set"*. In a simultaneous auction set, the result of an auction does not impact another auction in the same set. Hence, if we had used multi-bid auctions, in other words, if users could bid for more than one FTRP for the same time slot, some starving users might end up with receiving more than one FTRP while some other starving users continue to starve. This situation would obligate additional number of auctions in order to guarantee that the algorithm terminates and hence increase the complexity and running time of the algorithm. Therefore, having the bidders to bid only to one FTRP for each time slot renders our proposed algorithm

simpler by avoiding additional operations to ensure the termination of the algorithm.

The action flow diagram of the algorithm is presented in Figure 3.1.

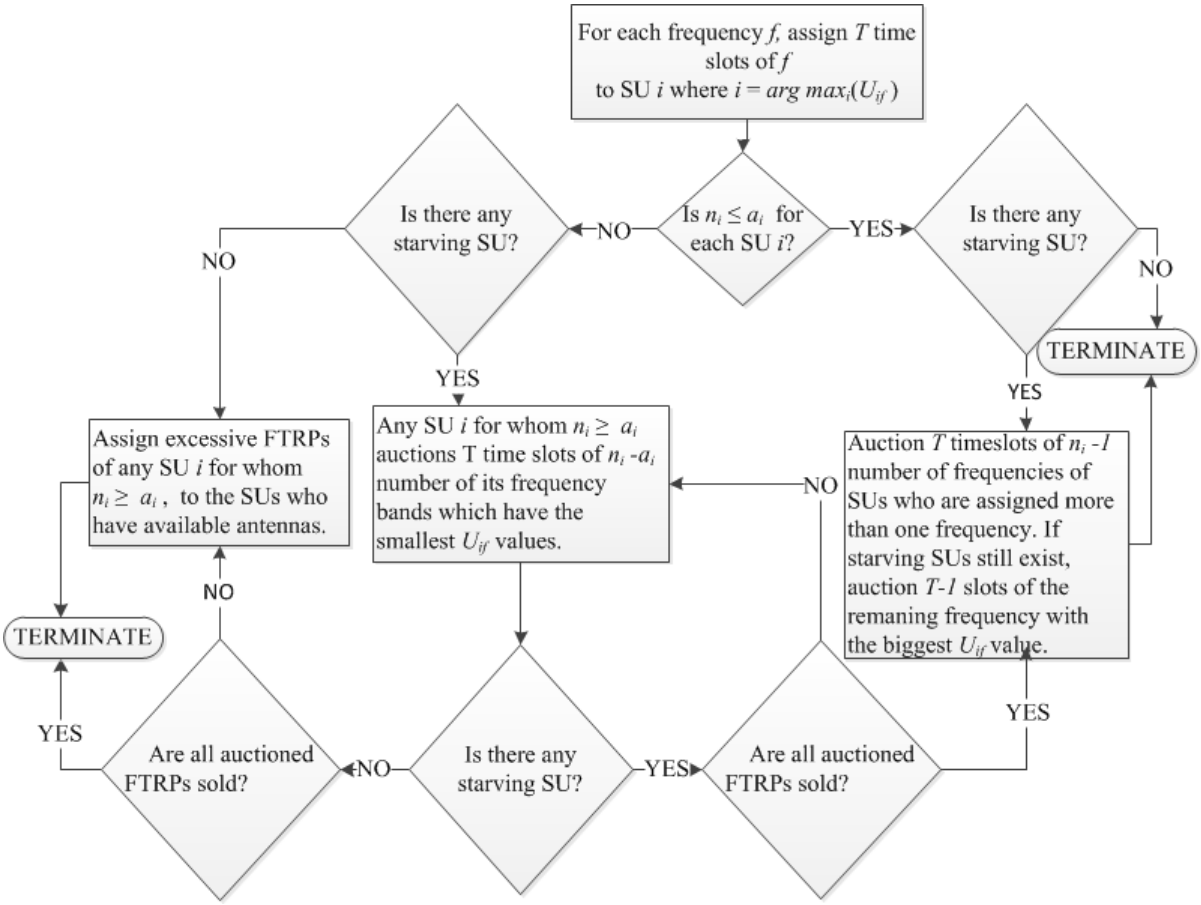


Figure 3.1. Action flow diagram of auction theory based scheduling algorithm.

3.2.2. Computational Complexity Analysis of Our Proposed Algorithm

We can analyse the complexity of the worst case and best case scenarios of the algorithm described in the preceding section.

Worst Case Scenario: The worst case scenario occurs when at the end of Step 1 all of the F frequencies are assigned to one user, remaining $N-1$ users starve and the

starving users always bid to the same FTRP in all of the auctions.

The order of the greedy assignment in Step 1 is $O(FN)$ because for each frequency f, its largest U_{if} value is found. In Step 2, the U_{if} values of F number of frequencies which are assigned to the same user are sorted and the complexity of this operation is $O(Flog(F))$. Since all starving SUs always bid to the same FTRP, in each simultaneous auction set only one starving user gets one FTRP. Thus, $N-1$ number of simultaneous auction sets must be held to assign an FTRP to each starving user. A simultaneous auction set takes $O((N-1)log(N-1))$ time because it involves sorting the bids of $N-1$ starving SUs. Thus $(O(FN)) + O(Flog(F)) + (N-1) \times O((N-1)log(N-1))$, which is roughly equal to $O((FN) + N^2log(N))$, is the total time complexity of the worst case scenario.

Best Case Scenario: Best case scenario occurs if at the end of initial greedy assignment in Step 1, all SUs are assigned at least one FTRP while (3.4) is not violated. In that case time complexity becomes $O(FN)$.

3.2.3. Simulation Results

For the simulations we use the same CRN environment explained in [1], in which a CRN cell has 600 meters radius and each scheduling period consists of 10 time slots, where each time slot is 100 ms long. U_{if} values for 5000 scheduling periods in each set of simulations are obtained by taking PU velocities, SU velocities, and number of PUs into account [1]. We consider a random way point mobility model for SUs and PUs as in [1]. In our experiments, we set both PU and SU velocities to 13 m/s. We vary the number of SUs (N) between 5 and 30. In each scheduling period, all SUs led by the same CBS have the same number of antennas and for some input sets this number varies between 1 and 5. Finally, since the TMS problem has a feasible solution if and only if $N \leq (F \times T)$, we have at least 3 available frequencies in each scheduling period .

In the Figures 3.2, 3.3, and 3.4 we present the simulation results of our algorithm in comparison to the CPLEX results obtained in [1]. In these figures, each resulting throughput is the average throughput value of 5000 scheduling periods with the same N, F and a_i parameters.

In Figure 3.2, we present the resulting total throughput values for $F = 3$, $F = 15$, and $F = 30$ when the number of antennas $a_i = 3$, $\forall i$. F_{CPLEX} denotes the number of frequencies used in the CPLEX simulations, whereas F_{AUC} denotes the number of frequencies used in our proposed auction based algorithm. In this figure, it is obviously seen that our algorithm results are very close to the corresponding CPLEX results for each simulation set. Note that there is a significant increase in the throughput in the case of $F = 30$ between $N = 5$ and $N = 10$ because when $N = 5$, since $a_i = 3$ only 15 of the frequencies are assigned to the SUs, the remaining frequencies stay unoccupied.

In Figure 3.3 the throughput values are compared for $a_i = 1$, $a_i = 3$, $a_i = 5$ where N is 15 and F varies between 3 and 30. Here, $a_{i_{CPLEX}}$ and $a_{i_{AUC}}$ denote the number of antennas of each SU i, in the CPLEX simulations and in our proposed algorithm simulations respectively. This figure illustrates that the throughput results of our proposed auction theory based algorithm are close to the CPLEX results when the SUs have varying number of antennas.

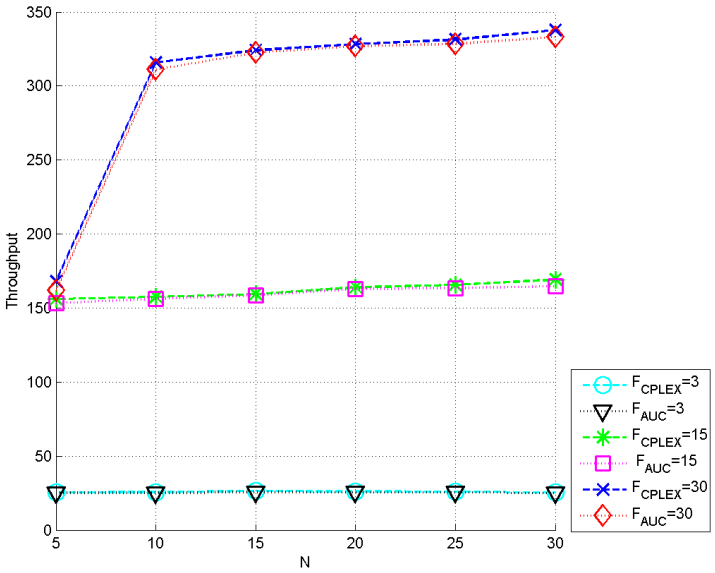


Figure 3.2. Auction algorithm results compared with CPLEX results for varying N and F.

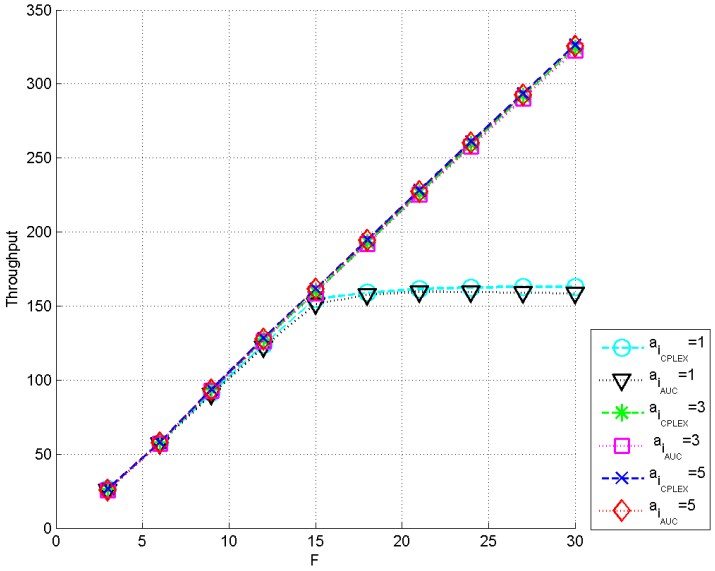


Figure 3.3. Auction algorithm results compared with CPLEX results for varying F and a_i.

Notice that in Figure 3.3, the throughput values increase as F increases for $a_i = 1$, $a_i = 3$, $a_i = 5$ while F is between 3 and 15, because in this interval $F \leq (N \times a_i)$ and all of the available frequencies can be assigned to the SUs. However, we observe that the throughput values do not increase much despite of the increase in F for $F > 15$ when $a_i = 1$. Since each transceiver can tune to at most one frequency at a time and an SU cannot transmit at the same time using frequencies more than the number of its transceivers, in these simulation sets $F - (N \times a_i)$ number of frequencies stay unoccupied for T time slots. Thus the increase in F does not affect the resulting throughput values much when $a_i = 1$ for $F > 15$.

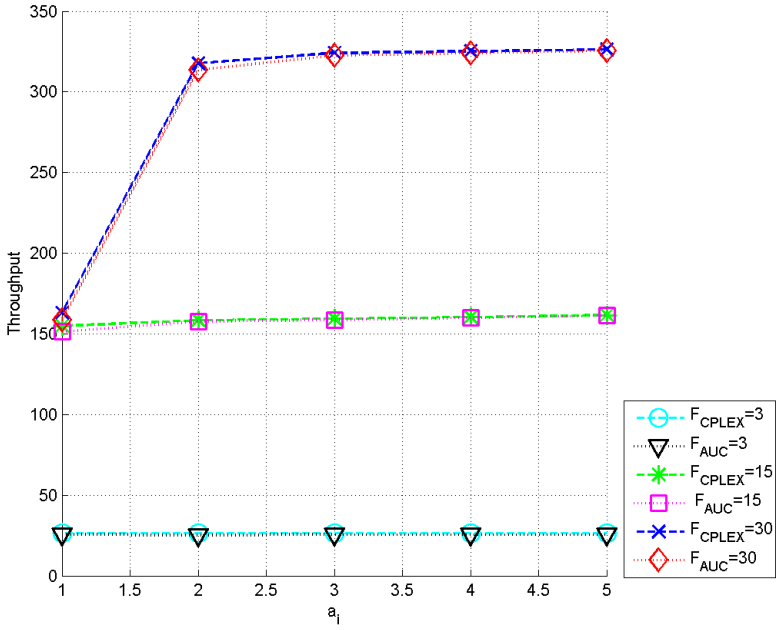


Figure 3.4. Auction algorithm results compared with CPLEX results for varying a_i and F.

