

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**CHANNEL SWITCHING COST-AWARE RESOURCE
ALLOCATION FOR MULTI-HOP COGNITIVE RADIO NETWORKS
WITH A SINGLE TRANSCEIVER**

MUSTAFA ÇAMURLI
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

GEBZE
2018

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**CHANNEL SWITCHING COST-AWARE
RESOURCE ALLOCATION FOR MULTI-
HOP COGNITIVE RADIO NETWORKS
WITH A SINGLE TRANSCEIVER**

MUSTAFA ÇAMURLI
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
ASSOC. PROF. DİDEM GÖZÜPEK KOCAMAN

GEBZE
2018

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞSEL RADYO AĞLARI İÇİN TEK
ANTENLİ CİHAZLAR ÜZERİNDE KANAL
DEĞİŞTİRME MALİYETİNİ DİKKATE
ALAN KAYNAK TAHSİSİ

MUSTAFA ÇAMURLI
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

DANIŞMANI
DOÇ. DR. DİDEM GÖZÜPEK KOCAMAN

GEBZE
2018

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/01../2018. tarih ve/04..... sayılı kararıyla oluşturulan jüri tarafından/01../2018. tarihinde tez savunma sınavı yapılan Mustafa.....ÇAMURU.....'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Doç. Dr. Didem Gözüpek Kocaman

ÜYE

: Doç. Dr. Hasani Gelebi

ÜYE

: Doç. Dr. Berh Canberli

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

Cognitive radio networks need to operate in a wide range of frequencies. This requirement brings up new challenges that do not exist in other wireless networks. Switching from a certain frequency to another frequency incurs a non-negligible cost in cognitive radio networks and depends on the wideness between the previous and current frequencies. This cost is especially important in ad hoc cognitive radio networks when the cognitive devices have a single transceiver. Frequency switching also leads to energy consumption. Research studies related to green networks indicate the need for methods that address energy consumption in cognitive radio networks; hence, the energy consumption due to frequency switching is of paramount importance.

In this thesis, we focus on two problems regarding cognitive radio network energy optimization. First, we focus on frequency switching cost aware resource allocation (FSCARA) problem. We analyze the impact of the frequency switching cost in terms of energy consumption by formulating an optimization problem that makes frequency and time slot allocation to the cognitive devices in an ad hoc cognitive radio network so that the energy cost related to frequency switching cost is minimized. We formulate our optimization problem as an integer linear program and comparatively evaluate the energy cost of varying switching energy consumption with constant switching energy consumption.

Second, we work on a related optimization problem called minimum changeover cost arborescence (MINCCA) problem. In an edge-colored graph, changeover cost corresponds to frequency switching cost in FSCARA problem. The objective of the MINCCA problem is to find a spanning tree which yields the minimum total changeover cost. We propose two heuristic algorithms for MINCCA problem. We compare the performance of our algorithms with the algorithm that calculates random changeover cost for varying parameters such as the number of colors, node and graph density. Simulation results demonstrate that our heuristic algorithms yield significantly better performance than random case.

Key Words: cognitive radio, ad-hoc networks, energy efficiency, channel switching, frequency switching, resource allocation.

ÖZET

Bilişsel radyo ağları, çalışmak için geniş frekans aralıklarına ihtiyaç duyarlar. Bu gereksinim diğer kablosuz ağlarda olmayan yeni zorlukları beraberinde getirir. Bilişsel radyo ağlarında bir frekanstan diğer frekansa geçişin maliyeti göz ardı edilemeyecek kadar büyük olup değişim, frekanslar arasındaki uzaklığa bağlıdır. Bu maliyet özellikle tek antenli bilişsel radyo ağı cihazları için önem taşımaktadır. Yeşil ağlar ile ilgili çalışmalar bilişsel radyo ağlarının enerji ihtiyacını düzenleyen metodlara ihtiyaç duyulduğunu göstermektedir. Bu tezde, bilişsel radyo ağlarında enerji optimizasyonu konusu ile ilgili iki farklı problem üzerinde yoğunlaştık.

İlk bölümde frekans değişiminin etkisini enerji tüketimi açısından inceledik. Öncelikle frekans değişim maliyetini dikkate alan kaynak tahsisi (FSCARA) problemine odaklandık. Tasarsız bilişsel radyo ağlarında bilişsel cihazlar için frekans ve zaman dilimi atamasını frekans değişimlerinden ortaya çıkan enerji maliyetini en aza indirecek şekilde yapan bir optimizasyon problemi formüle ettik. Bu optimizasyon problemini tamsayı lineer programlama yöntemi ile tasarladık ve değişken frekans geçişi ile sabit frekans geçişi arasındaki enerji maliyetlerini karşılaştırmalı olarak inceledik.

İkinci kısımda, benzer bir optimizasyon problemi olan ağaç yapılarında en az değişim maliyeti (MINCCA) problemi üzerine çalıştık. Kenarları renklendirilmiş bir çizgede değişim maliyeti, FSCARA problemindeki frekans değişimi maliyetine karşılık gelmektedir. MINCCA problemi en az toplam değişim maliyeti olan bir kapsayan ağacı bulmayı amaçlar. MINCCA probleminin çözümü için 2 adet sezgisel algoritma geliştirdik. Geliştirdiğimiz bu algoritmaları rastgele maliyet hesaplayan bir algoritma ile karşılaştırdık. Sonuçlar geliştirdiğimiz algoritmaların rastgelelikten daha iyi çalıştığını ve çizgede artan kenar ve renklere göre de tutarlı bir sonuç ortaya koyduğunu göstermektedir.