Finally in Figure 3.4 we compare the throughput results of our proposed algorithm with CPLEX results for $F = 3$, $F = 15$, $F = 30$ where $N = 15$ and a_i varies between 1 and 5. Again F_{CPLEX} and F_{AUC} denote the number of frequencies used in the CPLEX simulations and in our proposed algorithm simulations respectively. This figure illustrates that the throughput values do not change much when a_i varies for $F = 3$, $F = 15$ both

in CPLEX and our proposed algorithm results. However, there is a significant increase in the throughput when each SU has 2 antennas for $F = 30$. This is because when $a_i = 1$ only 15 of these frequencies could be assigned to 15 SUs. However when a_i becomes 2, all of the available 30 frequencies are assigned to the SUs.

The simulation results prove that our proposed auction theory based algorithm's throughput results are very close to the optimal CPLEX results for varying N, F and a_i parameters. Since our proposed algorithm has $O(FN + N^2 log(N))$ worst case computational complexity, it is efficient to use this algorithm as a throughput optimal CBS scheduler in the considered centralized, time slotted CRN model [29].

4. MAXIMIZING THE NUMBER OF SATISFIED USERS IN CENTRALIZED COGNITIVE RADIO NETWORKS

4.1. Problem Formulation

In a CRN, SUs may be executing applications which have various data rate requirements. For instance, an SU who is having a real-time video conference may require at least 1 Mbps to communicate effectively while an SU with an e-mail application would be satisfied with much slower data rate. An SU having a video conference may be unsatisfied unless its minimum throughput requirement is met and the allocated resources may be totally useless for it. In other words, providing this SU with a data rate of 10 Kbs or 100 Kbs would not make a difference since both of these data rates are insufficient for this SU. Consequently, it is wiser not to allocate any resources to this SU if its minimum data rate requirement would not be met and allocate the resources instead to SUs whose data rate requirements can be met.

Furthermore, allocation of data rate higher than the minimum requirement of an SU may not make that SU happier. For instance, providing a data rate more than 100 Mbps to an SU whose minimum data rate requirement is 100 Mbps may not give him additional advantage, that is 101 Mbps and 300 Mbps may be equally good for this SU. Hence, instead of allocating these excess resources to this SU, assigning them to other SUs who actually need these resources would make more number of SUs satisfied.

When we want to model a scheduler for the CBS operator whose major goal is to maximize the number of SUs who are satisfied in terms of throughput, we should consider the facts explained in the preceding two paragraphs. In [1] this maximizing the number of satisfied users (MNSU) problem is discussed for the time slotted, centralized CRN and a formulation of the problem with additional constraints is provided. Accordingly, in this thesis we propose two computationally efficient scheduling algorithms for

the MNSU problem presented in [1].

The centralized, time slotted CRN considered in [1] has the same architecture with the CRN we used in the TMS problem in Section 3. That is to say, in the considered CRN architecture, first of all the CBS computes the maximum number of packets that the SUs can transmit for each frequency and constructs the matrix U = $[U_{if}]$ where U_{if} is the maximum number of packets that can be transmitted by user i using frequency f in one time slot without disturbing the active licensed user of the frequency band. Subsequently, the scheduler at the CBS decides with which frequency each SU will transmit in each time slot during the entire scheduling period. At this point, the scheduling problem of the CBS whose goal is to maximize the number of SUs who are satisfied in terms of throughput is formulated in [1] as follows:

$$max(\sum_{i=1}^{N} g(\Omega_i)) \tag{4.1}$$

$$s.t.$$

$$\Omega_i = \sum_{f=1}^{F}\sum_{t=1}^{T} \frac{U_{if}X_{ift}}{T} \tag{4.2}$$

$$\sum_{f=1}^{F}\sum_{t=1}^{T} X_{ift} \geq 1; \forall i \in N \tag{4.3}$$

$$\sum_{i=1}^{N} X_{ift} \leq 1; \forall f \in F, \forall t \in T \tag{4.4}$$

$$\sum_{f=1}^{F} X_{ift} \leq a_i; \forall i \in N, \forall t \in T \tag{4.5}$$

In this formula Ω_i denotes the throughput of SU i in this scheduling period. Each SU i has a utility function $g(\Omega_i)$ which takes the throughput of SU i as input and produces either 1 or 0 representing whether this SU is satisfied with its throughput in this period

or not. This utility function can be formulated as follows [1]:

$$g(\Omega_i) = \begin{cases} 1 & \text{if } \Omega_i \geq \Omega_i^{min} \\ 0 & \text{otherwise} \end{cases}$$

Ω_i^{min} represents the minimum throughput requirement of SU i in this specific scheduling period. If this requirement is met by the CBS by assigning it proper FTRPs, then the SU is satisfied and the output of the utility function becomes 1. Otherwise, if the user's throughput is below Ω_i^{min}, the SU is not satisfied and the result of the utility function is 0.

Constraints (4.3), (4.4) and (4.5) are the same constraints as in the throughput maximization scheduling problem we discussed in the previous chapter. Since we do not want any SU to get totally unconnected, 4.3 ensures that each SU is assigned at least one time slot in each scheduling period. In order to prevent collision among SUs and to keep the aggregate interference experienced at the PUs below their maximum limit, 4.4 ensures that at most one SU is assigned to a specific frequency and time slot pair. Finally 4.5 represents the fact that an SU i cannot transmit at the same time using frequencies more than the number of its transceivers, a_i.

This MNSU problem is a strongly NP-complete problem [1] and a detailed explanation of the NP-completeness proof is provided in the next section. However, we can convert the problem into a BILP to obtain approximate CPLEX solutions. Since the utility function $g(\Omega_i)$ produces either 1 or 0, we can introduce a new variable s_i which denotes whether the SU is satisfied in terms of throughput or not. Therefore, the solution to the problem formulated in (4.1)-(4.5) becomes the same as the solution to the following BILP:

$$max(\sum_{i=1}^{N} s_i) \tag{4.6}$$

$$s.t.$$

$$\sum_{f=1}^{F}\sum_{t=1}^{T}\frac{U_{if}X_{ift}}{T} \geq \Omega_i^{min} \times s_i; \forall i \in N \tag{4.7}$$

$$\sum_{f=1}^{F}\sum_{t=1}^{T} X_{ift} \geq 1; \forall i \in N \tag{4.8}$$

$$\sum_{i=1}^{N} X_{ift} \leq 1; \forall f \in F, \forall t \in T \tag{4.9}$$

$$\sum_{f=1}^{F} X_{ift} \leq a_i; \forall i \in N, \forall t \in T \tag{4.10}$$

Constraint (4.7) models the behaviour of the function $g(\Omega_i)$. More precisely, if an SU i is satisfied, then its throughput is greater than or equal to its minimum throughput requirement, namely Ω_i^{min}. The rest of the constraints are the same as the (4.3)-(4.5).

4.1.1. Strongly NP-Completeness Proof of the MNSU Problem

MNSU problem is strongly NP-complete [1] because its sub-problem corresponds to the bin covering problem, which is also a strongly NP-complete problem [28].

Bin Covering Problem : Suppose we have n items with weights $w_1, ..., w_n$ and an infinite number of identical containers with quota q. The bin covering problem is to pack these n items into containers such that the number of containers that contain sets of items whose sums are at least q is maximized. That is, the goal is to distribute the items among as many containers as possible, given that the containers have a specified quota that must be satisfied. Note that, the total weight of the items placed in a container can be greater than q [28].

More formally, bin covering problem is formulated as an integer program [28]:

$$max(\sum_{i=1}^{n} y_i) \tag{4.11}$$

$$s.t.$$

$$\sum_{j=1}^{n} w_j c_{ij} \geq q y_i; \qquad \forall i = 1, ..., n \tag{4.12}$$

$$\sum_{i=1}^{n} c_{ij} \leq 1; \qquad \forall j = 1, ..., n \tag{4.13}$$

$$c_{ij} \in \{0, 1\} \ ; \qquad \forall i = 1, ..., n, j = 1, ..., n \tag{4.14}$$

$$y_i \in \{0, 1\} \ ; \qquad \forall i = 1, ..., n \tag{4.15}$$

where y_i represents whether the quota on the i^{th} bin is satisfied ($y_i = 1$) or not ($y_i = 0$), and $c_{ij} = 1$ if item j is assigned to bin i, and 0 otherwise. Constraint (4.12) ensures that the quota is satisfied for each bin that is instantiated, and constraint 4.13 ensures that items are assigned to at most one bin.

Bin covering problem which is a strongly NP-complete problem [28], is a natural model for resource or task allocation among multiple agents where the goal is to maximize the number of agents who achieve some quota. In that respect, MNSU problem behaves like the bin covering problem and we can prove that bin covering problem is a sub-problem of MNSU as follows: Let us assume $a_i \geq F$ for each SU i , thus constraint (4.10) is satisfied automatically. Moreover let $F \times T > N$ and Ω_i^{min} be the same for each SU i and be greater than all U_{if} values. That is, the minimum throughput requirements of the SUs are equal and none of the SUs is satisfied even if constraint (4.8) is already satisfied. Therefore, this special case of the MNSU problem corresponds to the bin covering problem where SUs are the containers with quota Ω_i^{min} and $FT - N$ number of FTRPs are items to be packed which have weights U_{if}.

4.2. Proposed Schedulers

Even though MNSU problem is strongly NP-complete, approximate solutions to this BILP can be derived by using CPLEX optimization software. However, a commercial optimization software may not always be available in real life situations like in the premises of the CBS operator. Besides, the running times of an optimization software may be so high that it may be unrealistic to use it in a CBS scheduler.

We propose two computationally efficient heuristic algorithms for the MNSU problem: Best First Resource Assignment (BFRA) and Resource Assignment With Partial Backtracking (RAPB) algorithms. In both of these algorithms each frequency f and timeslot t resource pair constitutes an $FTRP_{ft}$. Each $FTRP_{ft}$ has a weight W_{ift} for each SU i which is equal to the number of packets that SU i can transmit using frequency f in time slot t. Because the number of packets an SU can send by using a frequency is the same for all time slots of a scheduling period, W_{ift} is equal to the corresponding U_{if} value.

The strategy followed for the assignment of these resources to the SUs determine the total number of satisfied users (N_s). There are two important factors which must be concerned while building up this strategy: the instant packet needs of the SUs and the weights of the FTRPs according to different SUs.

At the beginning of the scheduling algorithm the packet need (PN_i) of each SU i is equal to $\Omega_i^{min} \times T$ which is the minimum number of packets SU i must transmit in the current scheduling period in order to be satisfied. However, this value changes dynamically throughout the algorithm as the FTRPs are assigned to the SUs. At any instant, PN_i is equal to the difference between $\Omega_i^{min} \times T$ and the total weight of the FTRPs assigned to SU i so far in the current scheduling period. It is wiser to assign the FTRPs primarily to the SUs who have less packet need because this way more number of SUs can be satisfied with the same amount of resource. For instance, instead of assigning all FTRPs to one SU who needs 400 packets per schedule, we can distribute these resources

to five SUs each of which needs only 80 packets and make five SUs satisfied instead of one.

The weight of each FTRP may differ according to each SU because for any $FTRP_{ft}$, W_{ift} is equal to the number of packets SU i transmits using frequency f in one time slot. In order to use the available frequencies more efficiently, it is wiser to assign a resource to the SU for whom the weight of the FTRP is the biggest.

Both BFRA and RAPB consider these two factors and search for an available SU i for whom $|PN_i - W_{ift}|$ is minimum for $FTRP_{ft}$. This value is called as *"absolute need weight difference"* and shown as D_{ift} for SU i and $FTRP_{ft}$.

BFRA and RAPB are explained in the following sections. Recall that F is the number of frequencies, T is the number of time slots in a scheduling period, N is the number of SUs in the service area of the CBS scheduler, PN_i is the current packet need of the SU i and D_{ift} is absolute weight difference of SU i with respect to $FTRP_{ft}$.

4.2.1. Best First Resource Assignment Algorithm

BFRA is a greedy heuristic algorithm which has very low computational complexity. The algorithm is explained through Step 1 to Step 3 and the action flow diagram of BFRA is provided in Figure 4.1.

STEP 1: Generate $F \times T$ number of FTRPs. For each SU find and assign an FTRP whose weight is maximum among available FTRPs. Put the remaining $(F \times T) - N$ number of FTRPs in a First In First Out (FIFO) resources queue (RQ).

STEP 2: Pop the head FTRP of the queue. Let us call it $FTRP_{ft}$. Search for the unsatisfied SUs who have an available antenna for the time slot t. If there is no such SU, put the FTRP to the excess resources queue (ERQ). Otherwise find the SU i for whom D_{ift} is minimum and assign $FTRP_{ft}$ to SU i. If either the RQ is empty or all SUs are

satisfied go to Step 3, else repeat Step 2.

STEP 3: For each FTRP in the ERQ, look for a satisfied SU for whom the weight of the FTRP is maximum among SUs that have available antenna for the time slot of the FTRP.

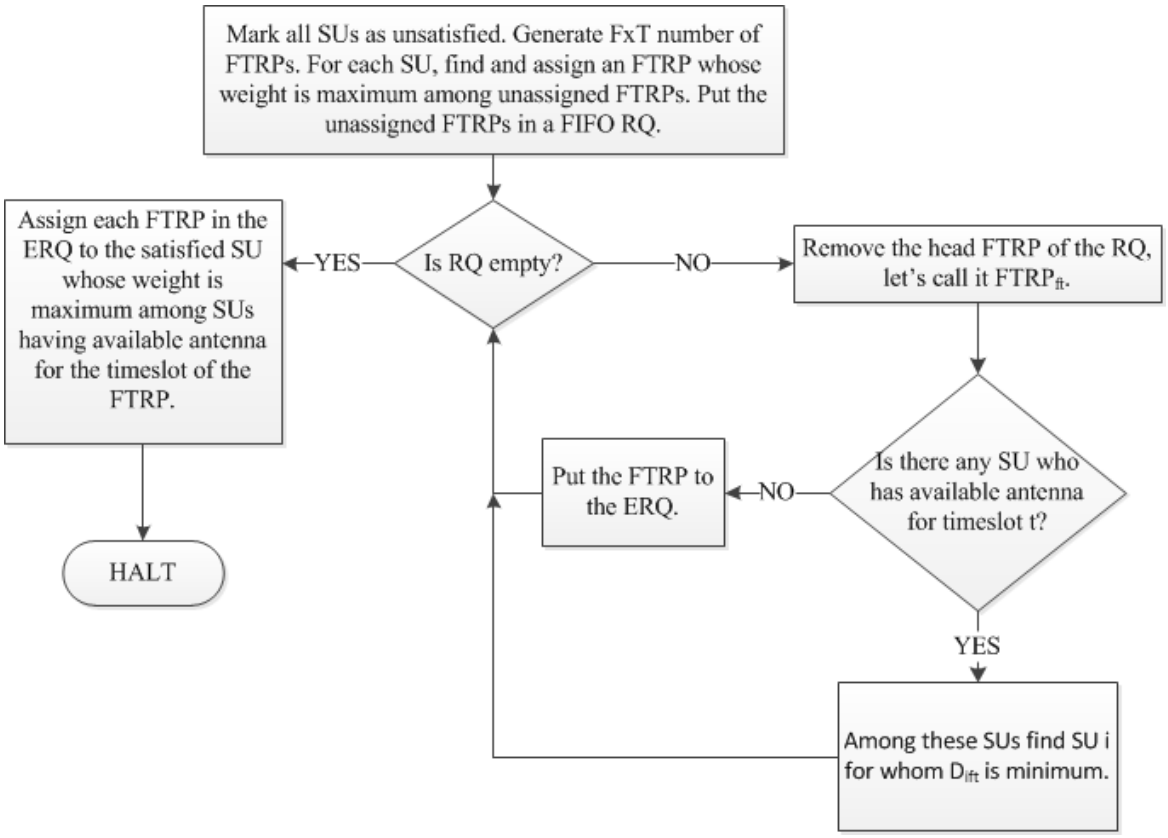


Figure 4.1. Action flow diagram of Best First Resource Assignment Algorithm.

Notice that although the major goal of BFRA is to maximize the N_s, it also aims to increase the total throughput of the SUs in the CRN by assigning the excess FTRPs (i.e. FTRPs which could not be assigned to unsatisfied SUs due to the antenna constraints of the SUs) to the already satisfied SUs at Step 3.

The order that the FTRPs are put into the RQ at Step 1 may affect the performance

of BFRA. Each FTRP assignment changes the PN value of an SU and occupies an antenna for the time slot it belongs to, therefore the assignment outcomes of anterior FTRPs may affect the assignments of posterior FTRPs. We put the FTRPs into the RQ in two different ways and name the algorithm with two sub versions: BFRA with frequency first ordering ($BFRA_{FFO}$), BFRA with time slot first ordering ($BFRA_{TFO}$). In $BFRA_{FFO}$ the FTRPs which belong to the same frequency are put consecutively into the RQ, namely as $FTRP_{11}, FTRP_{12}..., FTRP_{1T}, ..., FTRP_{F1}, FTRP_{F2}..., FTRP_{FT}$. On the other hand, in $BFRA_{TFO}$ the FTRPs which belong to the same time slot are put consecutively into the RQ, such as $FTRP_{11}, FTRP_{21}, ..., FTRP_{F1}, ..., FTRP_{1T}, FTRP_{2T}, ..., FTRP_{FT}$.

4.2.2. Resource Assignment With Partial Backtracking Algorithm

BFRA is a greedy algorithm with very low computational complexity. However its performance is highly dependent on the order of the FTRPs in the RQ. That is, although the posterior FTRPs in the RQ have bigger weights, the FTRPs which are near the front of the queue are more privileged in terms of being assigned to an SU. This is because in BFRA once an FTRP is assigned to an SU, it can not be replaced with another FTRP.

RAPB compensates this drawback of BFRA by replacing an already assigned FTRP with a newly popped FTRP whose weight is bigger for that particular SU. RAPB is explained through Step 1- Step 6. Notice that Step 1 and and Step 6 are the same as BFRA algorithm.

STEP 1: Generate $F \times T$ number of FTRPs. For each SU, find and assign an FTRP whose weight is maximum among available FTRPs. Put the remaining $(F \times T) - N$ number of FTRPs in a First In First Out (FIFO) resources queue (RQ).

STEP 2: If RQ is empty go to Step 6. Otherwise, pop the head FTRP of the queue, let us call it $FTRP_{ft}$. Mark all unsatisfied SUs as unchecked and go to Step 3.

STEP 3: Find the SU i for whom D_{ift} is minimum among all unsatisfied, unchecked SUs. If that SU has an available antenna for the time slot to which the FTRP belongs, namely time slot t, then assign the FTRP to it, mark this SU as satisfied if $PN_i \geq 0$, and go to Step 2. Otherwise go to Step 4.

STEP 4: Compare the weight of the FTRP, i.e. W_{ift}, with the weights of the already assigned FTRPs that occupy an antenna for time slot t. If W_{ift} is bigger than their smallest one, then take the FTRP with the smallest weight from the SU i, put it at the end of the RQ and instead assign $FTRP_{ft}$ to the SU i. Mark this SU as satisfied if $PN_i \geq 0$. Go to Step 2. On the contrary, if W_{ift} is the smallest one, mark that SU as checked and continue with Step 5.

STEP 5: If there are any unchecked users among unsatisfied SUs go to Step 3, else put the FTRP to the excess resources queue (ERQ), and go to Step 2.

STEP 6: If there exist some FTRPs in the ERQ, for each FTRP in the ERQ, look for a satisfied SU for whom the weight of the FTRP is maximum among SUs that have available antenna for the time slot of the FTRP.

Like BFRA, the performance of RAPB may be affected by the order of the FTRPs in the RQ because prior FTRP assignments lead the algorithm flow by changing current packet needs of the SUs. Therefore, we again put the FTRPs into the RQ in frequency first ordering and time slot first ordering, just as it was in BFRA. These subversions are named as $RAPB_{FFO}$ and $RAPB_{TFO}$.

The action flow diagram of RAPB is presented in Figure 4.2:

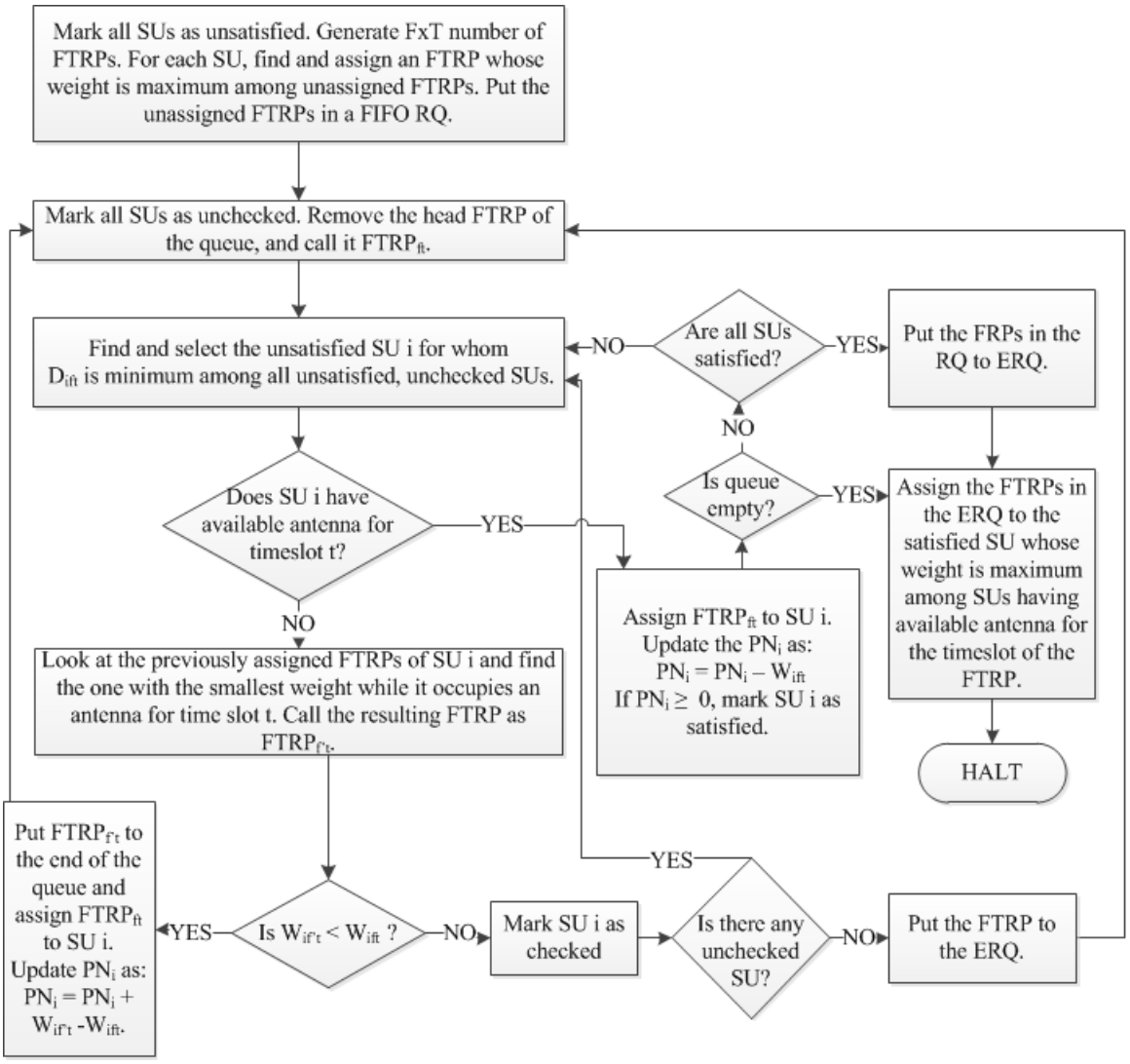


Figure 4.2. Action flow diagram of Resource Assignment With Partial Backtracking Algorithm.

4.2.3. Computational Complexity Analysis of the Proposed Algorithms

Worst-case performance analysis of the BFRA algorithm: In the worst case scenario, the operation of assigning an FTRP to each SU at Step 1 is $O(NF)$ because for each SU we find the frequency with which the SU sends maximum number of packets. Then for each $FTRP_{ft}$ in the RQ, we search for the SU i for whom D_{ift} is minimum among the SUs that have an antenna for time slot t. This search is $O(N)$ for each FTRP since it is sufficient to scan the SUs list once to learn both of these attributes. As there are $FT - N$

number of FTRPs in the RQ, Step 2 is $O((FT - N) \times N)$. Finally, if we assume that each SU has A number of antennas, the complexity of Step 3 is $O((FT - NA) \times N)$ which is equal to $O(FTN - AN^2)$. Here $(FT - NA)$ is the number of FTRPs that are in the ERQ. Eventually the worst case complexity of BFRA algorithm is : $O(NF) + O((FT - N) \times N) + O((FT - NA) \times N)$ which is roughly equal to $O(FNT)$.