Anahtar Kelimeler: bilişsel radyo, tasarsız ağ, enerji verimliliği, frekans değiştirme, kanal değiştirme, kaynak ataması.

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my supervisor, Assoc.Prof. Didem GÖZÜPEK KOCAMAN, who not only shared her profound scientific knowledge with me but also taught me great lessons of life. Her support, suggestions and encouragement gave me the drive and will to complete this work. It would be almost impossible to finish this work without her encouragement and constant help. I am also grateful to my parents and my wife for their support.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	iv
ÖZET	v
ACKNOWLEDGMENTS	vi
TABLE of CONTENTS	vii
LIST of ABBREVIATIONS and ACRONYMS	viii
LIST of FIGURES	ix
LIST of TABLES	x
1. INTRODUCTION	1
1.1. Contribution	1
1.2. Thesis outline	2
2. RELATED WORK	3
3. PROBLEM FORMULATION	5
3.1. ILP Formulation	5
3.2. Simulation Results	9
4. HEURISTOC ALGORITHM FOR CHANGE OVER COST MINIMIZATION PROBLEM	14
4.1. Problem Formulation	14
4.2. Heuristic Algorithms	14
4.3. Simulation Results	18
5. CONCLUSION	25
REFERENCES	26
BIOGRAPHY	29

LIST of ABBREVIATIONS and ACRONYMS

<u>Abbreviations and Acronyms</u>	<u>Explanations</u>
PU	: Primary User
SU	: Secondary User
CR	: Cognitive Radio
CRN	: Cognitive Radio Network
ILP	: Integer Linear Programming

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
3.1: Average switching cost for varying number of frequencies.	11
3.2: Average switching cost for varying number of secondary users.	12
3.3: Average switching cost for varying number of primary users.	13
4.1: MINCS algorithm for MINCCA.	15
4.2: MINCSS algorithm for MINCCA.	17
4.3: Impact of number of color changes for Regular Random graph.	19
4.4: Impact of vertex degrees for Regular Random graph.	20
4.5: Impact of number of node changes for Regular Random graph.	21
4.6: Impact of number of color changes for Watts Strogatz Graph.	22
4.7: Impact of vertex degrees for Watts Strogatz Graph.	23
4.8: Impact of number of node changes for Watts Strogatz Graph.	24

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
3.1: Table for decision variables.	5
3.2: Table for input variables.	6
3.3: Parameter names and range of values evaluated in the experiments.	11

1. INTRODUCTION

Fixed spectrum assignment policy in current wireless networks leads to inefficiency in spectrum usage. To this end, researchers have proposed dynamic spectrum access concept, which refers to opportunistic usage of frequencies by intelligent devices called cognitive radios (CR) or secondary users (SU). Users who have exclusive rights to use the spectrum are called primary users (PU). SUs need to operate in such a way that the operation of PUs is not affected. Availability of a particular frequency to a particular SU depends on the spectrum usage behaviors of the PUs in the vicinity of the SUs [1].

An SU consumes energy to change its operation frequency. This energy consumption in general depends on the wideness between the previous and current frequency bands [2]. For instance, switching from central frequency of 700 MHz to 10 GHz leads to larger energy consumption than switching from 700 MHz to 750 MHz. This behavior is directly related to the different channel switching delay while switching to different frequency bands [3] since spending more time for switching naturally consumes more energy. This energy consumption might be negligible in wireless technologies that operate in a narrow band; however, cognitive radios are expected to operate in a wide range of spectrum and hence the energy consumption due to channel switching is crucial for cognitive radio networks [2]. The emerging green networking paradigm that focuses on energy efficiency of both wired and wireless technologies puts emphasis on techniques that enhance the energy efficiency of new technologies such as cognitive radio networks [4].

1.1. Contribution

In the work we analyze to impact of frequency switching for multi-hop CRNs with a single transceiver in terms of energy consumption. To this end, we formulate an optimization problem as an Integer Linear Programming (ILP). Simulation results demonstrate that taking into account varying switching cost is of paramount importance.

We also develop and discuss two heuristic algorithms: Minimum Changeover Solver (MINCS) and Minimum Changeover Sorted Solver (MINCSS) for MINCCA problem. Our simulation results indicate that MINCS and MINCSS perform significantly better than random case.

1.2. Thesis Outline

This thesis consists of two major parts. First, we demonstrate problem formulation and results for energy consumption minimization caused by channel switching in CRN networks. Second, we present two different greedy algorithms for changeover cost minimization problem.

- Chapter 3 includes ILP formulation tests and results for the FSCARA problem. In this chapter we define input and decision variables and also objective function which sets the minimization criteria for our model. Then we show test results and compare them by constant frequency switching cost case.
- Chapter 4 presents two heuristic algorithms we proposed for the MINCCA problem as well as the comparative simulation results.