Worst-case performance analysis of RAPB Algorithm: Since Step 1 of RAPB is the same as Step 1 of BFRA, in the worst case scenario this step has $O(NF)$ complexity. Step(2) - Step(5) include the steps to assign the $FT - N$ number of FTRPs to the unsatisfied SUs. In the worst case each FTRP is put back into the RQ for $N-1$ times and thus assigned to an SU for N times. Before each assignment, first the SU with minimum D value is found and this operation has $O(N)$ complexity. Furthermore, if we assume that all SUs have A antennas, then at each assignment A comparisons can be made to find an available antenna. Consequently, the computational complexity of the assignments of $FT-N$ number of FTRPs throughout Step 2- Step 5 is $O((FT-N)\times(N+A)\times N)$. Finally Step 6 is the same as Step 3 of BFRA algorithm and has order of $O((FT - NA) \times N)$. Summing these up and considering that the problem has an infeasible solution if and only if $FT \geq N$, and we in general have $N > A$, the complexity of RAPB algorithm is roughly $O(FTN^2)$.

4.2.4. Simulation Results

For the simulations the same CRN environment which was explained in the simulations section of Chapter 3 is used. Again the U_{if} values for 5000 scheduling periods in each set of simulations are obtained by taking PU velocities, SU velocities, and number of PUs into account [1]. In our experiments we vary N between 5 and 30, F between 3 and 30, a_i between 1 and 5 and finally PN_i between 20 and 300. Besides the simulations in which the PN_i value of each SU i is equal, we have had some simulations where the PN of an SU differs from the others' by some amount. In such experiments, the PN_i values of the N number of SUs are the elements of one of these sets: $Set1 = 40, 45, 50, ..., 40 + 5(N - 1)$,

$Set2 = 40, 50, 60, ..., 40 + 10(N-1)$, $Set3 = 40, 55, 70, ..., 40 + 15(N-1)$.

In Table 4.1 the second, third, fourth and fifth columns represent the N, F, a_i, and PN_i parameters of the simulation sets whose results are illustrated in the figure stated in the first column. In this table, the PN_i column contains *"Equal: 20, 40,..., 300"* expression if the PN_i value is equal for each SU i and this value varies between 20 and 300. On the contrary, if the PN_i value is different for each SU i, then the PN_i column contains *"Different:* $PNDiff = 5, 10, 15$*"* expression which shows that the PN values of the SUs are the elements of $Set1$ or $Set2$ or $Set3$.

Table 4.1. Simulation parameters.

Figure	N	F	a_i	PN_i
4.3	5, 15, 30	15	3	Equal: 20, 40,..., 300
4.4	15	3, 15, 30	3	Equal: 20, 40,..., 300
4.5	15	15	1, 3, 5	Equal: 20, 40,..., 300
4.6-1st	5, 10,..., 30	15	3	Different: $PNDiff = 5, 10, 15$
4.6-2nd	15	3, 6,...,30	3	Different: $PNDiff = 5, 10, 15$
4.6-3rd	15	15	1, 2,..., 5	Different: $PNDiff = 5, 10, 15$
4.7	5, 15, 30	15	3	Equal: 20, 40,..., 300
4.8	15	3, 15, 30	3	Equal: 20, 40,..., 300
4.9	15	15	1, 3, 5	Equal: 20, 40,..., 300
4.10-1st	5, 10,..., 30	15	3	Different: $PNDiff = 5, 10, 15$
4.10-2nd	15	3, 6,...,30	3	Different: $PNDiff = 5, 10, 15$
4.10-3rd	15	15	1, 2,..., 5	Different: $PNDiff = 5, 10, 15$
4.11	5, 15, 30	15	3	Equal: 20, 40,..., 300
4.12	15	3, 15, 30	3	Equal: 20, 40,..., 300
4.13	15	15	1, 3, 5	Equal: 20, 40,..., 300
4.14-1st	5, 10,..., 30	15	3	Different: $PNDiff = 5, 10, 15$
4.14-2nd	15	3, 6,...,30	3	Different: $PNDiff = 5, 10, 15$
4.14-3rd	15	15	1, 2,..., 5	Different: $PNDiff = 5, 10, 15$
4.15	5, 15, 30	15	3	Equal: 20, 40,..., 300
4.16	15	3, 15, 30	3	Equal: 20, 40,..., 300
4.17	15	15	1, 3, 5	Equal: 20, 40,..., 300
4.18	5, 10,..., 30	15	3	Different: $PNDiff = 5, 10, 15$
4.19	15	3, 6,...,30	3	Different: $PNDiff = 5, 10, 15$
4.20	15	15	1, 2,..., 5	Different: $PNDiff = 5, 10, 15$

In addition to the average N_s, in each simulation set we have evaluated the average throughput value for 5000 scheduling periods. Thus, we have also measured the performances of these algorithms in terms of total network throughput.

First of all, we have compared $BFRA_{FFO}$ with $BFRA_{TFO}$ and $RAPB_{FFO}$ with $RAPB_{TFO}$ to see which versions of the algorithms output better results in terms of the MNSU problem and the throughput of the CRN.

Figures 4.3, 4.4 and 4.5 illustrate the comparison of the average N_s and the average throughput results of $BFRA_{FFO}$ and $BFRA_{TFO}$ for PN_i varying between 20 and 300. In each simulation set, the PN_i value is equal for each SU i. Here, $BFRA-FFO_x$ and $BFRA-TFO_x$ denote the number of *"x"* used in the simulations of $BFRA_{FFO}$ and $BFRA_{TFO}$ respectively. This *"x"* stands for N, F and a_i in turn.

In Figure 4.3 we observe the N_s and the throughput values of the SUs where $F = 15$, $a_i = 3$, $N = 5, 15, 30$ for PN_i varying between 20 and 300. Notice that the N_s decreases as the PN_i value increases, especially when $N = 30$. Though in all of the cases the MNSU results of $BFRA_{FFO}$ and $BFRA_{TFO}$ are very close to each other, we still observe that $BFRA_{FFO}$ performs slightly better than $BFRA_{TFO}$ in some simulations. Furthermore, the throughput results of $BFRA_{FFO}$ are generally better than $BFRA_{TFO}$ for $N = 5$, $N = 10$, and $N = 15$.

In Figure 4.4 $BFRA_{FFO}$ and $BFRA_{TFO}$ are compared for $N = 15$, $a_i = 3$ and $F = 3, 15, 30$ for PN_i varying between 20 and 300. We observe that throughput values of the two versions are almost equal in all these cases. However, the N_s versus PN_i graph illustrates that $BFRA_{FFO}$ performs better in terms of N_s.

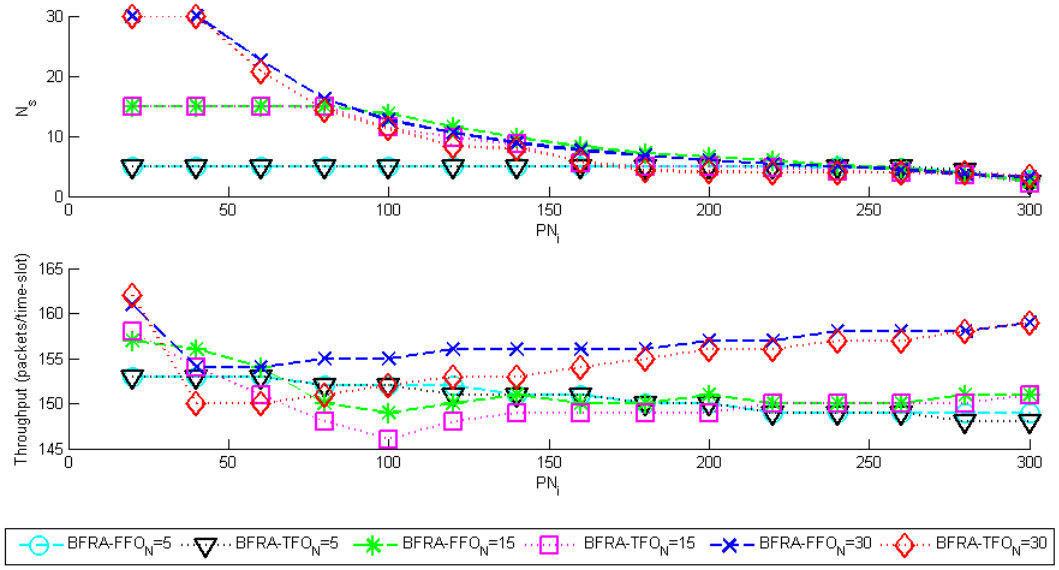


Figure 4.3. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying N.

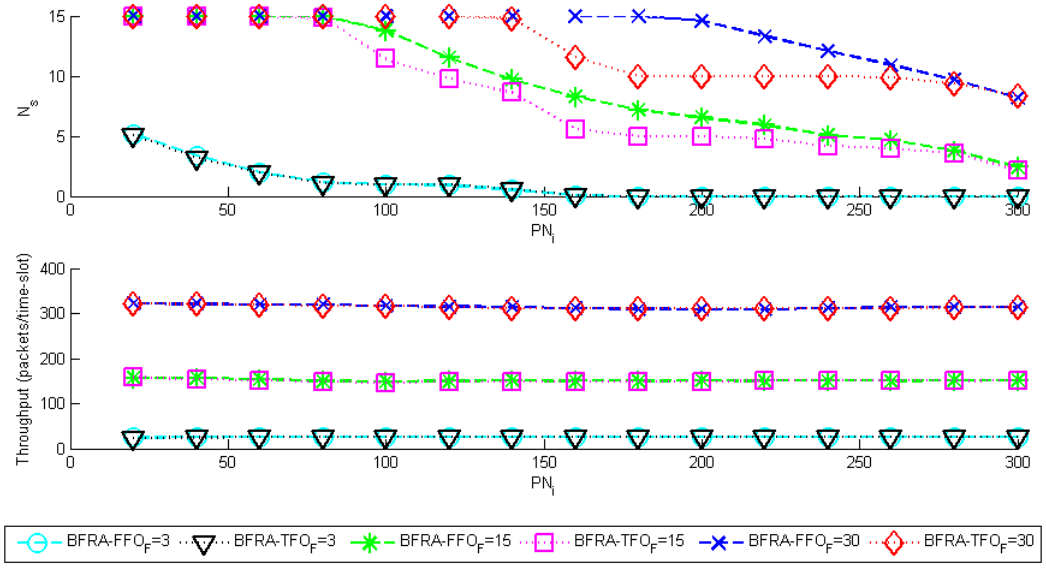


Figure 4.4. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying F.

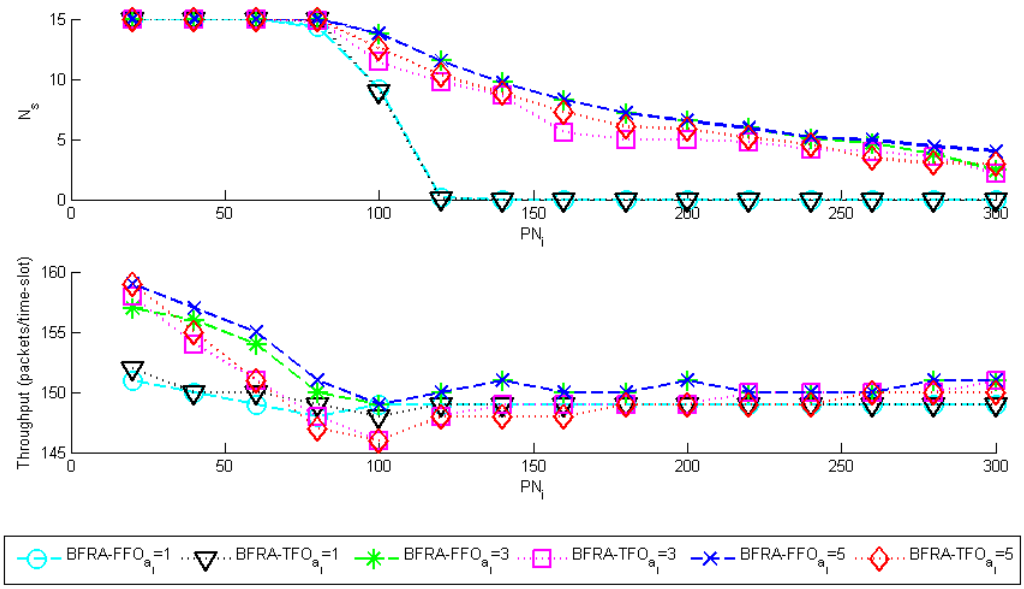


Figure 4.5. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ for varying a_i.

In Figure 4.5 the comparison is made for $N = 15$, $F = 15$, $a_i = 1, 3, 5$. In this figure the $BFRA_{FFO}$ results are either equal to or better than $BFRA_{TFO}$ in terms of both the N_s and the throughput values.