2. RELATED WORK

Studies in the literature analyzed the impact of channel switching in CRNs mostly in terms of delay [5]-[10]. Most of these works focus primarily on minimizing the number of channel switching along the routing path; in other words, they assume constant switching delay whenever channel switching occurs irrespective of from which frequency and to which frequency the channel switching has taken place. However, frequency switching delay, which also is referred to as hardware delay caused by the delay of the frequency synthesizer of the antenna is not constant. Instead, it is highly dependent on the source and final frequencies. Since frequency switching is handled in a step by step manner, oscillator must be locked into one phase before switching to the next one on the frequency band. In the literature, only the works in [5] and [6] consider the dependency of channel switching delay on the wideness between the previous and current frequencies.

Another aspect of the cost of frequency switching is channel switching power. The works [26], [27] show that energy required for channel switching is almost the same as the required energy for data transmission. For example, the works in [26], [27] demonstrate that the power that wireless card running on Linux for switching is the same the power for data transmission (both 750 mW). In a similar way, in the work [2], they take the channel switching power as 1000 mW in their setup.

The work in [3], on the other hand, focuses on scheduling in centralized cognitive radio networks and analyzes the impact of channel switching delay on throughput. The authors propose a scheduling algorithm that considers the different hardware delay that occurs while switching to different frequency bands. The work in [11] focuses on minimizing the service interruption by taking into account the dependency of switching delay on the frequency separation distance. However, unlike our work, the work in [2] concentrates on scheduling in centralized cognitive radio networks.

Most works about routing in cognitive radio networks consider the case where the CR devices have multiple transceivers/interfaces [11]-[15]. However, having multiple transceivers is costly and therefore, scenarios where CRs have a single transceiver are more realistic. There are some works in the literature that focuses on the case with single transceivers [16]-[18]. In a multi-hop CRN, channel switching

cost becomes especially important when each CR has a single transceiver. However, unlike our work, none of the works in [16]-[18] focus on the energy cost related to the channel switching of the single transceiver.

In the literature, many applications involving network optimization problems are modeled using edge-colored graphs. In an edge-colored graph, there can be a route which has color changes along a path. These color changes are considered as the cost of the operating application. The cost metric is usually referred to as reload cost or changeover cost.

Reload cost concept was introduced by Wirth and Steffan [28] which is taken as a problem of finding a spanning tree having minimum diameter in terms of reload cost. Later on, in the work [29] the same problem was introduced in a different way. The work [30] concentrates on the minimum reload cost cycle cover problem. In the study, [31] introduced the Minimum Changeover Cost Arborescence (MINCCA) problem for directed edge-colored graphs. Most works [33]-[36] focus on computational complexity and parametrized complexity of the MINCCA problem. In addition, the work [32] focused on the special cases of the problem such as bounded degree, bounded cost values and bounded number of colors.

The concept of changeover cost can be used in many network optimization problems which can be modeled by edge-colored graphs. In communication networks, especially in cellular networks whose structure is mostly heterogeneous, changeover cost is incurred while sending packets from one endpoint to another. Optimizing the changeover cost in such networks increases the efficiency of data transmission. Transportation networks is another application area for the MINCCA problem. Each transportation option can be represented by a color and cargo delivery from one carrier to another can be modeled by the concept of changeover cost. Cognitive Radio Network (CRN) is a wireless network model where cognitive agents tend to communicate with other agents by using the best available frequencies in order to optimize power usage and delay. CRN networks can also be modeled by edge-colored graphs where colors correspond to available frequencies for each agent. In such a network, power and delay optimization can be modeled as the MINCCA problem, which we focus on in this thesis.

3. PROBLEM FORMULATION

3.1. ILP Formulation

We represent the ad hoc CRN as a graph $G = (V, E)$, where the vertices in V correspond to the SUs and the edges in E corresponds to the communication links between the SUs. We consider the situation where the routing is predetermined; in other words, if a link exists between a pair of SUs, it means that link is used to carry data along some route. The scheduling period consists of T time slots. There are a total number of F frequencies. A subset of these F frequencies is available for each SU in each time slot. Availability of the frequencies is determined by the PU spectrum occupancy behavior in the geographical region of the ad hoc CRN. The goal of our integer linear programming (ILP) formulation is to assign a time slot and frequency to each SU so that the total energy consumption of the SUs due to frequency switching is minimized.

Table 3.1: Table for decision variables.

Variable	Description
x_{vf}	$= \begin{cases} 1, & \text{if vertex } v \text{ is assigned frequency } f \text{ in some time slot} \\ 0, & \text{otherwise.} \end{cases}$
x_{eft}	$= \begin{cases} 1, & \text{if edge } e \text{ is assigned frequency } f \text{ in time slot } t \\ 0, & \text{otherwise.} \end{cases}$
x_{vft}	$= \begin{cases} 1, & \text{if vertex } v \text{ is assigned frequency } f \text{ in time slot } t \\ 0, & \text{otherwise.} \end{cases}$
y_{vft}	$= \begin{cases} 1, & \text{if } x_{vft} = 1 \text{ or } x_{vf(t-1)} = 1 \\ 0, & \text{otherwise.} \end{cases}$
$y_{vff't}$	$= \begin{cases} 1, & \text{if } y_{vft} = 1 \text{ or } y_{vf't} = 1 \\ 0, & \text{otherwise.} \end{cases}$