Besides the simulations in which the PN_i value of each SU i is equal, Figure 4.6 illustrates the experiments in which the PN values of the SUs increase by 5, 10 and 15. In this figure $BFRA - FFO_{PNDiff}$ and $BFRA - TFO_{PNDiff}$ denote the difference between the PN values of each consecutive SU in $BFRA_{FFO}$ and $BFRA_{TFO}$ simulation sets respectively.

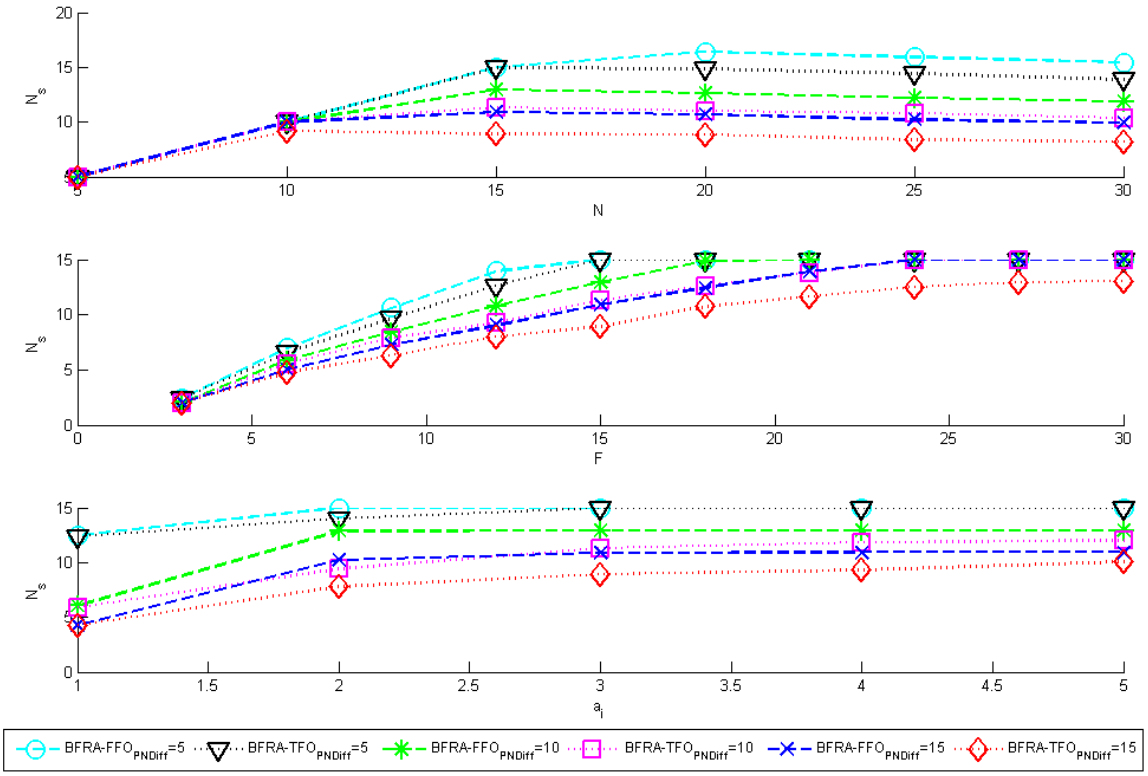


Figure 4.6. Comparison of $BFRA_{FFO}$ and $BFRA_{TFO}$ when the PN_i is different for each SU i.

Figure 4.6 illustrates the N_s results for varying N, F and a_i parameters. In the first graph $F = 15$, $a_i = 3$ and N varies between 5 and 30. In the second graph F varies between 3 and 30 for $N = 15$, $a_i = 3$. Finally, in the last graph N and F are equal to 15 and a_i varies between 1 and 5. For all of these simulation sets, we observe that the outputs of the $BFRA_{FFO}$ are better than the $BFRA_{TFO}$ outputs in terms of the N_s.

Relying on the simulations illustrated in the figures above, we can assert that the performance of $BFRA_{FFO}$ is better than the performance of $BFRA_{TFO}$ subject to the MNSU problem.

We have had the same simulations with $RAPB_{FFO}$ and $RAPB_{TFO}$ to find out which version of RAPB algorithm outputs better results. In Figures 4.7, 4.8, 4.9 and 4.10 the results of these simulations are presented. The PN_i values of the N number of SUs

are equal in the simulations illustrated in Figures 4.7, 4.8, 4.9. On the contrary, in Figure 4.10 the PN values of the SUs are the elements of $Set1$ or $Set2$ or $Set3$. In Figures 4.7, 4.8, 4.9 $RAPB-FFO_x$ and $RAPB-TFO_x$ denote the number of *"x"* used in the simulations of $RAPB_{FFO}$ and $RAPB_{TFO}$ respectively. Again *"x"* stands for N, F and a_i in turn.

Figure 4.7 illustrates the simulations where $F = 15$, $a_i = 3$, $N = 5, 15, 30$ for PN_i varying between 20 and 300. The first graph shows that the N_s results of the two versions are almost equal for these simulation scenarios. However, we observe that the throughput results of $RAPB_{FFO}$ are generally better than $RAPB_{TFO}$ results, especially for $N = 5$.

In Figure 4.8 the N_s and throughput results are shown for $N = 15$, $a_i = 3$, $F = 3, 15, 30$ and PN_i varying between 20 and 300. In this figure, we observe that the throughput values of the two versions are almost equal for these simulation parameters. Yet, the N_s results of $RAPB_{FFO}$ are better than the results of $RAPB_{TFO}$, especially when the PN_i values of the SUs increase.

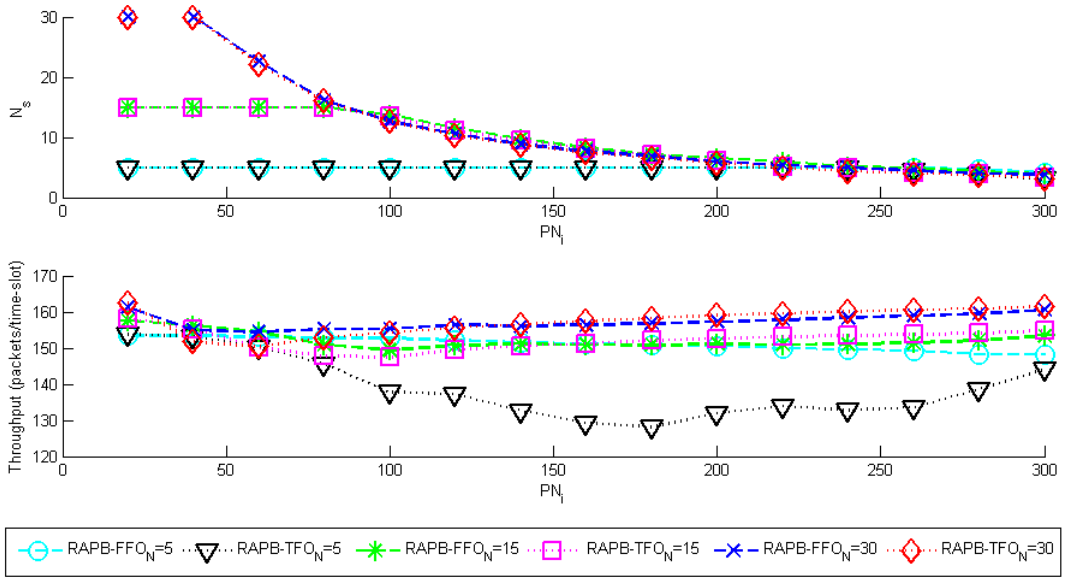


Figure 4.7. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying N.

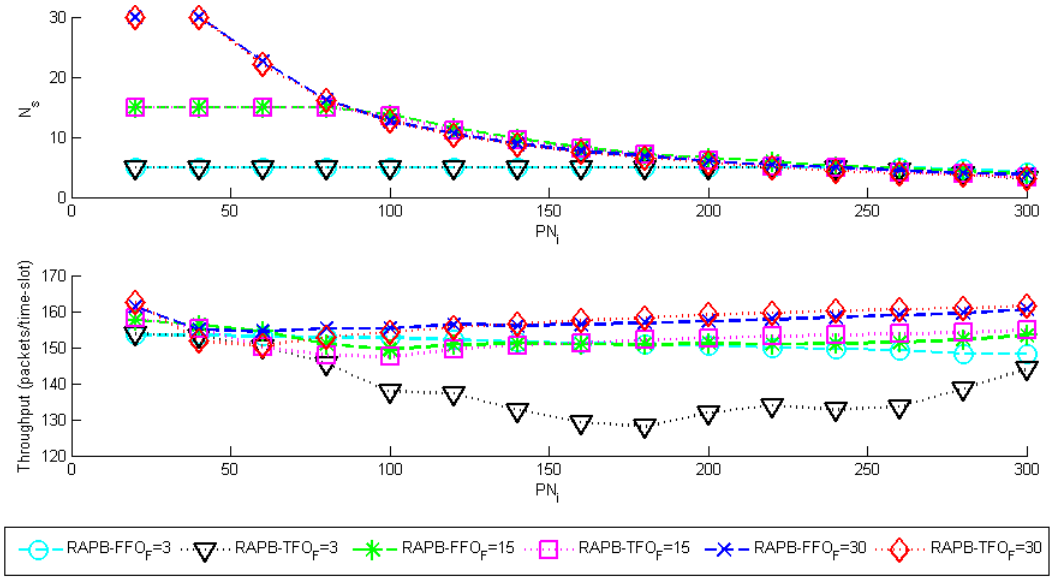


Figure 4.8. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying F.

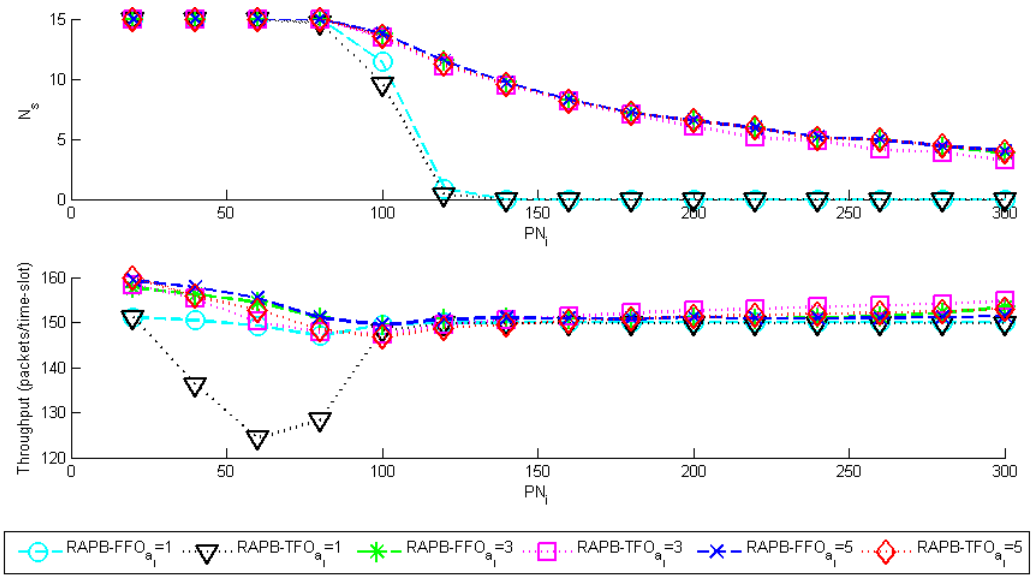


Figure 4.9. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ for varying a_i.