Table 3.1 and 3.2 present the decision and input variables, respectively, of our ILP formulation. If $x_{vft} = 1$, it means the single transceiver of the SU corresponding to vertex v is tuned to use frequency f in one of its incident links (an incident link e with $x_{eft} = 1$) in time slot t . Besides, $y_{vff't}$ is the main decision variable used in the

calculation of the objective function. Basically, $y_{vff't} = 1$ if vertex v switches from frequency f to f' in time slot t

Table 2 shows input variables that are given to the ILP formulation. The value of switching cost $c_{ff'}$ denotes the energy consumption that occurs while switching from frequency f to f' and depends on the hardware of the cognitive nodes. Besides, the values of a_{vft} are determined by the spectrum usage behavior of the PUs.

Objective function of our ILP formulation is as follows:

$$\min \sum_{v=1}^{|V|} \sum_{f=0}^F \sum_{f'=0}^F \sum_{t=0}^T c_{ff'} \times y_{vff't} \quad (3.1)$$

Table 3.2: Table for input variables.

Input Variable	Description
$c_{ff'}$	= Switching cost (in terms of energy consumption) to transition from frequency f to frequency f'
a_{vft}	= $\begin{cases} 1, & \text{if frequency } f \text{ is available for vertex } v \text{ in time slot } t \\ 0, & \text{otherwise.} \end{cases}$
$G = (V, E)$	= Network represented as a graph with vertex set V and edge set E
T	= Total number of time slots
F	= Total number of frequencies

The objective function in Equation (3.1) minimizes the switching cost of all vertices (nodes) in the graph (network). The decision variable $y_{vff't}$ indicates whether there has been a frequency switching at vertex v from f to f' in time slot t ; i.e., basically, $y_{vff't} = y_{vft} \times y_{vf'(t-1)}$ (this relation will be provided by the constraints and explained later).

Recall that routing is predetermined in our scenario. In other words, every edge $e \in E$ carries data and should therefore be assigned at least one time slot during the scheduling period. The following constraint achieves this behavior:

$$\sum_{f=1}^F \sum_{t=1}^T x_{eft} \geq 1; \forall e \in E \quad (3.2)$$

Since only available frequencies can be assigned to a particular vertex, we need the following constraint:

$$x_{vft} \leq a_{vft}; \forall v, f, t \quad (3.3)$$

Since each node (vertex) has a single transceiver, at most one incident edge can be given service in each time slot. Therefore, we need to ensure that $x_{vft} = \sum_{e \in E(v)} x_{eft}$. Furthermore, since the values of the decision variables $y_{vff't}$ depend on the frequency assignments in the previous time slot, in order to properly handle the situation in the first time slot, we define an auxiliary frequency $f = 0$ and an auxiliary time slot $t = 0$. The auxiliary frequency $f = 0$ is not assigned to any vertex for time slots $t \geq 1$. Besides, all vertices are assigned frequency $f = 0$ in time slot $t = 0$ (will be enforced by constraints (3.13) and (3.14)). The inputs $c_{ff'}$ are given such that the switching cost from frequency 0 to all other frequencies equals zero. This behavior is necessary in order to linearize our integer programming formulation and handle the border case in the first time slot. Hence, we need the following constraints:

$$x_{vft} = \sum_{e \in E(v)} x_{eft} = 0; \text{ for } f = 0 \quad (3.4)$$

$$x_{vft} = \sum_{e \in E(v)} x_{eft} \geq 0; \forall f \geq 1 \quad (3.5)$$

The following constraints ensure the multiplication of the decision variables such that $y_{vff't} = y_{vft} \times y_{vff'(t-1)}$. This way, if there is a switching between frequencies f' and f at vertex v , the variable $y_{vff't}$ is set to 1.

$$y_{vff't} \leq y_{vft}; \forall v, f, t \geq 1 \quad (3.6)$$

$$y_{vff't} \leq y_{vf(t-1)}; \forall v, f', t \geq 1 \quad (3.7)$$

$$y_{vft} + y_{vf'(t-1)} \leq y_{vff't}; \forall v, f, f', t \geq 1 \quad (3.8)$$

Notice that it is not a requirement that each vertex is assigned some frequency in every time slot; some vertices may not be assigned a frequency in some time slots. We refer to such a time slot as a *silent time slot* for that vertex. We assume that the transceiver of the node is tuned to the last used frequency in that time slot. When some other frequency is used in a time slot following the silent time slot(s), an energy consumption cost corresponding to frequency switching occurs. We need to make sure that the decision variable $y_{vff't}$ handles the silent time slots properly. To this end, we need to ensure the following set of nonlinear inequalities: a) If $x_{vft} = 0$, then $y_{vft} = y_{vf(t-1)}$ and b) If $x_{vft} = 1$, then $y_{vft} = y_{vf(t-1)}$. These nonlinear constraints can be linearized as follows:

$$y_{vft} - y_{vf(t-1)} \leq x_{vft}; \forall v, f, t \geq 1 \quad (3.9)$$

$$x_{vft} \leq y_{vft}; \forall v, f, t \geq 1 \quad (3.10)$$

$$y_{vf(t-1)} - y_{vft} \leq x_{vft}; \forall v, f, t \geq 1 \quad (3.11)$$

Frequency $f = 0$ and time slot $t = 0$ are auxiliary time slots to handle the value of $y_{vff't}$ in the first time slot $t = 1$. All vertices are assigned frequency $f = 0$ in time slot $t = 0$. Switching cost between frequency $f = 0$ and any other frequency is zero. Frequency $f = 0$ cannot be used in other time slots $t \neq 0$.