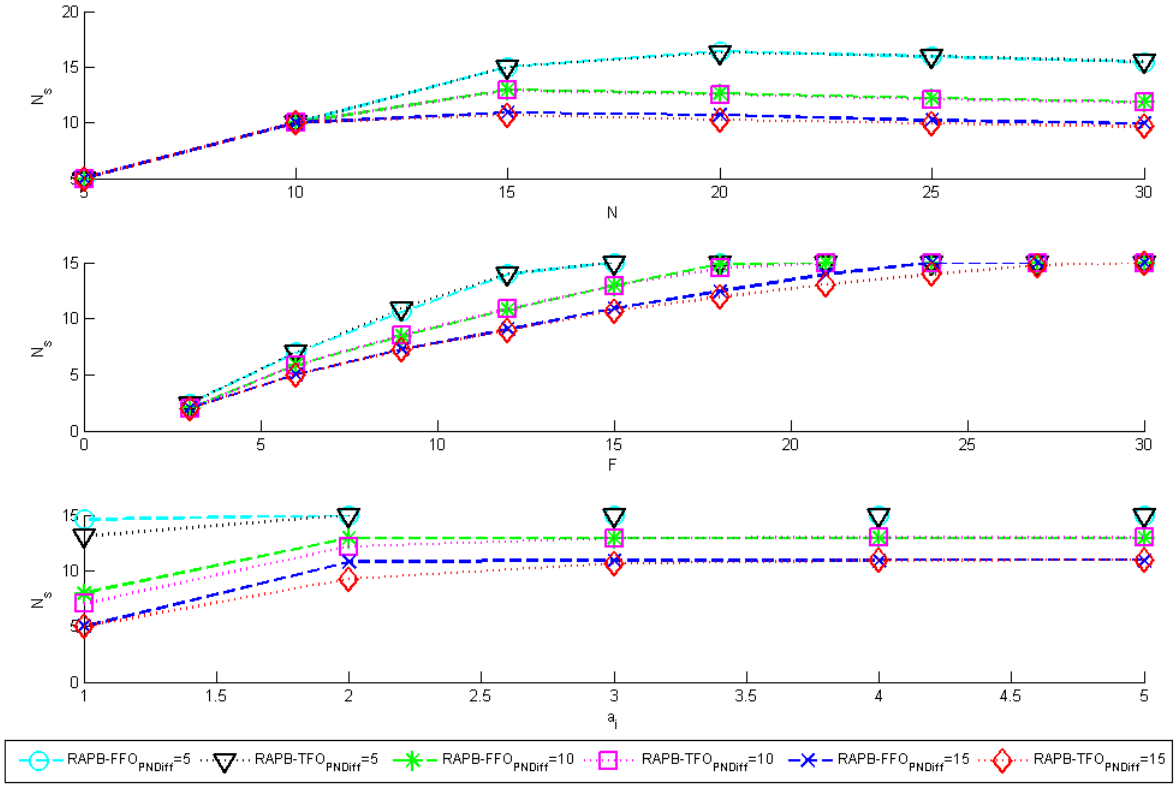


Figure 4.10. Comparison of $RAPB_{FFO}$ and $RAPB_{TFO}$ when the PN value differs for each SU.

In Figure 4.9 the two versions are compared for $N = 15$, $F = 15$, $a_i = 1, 3, 5$ and PN_i varying between 20 and 300. Even the N_s results are very close to each other, we can still claim that the N_s results of $RAPB_{FFO}$ are better than $RAPB_{TFO}$ results.

Finally, we have executed simulations where the PN_i values of the SUs in a scheduling period are different. That is, the PN values of the SUs are the elements of one of the sets: $Set1$, $Set2$, $Set3$. These simulations are executed for varying N, F and a_i. Figure 4.6 illustrates the N_s results versus these changing parameters. In all of these three graphs, we observe that actually $RAPB_{FFO}$ and $RAPB_{TFO}$ algorithms are almost equally good in terms of the N_s for these input parameters.

As a result of the comparisons we have made between $RAPB_{FFO}$ and $RAPB_{TFO}$, we have observed that even these two versions are generally equally good in terms of the

N_s and throughput of the SUs, $RAPB_{FFO}$ gives better results in a few input cases. Thus, we can claim that $RAPB_{FFO}$ performs better for the MNSU problem.

We have observed that putting the FTRPs into the RQ in a FFO manner gives better results for both of the BFRA and RAPB algorithms. Thus, in order to compare BFRA and RAPB algorithms' N_s results, we compared the simulation results of $BFRA_{FFO}$ and $RAPB_{FFO}$ versions with each other. These comparisons are illustrated in Figures 4.11, 4.12, 4.13 and 4.14. In Figures 4.11, 4.12, 4.13 $BFRO - FFO_x$ and $RAPB - FFO_x$ denote the number of *"x"* used in the simulations of $BFRA_{FFO}$ and $RAPB_{FFO}$ respectively. Again *"x"* stands for N, F and a_i in turn.

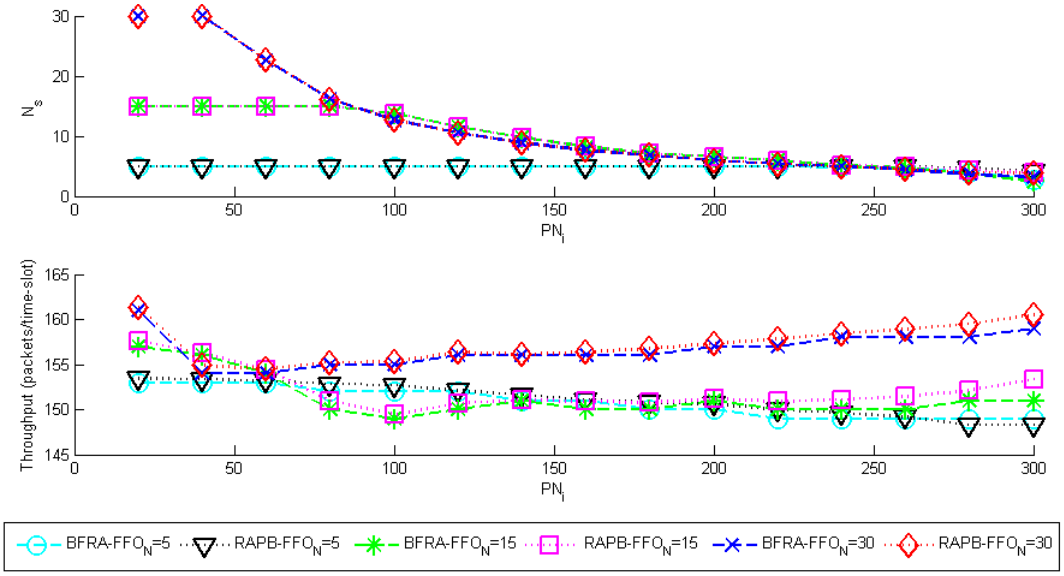


Figure 4.11. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying N.

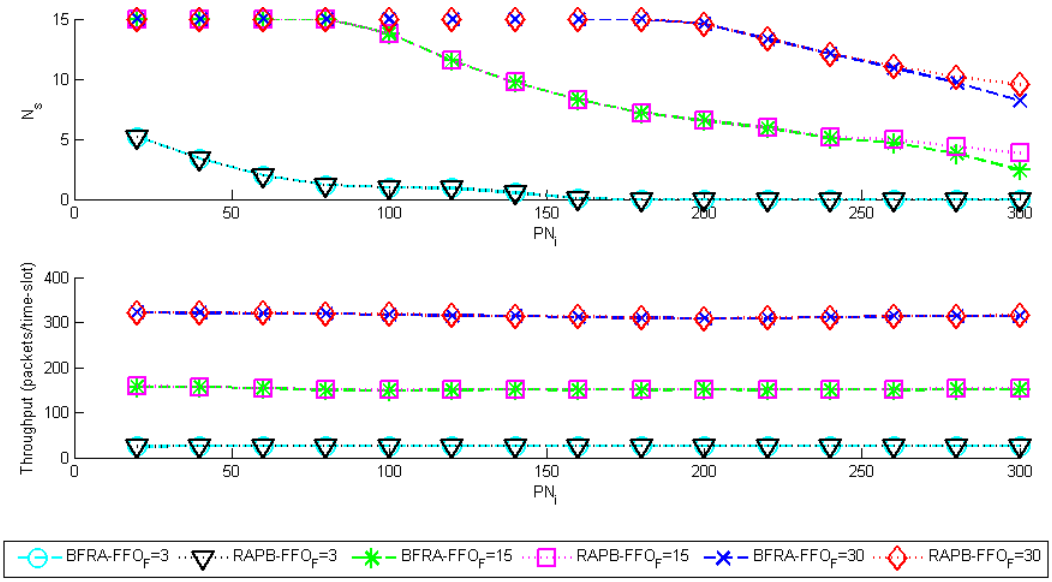


Figure 4.12. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying F.

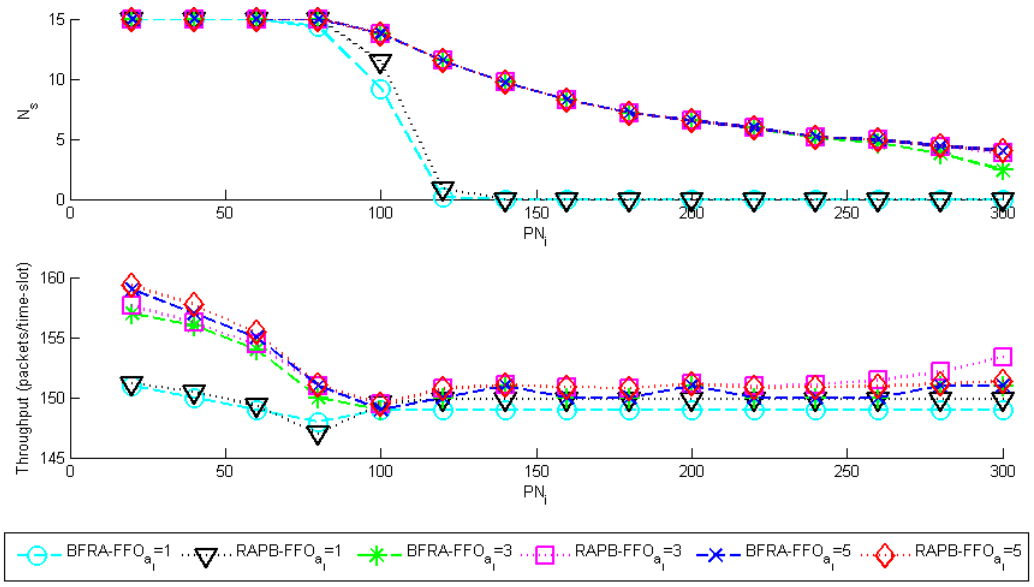


Figure 4.13. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ for varying a_i.

In Figure 4.11, $BFRA_{FFO}$ and $RAPB_{FFO}$ algorithm simulation results are compared for $F = 15$, $a_i = 3$, $N = 5, 15, 30$ and PN_i varies between 20 and 300. In this figure, the N_s results of the two algorithms are almost equal. However, $RAPB_{FFO}$ algorithm's throughput results are greater than $BFRA_{FFO}$ algorithm's results, especially when $PN_i > 240$.

In Figure 4.12 the algorithms are compared for $N = 15$, $a_i = 3$ and $F = 3, 15, 30$. We observe that their throughput and N_s results are very close to each other for these simulation scenarios. Additionally, we see that the N_s results of $RAPB_{FFO}$ becomes greater than $BFRA_{FFO}$ results when $PN_i > 260$.

In Figure 4.13 the simulation results for $N = 15$, $F = 15$, $a_i = 1, 3, 5$ are illustrated. We observe that in these simulations, $RAPB_{FFO}$ algorithm results are either equal to or better than $BFRA_{FFO}$ algorithm results both in terms of the N_s and throughput.

Finally in Figure 4.14 the two algorithms compared when the PN_i values of the N number of SUs are different in a scheduling period. That is, the set of PN_i values of the SUs is equal to $Set1$ or $Set2$ or $Set3$. In the first graph of the figure, N varies between 5 and 30 while $F = 15$, $a_i = 3$. In the second graph, $N = 15$, $a_i = 3$ and F varies between 3 and 30. Lastly, in the third graph F and N are equal to 15 while a_i varies between 1 and 5. For all of these simulations we observe that the N_s results of $RAPB_{FFO}$ are either equal to or slightly better than the outputs of $BFRA_{FFO}$ algorithm.

According to the simulations which are illustrated in Figures (4.3) -(4.14), we claim that $RAPB_{FFO}$ algorithm gives the best results among all of our proposed algorithms. Therefore, we have compared the N_s results of the $RAPB_{FFO}$ algorithm with the CPLEX solutions which are obtained by solving the BILP in 4.6- 4.10. Additionally, we have compared the throughput results of the $RAPB_{FFO}$ algorithm with the throughput values obtained by the CPLEX solutions. Since the N, F and a_i parameters differ for different simulation sets, the size of the problem changes from one simulation set to another. Thus

in the CPLEX computations, for each simulation set, we experimentally found and used the minimum epGap value [30] with which the solutions are gained in a reasonable time. Thus, fairness is provided to the small sized simulation sets.