$$x_{v00} = 1; \forall v \quad (3.12)$$

$$y_{v00} = 1; \forall v \quad (3.13)$$

Since each node has a single transceiver and a transceiver can tune to at most one frequency at a time, we need to add a constraint that ensures that at most one frequency is assigned to each vertex in each time slot:

$$\sum_{f=1}^F x_{vft} \leq 1; \forall v, t \quad (3.14)$$

We also need to model the relationship between the decision variables x_{vt} and x_{vft} . In particular, if $x_{vt} = 1$, then there must be a frequency f assigned to vertex v in timeslot t :

$$\sum_{f=1}^F x_{vft} = x_{vt}; \forall v, t \quad (3.15)$$

In order to increase the utilization in the network and achieve more throughput, less queueing delay and more realistic network behavior, we need to avoid the situation that two adjacent vertices are idle. To this end, at least one of two neighboring nodes should be assigned some frequency in each time slot. Let $N(v)$ denote the neighborhood of vertex v , i.e., the set of vertices that are adjacent to vertex v . The following constraint models this behavior:

$$x_{vt} + x_{v't} \geq 1; \forall t, v, v' \in N(v) \quad (3.16)$$

3.2. Simulation Results

In this section, we present our simulation results and comparatively evaluate our ILP formulation solutions with different input parameter settings. We have implemented our ILP formulation introduced in Section 3.1 by using CPLEX [19]. We have made performance evaluation under different number of frequencies, primary users and secondary users. In our simulation environment, PUs and SUs are distributed randomly in a $400 m^2$ two dimensional Euclidean space. As in [11], we set the operation range of each SU to a radius of 110 m. If two SUs are within the operation range of each other, then there is an edge connecting the two vertices corresponding to these SUs in the input graph G fed into our ILP formulation. Likewise, we set the operation range of each PU to 170 m radius. An SU within the operation range of a PU cannot use the frequency used by that PU in that time slot. Furthermore, each

scheduling period consists of 20 time slots, i.e., $T = 20$. All PUs are assigned a random frequency at the beginning of the simulation. In every time slot, each PU changes its frequency with a probability of $P_c = 0.1$. If a PU changes its frequency, the probability of selecting each frequency is equally likely.

In 2009 full power analog television stopped operation and hence, the frequency range between 54 MHz and 806 MHz were cleared due to the transition to digital TV [20]. In accordance with these frequency bands that became available for cognitive radio operation, we set the frequency range in our experiments between 460 MHz and 700 MHz with the bandwidth of each channel being 20 Mhz. In other words, the highest number of frequencies used in our experiments is 13. According to the work in [2], switching latency is 0.1 ms/MHz. In the same article, channel switching power is assumed to be 1 W. In accordance with this information, we set in our model the energy consumption incurred due to switching between two frequencies with a difference of 20 MHz as 0.002 Joule.

We have evaluated the impact of the following three parameters in our simulations: the number of primary users, the number of secondary users, and the number of frequencies. Values of the other parameters remain constant in all experiments. As a result, we calculate two separate solutions corresponding to two different cost schemes. Notice that the energy consumption due to frequency switching is proportional to the time it takes to complete the frequency switching, i.e., switching delay. We refer to the first scheme as varying cost. Varying cost scheme relies on the fact that switching delay and hence the energy consumption due to frequency switching depends on the wideness between the previous and current frequencies. We refer to the second scheme as fixed cost. Fixed cost scheme relies on the assumption that switching delay and hence the energy consumption due to frequency switching is the same (constant value) as long as the previous and current frequencies are different. We have taken this constant value as 0.002 Joule in our simulations.

Table 3 displays the range of values for the evaluated parameters. When investigating the impact of a particular parameter, we set the value of the other parameters to their middle values, which are also shown in Table 3.3. We take the sample mean of 15 random experiments in each setup.

Table 3.3: Parameter names and range of values evaluated in the experiments.

Parameter	Range Min	Range Max	Middle Value
Number of frequencies	5	13	10
Number of SUs	5	13	10
Number of PUs	4	7	5

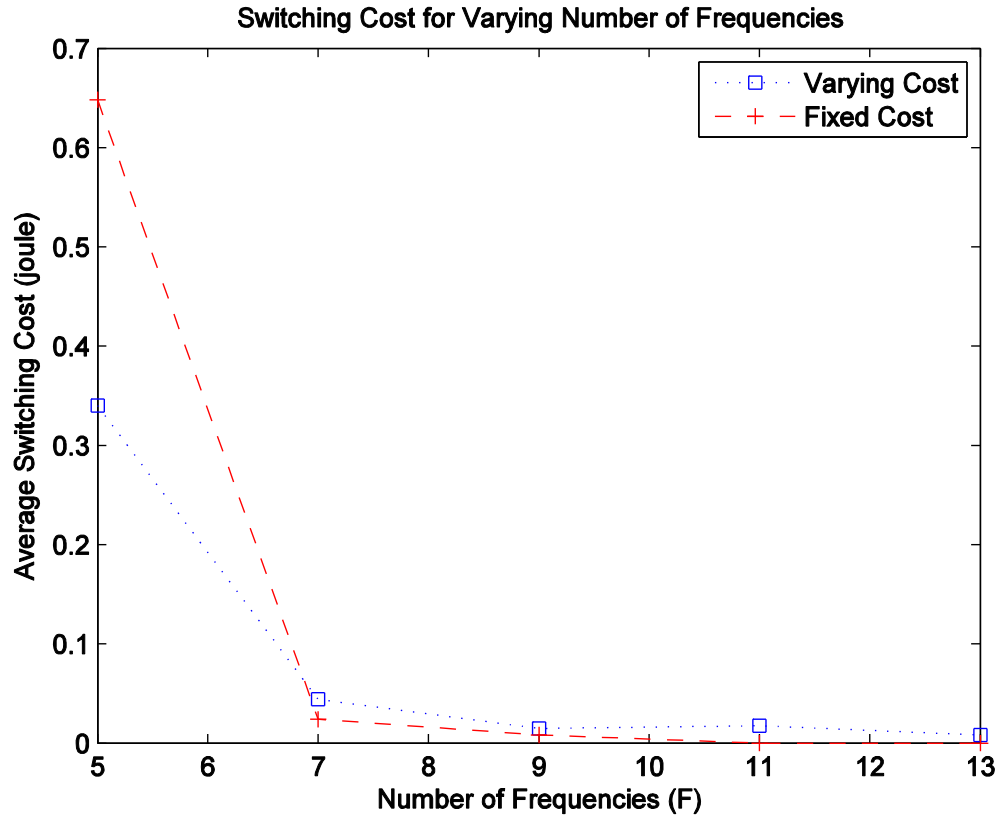


Figure 3.1: Average switching cost for varying number of frequencies.

Figure 3.1 shows that varying cost scheme is advantageous when the number of frequencies in the CRN is less than 7. Furthermore, when the number of frequencies is high, the cardinality of the frequency set that is available for the SUs also increases. This increasing availability makes it possible to switch to closer frequencies and hence decreases the energy consumption due to frequency switching.

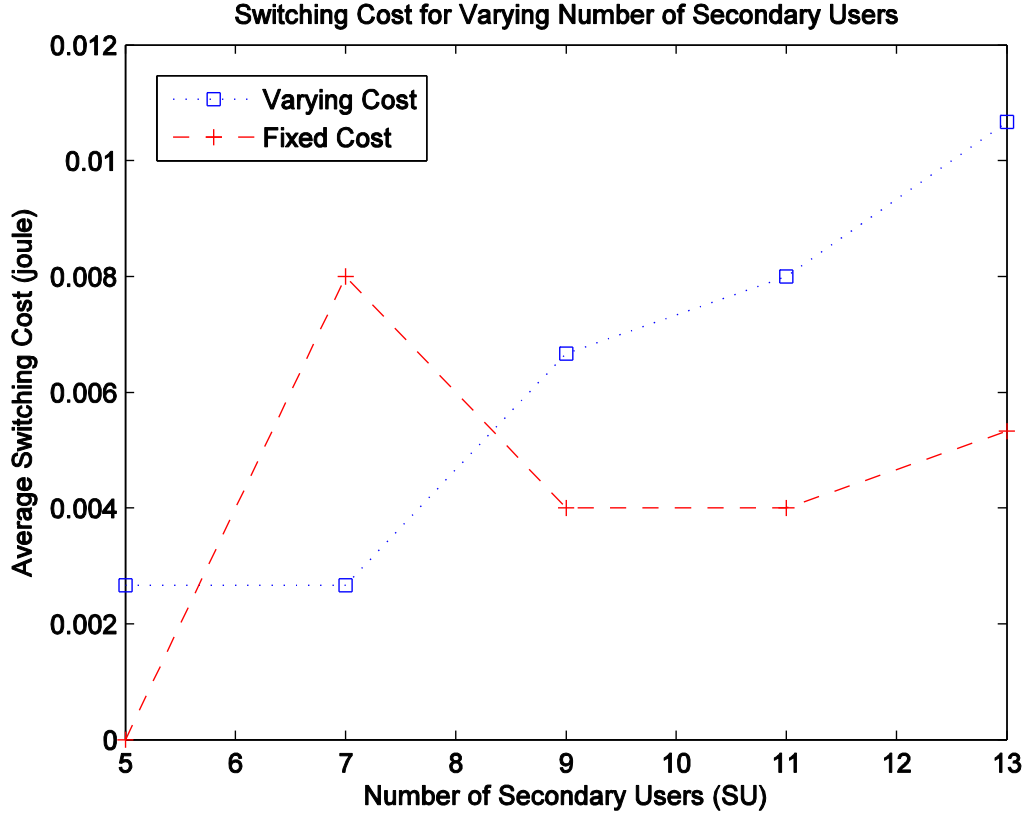


Figure 3.2: Average switching cost for varying number of secondary users.