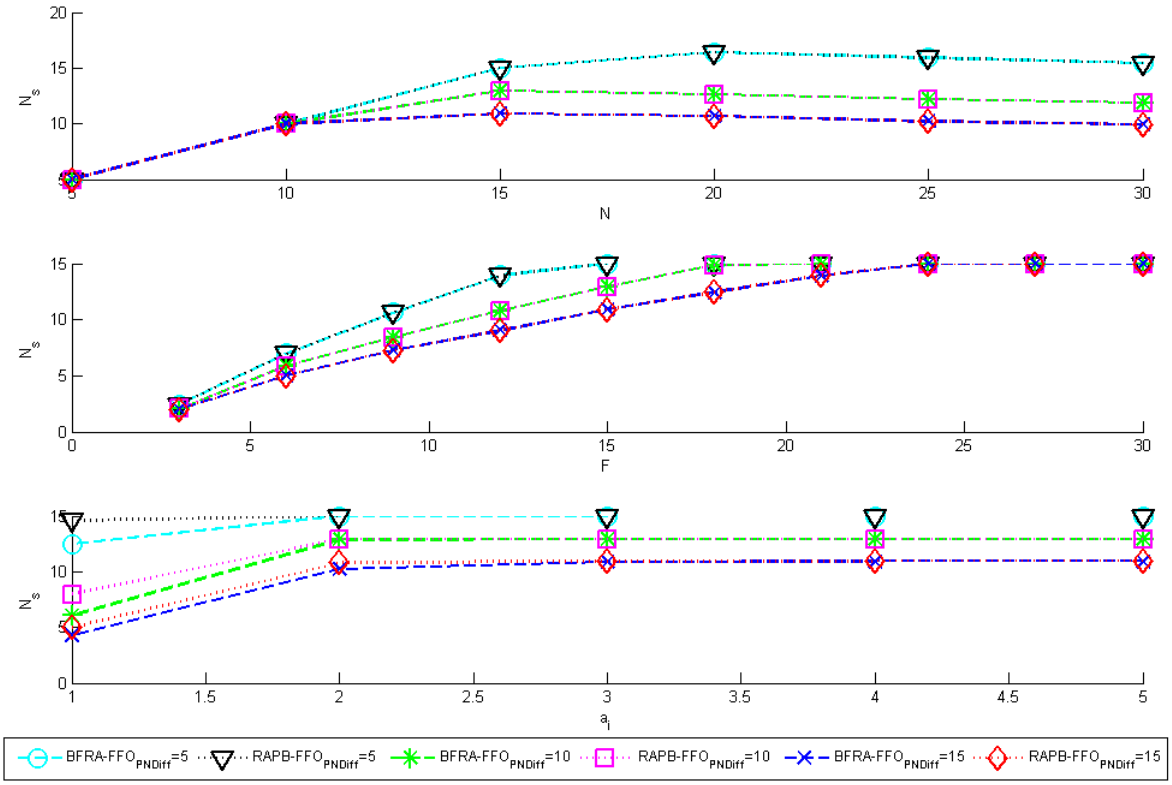


Figure 4.14. Comparison of $BFRA_{FFO}$ and $RAPB_{FFO}$ when the PN value differs for each SU.

In Figure 4.15 the outputs of $RAPB_{FFO}$ algorithm are compared with the CPLEX solutions for $F = 15$, $a_i = 3$, $N = 5, 10, 15$ for PN_i varying between 20 and 300. In this figure we observe that the N_s results of our proposed algorithm are very close to the corresponding CPLEX solutions. Furthermore, we clearly see that the throughput outputs of our proposed algorithm are better than or equal to the throughput values which are the outcomes of the CPLEX solutions.

In Figure 4.16 the outputs of $RAPB_{FFO}$ algorithm and the CPLEX solutions are compared for $N = 15$, $a_i = 3$, $F = 3, 15, 30$ where PN_i varies between 20-300. The PN

values of the SUs are equal in a scheduling period. In this figure, we observe that the N_s results of the $RAPB_{FFO}$ algorithm are very close to the CPLEX solutions. Moreover, $RAPB_{FFO}$ throughput values are better than the throughput values of the CPLEX solutions.

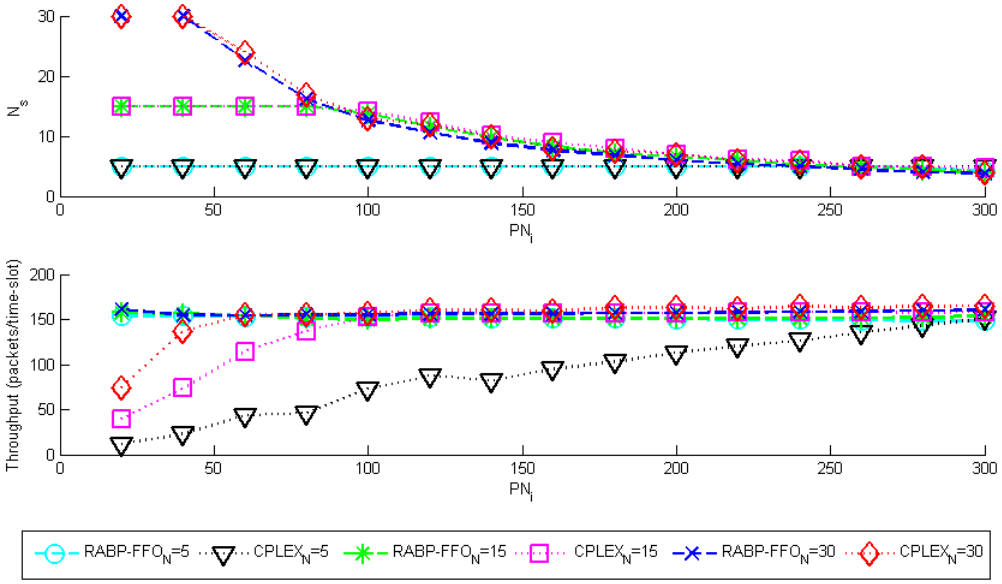


Figure 4.15. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying N.

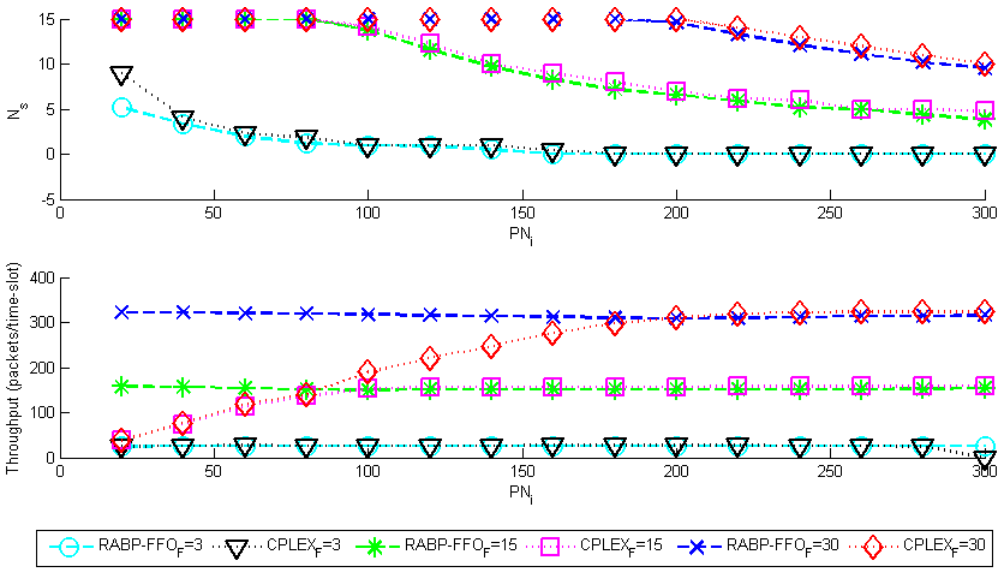


Figure 4.16. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying F.

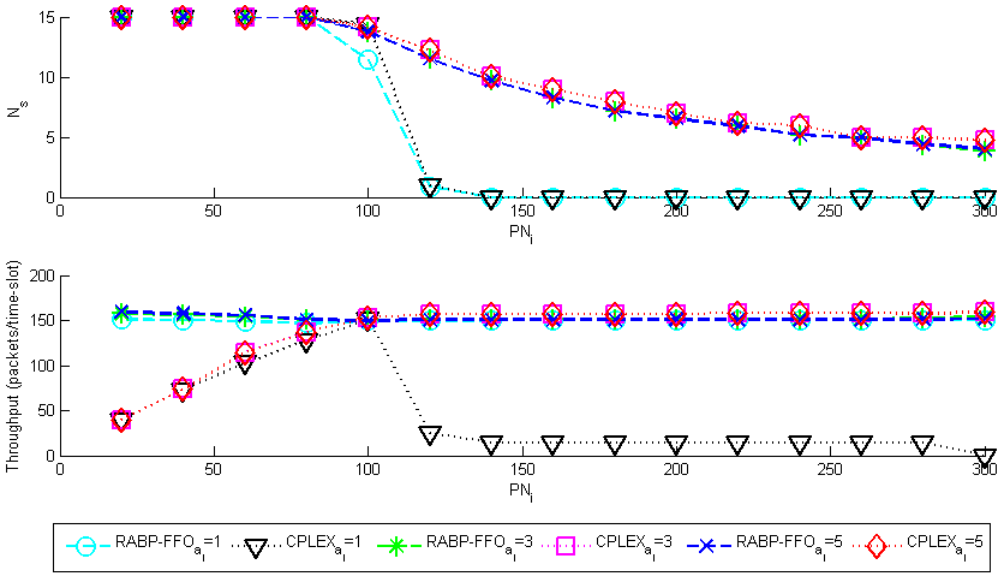


Figure 4.17. Comparison of $RAPB_{FFO}$ outputs with $CPLEX$ solutions for varying a_i.

Finally in Figure 4.17 the simulations are illustrated for $N = 15$, $F = 15$, $a_i = 1, 3, 5$ for PN_i varying between 20 and 300. In these simulations, the N_s results of $RAPB_{FFO}$ algorithm are very close to the CPLEX solutions. Moreover, the resulting throughput values are better than CPLEX throughput values, especially for $a_i = 1$.

The outputs of $RAPB_{FFO}$ algorithm and the CPLEX results of the simulation sets in which the PN_i values of the SUs are the elements of $Set1$ or $Set2$ or $Set3$, are illustrated in Figures 4.18, 4.19, and 4.20. In these figures $RAPB - FFO_{PNDiff}$ and $CPLEX_{PNDiff}$ denotes the difference between the PN values of each consecutive SU in $RAPB_{FFO}$ and CPLEX simulation sets respectively.

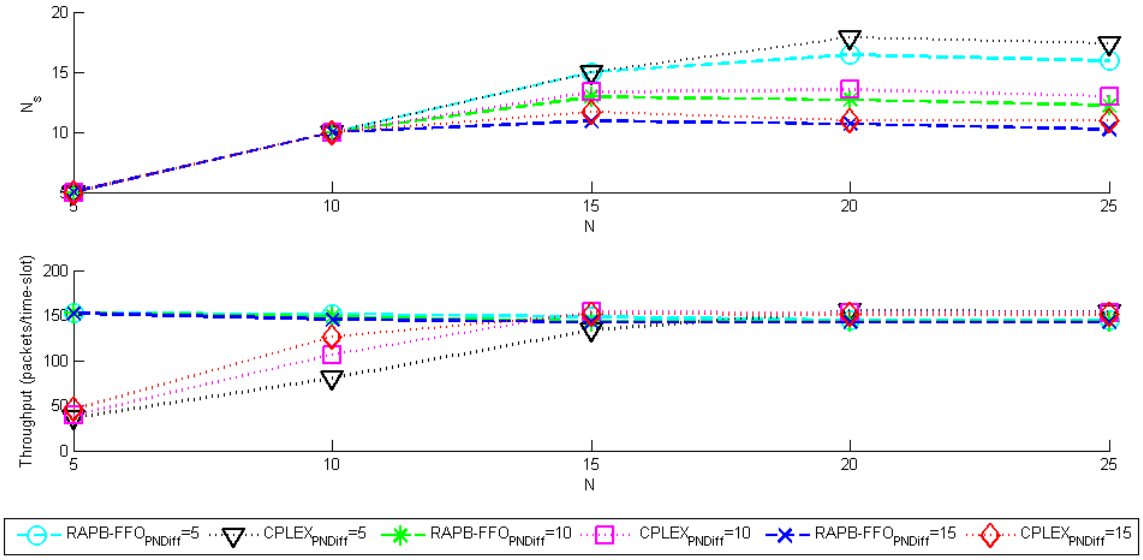


Figure 4.18. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying N.