In Figure 3.2, we observe that the number of secondary users does not have a consistent and stable impact on the performance of the fixed cost scheme, which fluctuates and is hence unpredictable. In contrast, the switching cost of varying cost scheme increases linearly with number of secondary users. In some parts, fixed cost scheme performs better than varying cost. However, its unpredictability can discourage cognitive radio systems to rely on fixed switching cost in order to schedule their networks. As a result, predictable and consistent performance of varying cost scheme (in addition to the fact that it is more realistic) can make it preferable.

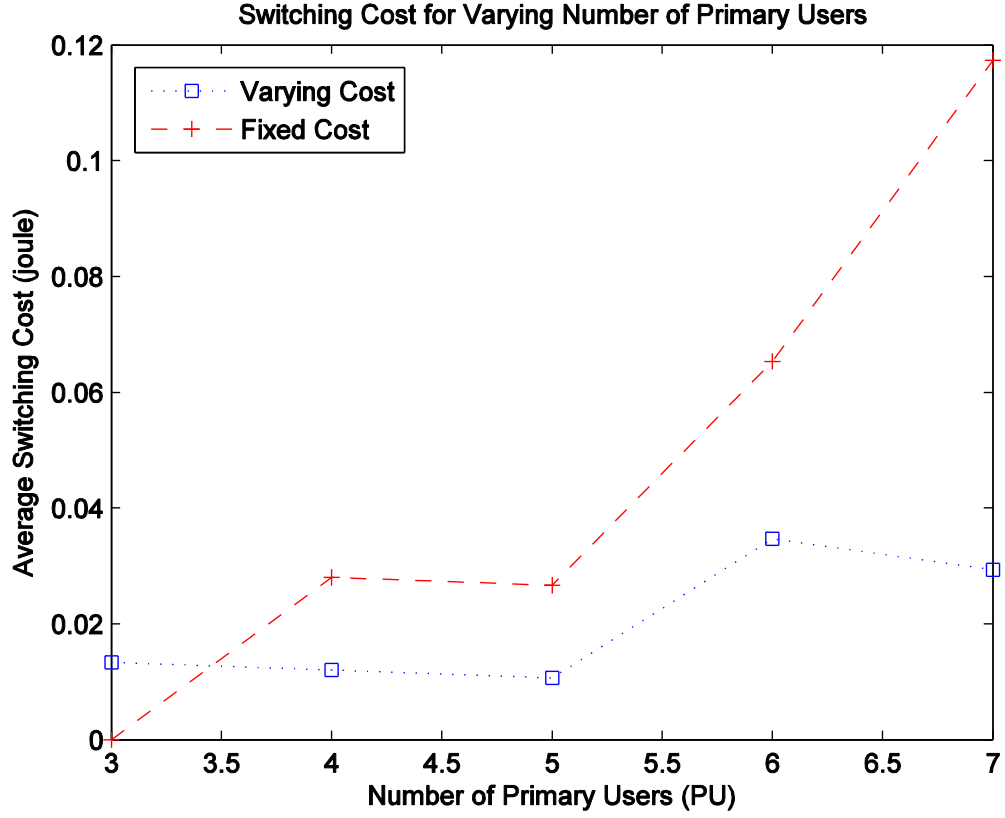


Figure 3.3: Average switching cost for varying number of primary users.

Figure 3.3 shows the impact of increasing number of PUs. Since the frequencies that are available for the SUs decrease as the number of PUs increases, SUs become obliged to make frequency switching more often and between increasingly distant frequencies. Especially when we have more than 5 primary users, average switching cost of fixed cost solutions increases faster than varying cost solutions. Figure 3.3 demonstrates the strength of our varying cost scheme since it outperforms the constant cost scheme especially when there is a high number of PUs.

4. HEURISTIC ALGORITHMS FOR MINCCA PROBLEM

4.1. Problem Formulation

Given an edge-colored connected undirected graph $G = (V, E)$, a root vertex r and a coloring function $C: E(G) \rightarrow c$, $COC: (e_1, e_2) \rightarrow \mathbb{R}^+ \forall e_1, e_2 \in E(G)$ denotes the changeover cost function. The goal is to find a spanning arborescence of G rooted at r such that the total changeover cost is minimized.

Given a tree T and a root vertex r , for each path from root vertex r to all leaves constitute a series of color selections depending on which edges are used. For each path, a changeover cost function can be defined as follows:

- $P = (e_1, e_2, \dots, e_k)$, where path has length k .
- $COC_p = \sum_{i=1}^k COC(e_i, e_{i+1})$ where $k > 0$, otherwise $COC_p = 0$

Total changeover cost is the sum of all paths from root vertex r to all leaves. Let $P_{all} = \{P_1, P_2, \dots, P_n\}$ be the set of all paths from root vertex to leaves. Then total changeover cost is $COC_{total} = \sum_{i=1}^n COC_i$.