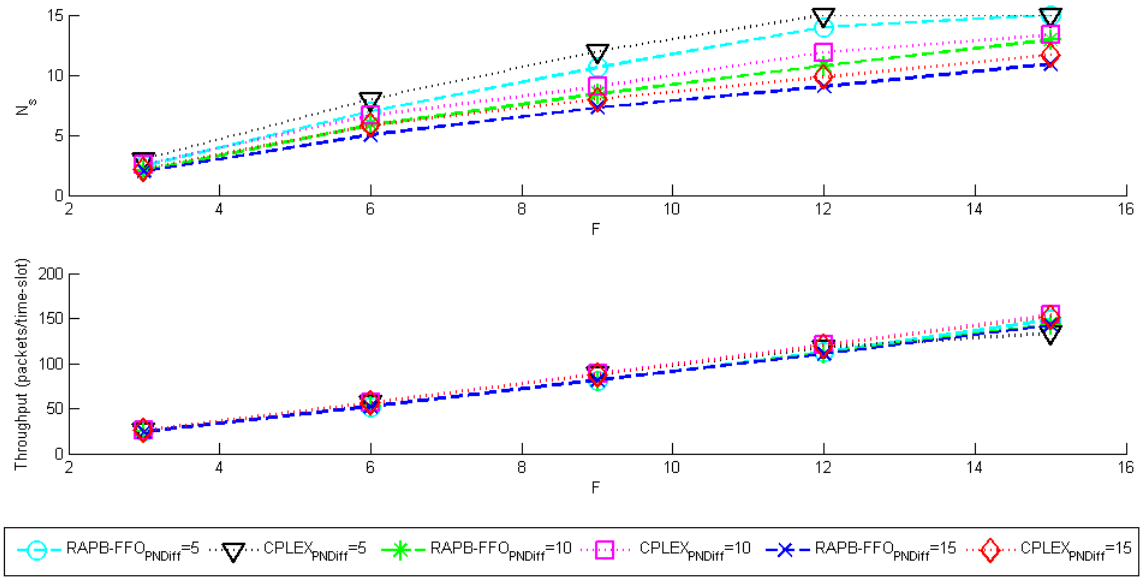


Figure 4.19. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying F.

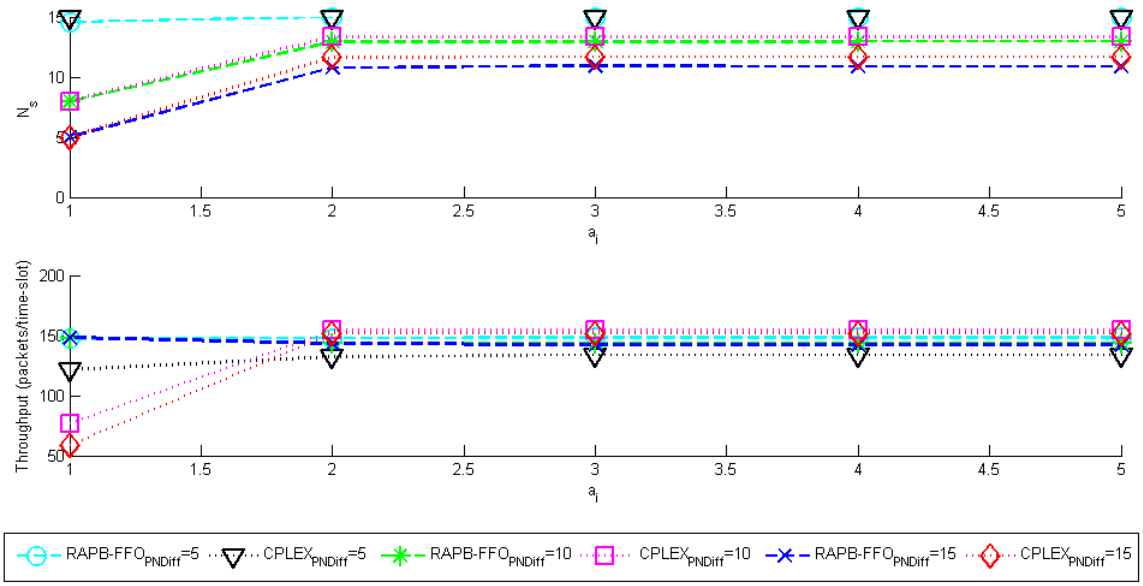


Figure 4.20. Comparison of $RAPB_{FFO}$ results with $CPLEX$ solutions when the PN value differs for each SU for varying a_i.

Figure 4.18 illustrates the simulations where N varies between 5 and 30, $F = 15$, $a_i = 3$. We observe that the N_s increases as the difference between the PN values of the consecutive SUs decreases in both the $RAPB_{FFO}$ outputs and the CPLEX solutions. Furthermore $RAPB_{FFO}$ throughput values are again either equal to or better than the CPLEX solutions.

Figure 4.19 illustrates the simulations where F varies between 3 and 30, $N = 15$, $a_i = 3$. We observe that for $F = 24$, all simulation sets saturate. That is, N_s becomes 15. Furthermore, we see that the PN set which consists of smallest values ($Set1$) saturate first. In this figure, the N_s results of $RAPB_{FFO}$ algorithm are very close to the CPLEX solutions and the throughput outputs are greater than or equal to the throughput values of CPLEX solutions.

Finally Figure 4.20 illustrates the simulations where $N = 15$, $F = 15$ and a_i varies between 1 and 5. In this figure we observe that the N_s increases when a_i becomes 2 when the PN_i values belong to either $Set2$ or $Set3$. The N_s results of our proposed algorithm are very close to the CPLEX solutions.

In conclusion, throughout the several simulations we have made, we have observed that the N_s results of BFRA and RAPB algorithms are very close to the CPLEX solutions. Furthermore the resulting throughput of the SUs in the CRN are greater than the throughput outcomes of the CPLEX solutions. Since both of these algorithms have low computational complexity, they are efficient heuristic algorithms for the MNSU problem in centralized CRNs.

5. CONCLUSIONS

In this thesis we have presented heuristic algorithms for the TMS and MNSU problems in a time slotted, centralized CRN model. The auction theory based scheduler that is proposed for the TMS problem, is based on a novel auction policy which uses first-price sealed bid auctions. The algorithm has $O((FN) + N2log(N))$ complexity in the worst case scenario and $O(FN)$ when the best case holds, where N is the total number of secondary users and F is the total number of frequencies in the cognitive radio cell. Our simulations with varying number of secondary users, frequencies and antennas show that the results of our proposed algorithm are very close to the values obtained by the CPLEX.

For the MNSU problem that is proven to be strongly NP-complete, we have proposed BFRA and RAPB heuristic algorithms. BFRA and RAPB algorithms have $O(FNT)$ and $O(FTN^2)$ worst case computational complexities respectively, where T is the number of time slots in a scheduling period. We have compared the simulation results of these two heuristic algorithms for varying number of secondary users, frequencies, antennas and packet needs and have observed that, even they are usually equally good, in some simulation scenarios RAPB algorithm has better outputs. Simulation results show that the N_s results of RAPB algorithm are very close to corresponding CPLEX solutions. Moreover, the throughput values of the RAPB algorithm are generally better than the throughput outcomes of the CPLEX solutions.

REFERENCES

1. Gözüpek, D., and F. Alagöz, An interference aware throughput maximizing scheduler for centralized cognitive radio networks, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2010, pp. 16.

2. Akyildiz, I. F., W. Y. Lee, M.C. Vuran and S. Mohanty, A Survey on Spectrum Management in Cognitive Radio Networks, IEEE Communications Magazine, April 2008

3. Mitola, J., Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio, Ph.D. Dissertation, Royal Inst. Technol.(KTH), Stockholm, Sweden, 2000.

4. Akyildiz, I. F., W. Y. Lee, M.C. Vuran and S. Mohanty, NeXt Generation / Dynamic Spectrum Access / Cognitive Radio Wireless Networks: A Survey, Computer Networks Journal, vol. 50, pp. 2127-2159, Elsevier, September 2006.

5. CPLEX, 1997, http://www.ilog.com/products/cplex, January 2011.

6. Chang, H.B., K.C. Chen, N. Prasad and C.W. Su, Auction Based Spectrum Management of Cognitive Radio Networks, IEEE Vehicular Technology Conference, 2009.

7. Niyato, D., and E. Hossain, Competitive Pricing for Spectrum Sharing in Cognitive Radio Networks: Dynamic Game, Inefficiency of Nash Equilibrium, and Collusion, IEEE Journal on Selected Areas in Communications, vol. 26, pp. 192-202, 2008.

8. Niyato, D., and E. Hossain, Competitive Spectrum Sharing in Cognitive Radio Networks: A Dynamic Game Approach, IEEE Transactions on Wireless Communications, vol. 7, pp. 2651-2660, 2008.

9. Zhu, J., and K. Liu, Multi-Stage Pricing Game for Collusion-Resistant Dynamic Spectrum Allocation, IEEE Journal on Selected Areas in Communications, vol. 26, pp. 182191, 2008.

10. Wang, F., M. Krunz, and S. Cui, Price-Based Spectrum Management in Cognitive Radio Networks, IEEE Journal on Selected Areas in Communications, vol. 2, pp. 7487, 2008.

11. Zheng, L., X. Wang, and L. Gao, Auction-Based Spectrum Sharing for Cognitive Radio Networks, IEEE International Conference on Computer Communications, 2009.

12. Tan, X., Y. Liu, and G. Xu, Dynamic Spectrum Allocation in Cognitive Radio: Auction and Equilibrium, International Forum on Information Technology and Applications, 2009.

13. Huang, J., R. Berry and M. L. Honig, Auction-Based Spectrum Sharing, ACM Mobile Networks and Applications Journal, vol. 11, no. 3, pp. 405-418, June 2006.

14. Xinbing, W., Z. Li, P. Xu, Y. Xu, X. Gao, H. Chen, Spectrum Sharing in Cognitive Radio Networks – An Auction Based Approach, IEEE Transactions on System, Man and Cybernetics–Part B: Cybernetics, vol. 40, no. 3, pp: 587-596, 2010.

15. Neumann, John von, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press. 1944, sec.ed. 1947, th.ed. 1953.

16. Ramis, J., L. Carrasco, G. Femenias, F. Riera-Palou, Scheduling Algorithms for 4G Wireless Networks, IFIP Personal Wireless Communications, vol. 245, pp: 264-276, 2007.

17. Liu, Y., and E. Knightly, Opportunistic Fair Scheduling Over Multiple Wireless Channels, IEEE International Conference on Computer Communications, vol. 2, pp. 1106-1115, 2003.

18. Liu, X., E.K.P. Chong, and N.B. Shroff, Transmission Scheduling for Efficient Wireless Utilization, IEEE International Conference on Computer Communications, vol. 3, pp. 776-785, 2000.

19. Uc-Rios, Carlos E., D. Lara-Rodriguez, A Low Complexity Scheduling for Maximizing Satisfied Users in Wireless Networks, International Conference on Signal Processing and Communication Systems, 2010.

20. Pal, S., S. Kundu, M. Chatterjee and Sajal K. Das, Combinatorial Reverse Auction Based Scheduling in Multi-rate Wireless Systems, IEEE Transactions on Computers, Vol. 56, No. 10, Oct. 2007, pp. 1329-1341.

21. Andrews M., et al., Providing Quality of Service over a Shared Wireless Link, IEEE Communications Mag., vol. 39, pp. 150-154, February 2001.

22. Pal, S., S. Kundu, A. R. Mazloom, and S. K. Das, Channel State Estimation and Scheduling Algorithms for Multi-Rate Wireless Systems, IEEE Conf. on Local Computer Networks, Oct. 2007.

23. Urgaonkar, R., and M. J. Neely, Opportunistic Scheduling with Reliability Guarantees in Cognitive Radio Networks, IEEE Transactions on Mobile Computing, vol. 8, no. 6, pp. 766-777, June 2009.

24. Lu, Q., T. Peng, W. Wang, W. Wang, C. Hu, Utility-Based Resource Allocation in Uplink of OFDMA-Based Cognitive Radio Networks, International Journal of Communication Systems , Vol. 23 Issue 2,February 2010.

25. Le, L. B., and E. Hossain, Resource Allocation for Spectrum Underlay in Cognitive Wireless Networks, IEEE Transactions on Wireless Communications, vol. 7, no. 12, pp. 5306-5315, Dec. 2008.

26. Lotfinezhad, M., B. Liang, and E. Sousa, Optimal Control of Constrained Cognitive Radio Networks with Dynamic Population Size, IEEE Conference on Computer Communications, San Diego, California, March 2010.

27. Menezes, F., and P. K. Monteiro, An Introduction to Auction Theory. Oxford University Press, 2005.

28. Fukunaga, Alex S., and Richard E. Korf, Bin Completion Algorithms for Multicontainer Packing, Knapsack, and Covering Problems, Journal of Artificial Intelligence Research 28, 393-429, March 2007.

29. Elyildirim B., D. Gozupek, F.Alagoz, An Auction Theory Based Algorithm for Throughput Maximizing Scheduling in Centralized Cognitive Radio Networks, IEEE Communications Letter, vol. 15, in Press, 2011.

30. Ilog Cplex 11.0 Parameter Reference Manual, September 2007, http://www-eio.upc.es/lceio/manuals/cplex-11/pdf/refparameterscplex.pdf, February 2011.