Changeover cost function has the following properties:

- $COC(e_1, e_2) = |C(e_1) - C(e_2)|$
- $COC(e_1, e_2) \equiv COC(e_2, e_1)$
- $COC(e_1, e_1) = 0$

4.2. Heuristic Algorithms

In this section, we present two heuristic algorithms: Minimum Changeover Solver (MINCS) and Minimum Changeover Sorted Solver (MINCSS).

Figure 4 describes the MINCS algorithm, which starts with root vertex and greedily selects a vertex which generates minimum changeover cost in each iteration. A set called *visited* is used to keep track of the visited vertices. Selected vertices are

added to the visited set. When a vertex is selected, the edge which connects the selected vertex to visited vertices is also selected. Algorithm runs until all vertices are visited. Finally, the selected edges yield the resulting spanning tree.

MINCSS algorithm follows a similar strategy. However, initially it makes some modification on the input graph. We assume that if colors are alpha numeric, we can map all used colors to a numeric value. Firstly, the algorithm sorts edges by numeric color values. Then a spanning tree is constructed by using this sorted vertex list. In order to construct the spanning tree, we start with an empty graph. Then in each iteration, we add the median vertex to the spanning tree unless it creates a cycle. If the addition is successful, we remove the median vertex from the sorted list.

Algorithm: Minimum Changeover Solver (MINCS)

Input: A edge colored graph $G=(V,E)$, root vertex r

Output: A spanning tree

```

1  spanning_tree  $\leftarrow V(G)$ 
2  vertex_set  $\leftarrow V(G)$ 
3  visited  $\leftarrow \{r\}$ 
4  incoming_color_table[ $r$ ]  $\leftarrow 0$ 
5
6  while visited  $\neq$  vertex_set do
7      min_cost  $\leftarrow \infty$ 
8      source_vertex  $\leftarrow \text{NULL}$ 
9      min_cost_vertex  $\leftarrow \text{NULL}$ 
10     min_incoming_color  $\leftarrow 0$ 
11
12     for  $i = 0$  until  $|visited|$  do
13          $v \leftarrow visited.index(i)$ 
14         neighbours  $\leftarrow v.get\_unvisited\_neighbors()$ 
15
16         for  $j = 0$  until length of neighbours do
17              $n \leftarrow neighbors.index(j)$ 

```

Figure 4.1: MINCS algorithm for MINCCA.

Figure 4.1: Continue.

```

18      current_color = get_color(get_edge(n, v))
19      temp_cost
      = get_cost(incoming_color_table[v], current_color)
20
21      if min_cost > temp_cost then
22          min_cost  $\leftarrow$  temp_cost
23          source_vertex  $\leftarrow$  v
24          min_cost_vertex  $\leftarrow$  n
25          min_incoming_color  $\leftarrow$  current_color
26
27      visited  $\leftarrow$  visited  $\cup$  {min_cost_vertex}
28      incoming_color_table[min_cost_vertex]
           $\leftarrow$  min_incoming_color
29      spanning_tree  $\cup$  get_edge(source_vertex, min_cost_vertex)
30  return spanning_tree
31
32  function get_cost(c1, c2)
33      if c1 == 0 or c2 == 0 then
34          return 0
35      return absolute(c1 - c2)

```

MINCSS is a greedy heuristic algorithm. It uses four variables to construct the spanning tree. Lines 1 - 4 are initialization of these variables: *panning_tree* , *vertex_set* , *visited* and *incoming_color_table*. *spanning_tree* and *vertex_set* are initialized to all vertices of input graph G, *visited* is initialized to root vertex and *incoming_color_table* is an array each index of which corresponds to a vertex that holds the color of the incoming edge. The vertex *r* index of *incoming_color_table* is initialized to 0. The algorithm iterates until all vertices in graph G are visited. In lines 7, 8, 9 and 10, temporary *min_cost*, *source_vertex*, *min_cost_vertex* and *min_incoming_color* variables are initialized to infinity and empty. The for loop in line 12 iterates over all visited vertices and lines 13 and 14 gather neighbors of the current *visited* vertex. Line 16 iterates over all neighbors and lines 17, 18 and 19

calculates the changeover cost for the current neighbor. The if statement in line 21 decides on whether the current neighbor creates the minimum cost. If it yields the minimum changeover cost, *min_cost*, *source_vertex*, *min_cost_vertex* and *min_incoming_color* are assigned to *temp_cost*, source vertex *v*, current vertex *n*, and *current_color* variables, respectively. Finally, lines 27, 28 and 29 add the *min_cost_vertex* vertex which generates the minimum color cost to the *visited* set. Then *min_cost_vertex* index of *incoming_color_table* is assigned with *min_incoming_color* and it adds the edge which connects *source_vertex* and *min_cost_vertex* to the *spanning_tree*.

Now we analyze the computational complexity of the MINCS algorithm. In the worst case, the input is a complete graph. Therefore, the while loop in line 4 iterates for all vertices. The for loop in line 7 iterates once in the beginning and $|V| - 1$ times in the last iteration. The for loop in line 10 iterates $|V| - 1$ times for each iteration. To sum up,

$$i) \text{ MINCS} \in O(|V| \times (|V| - 1) \times (|V| - 1)) \in O(|V|^3)$$

Algorithm: Minimum Changeover Sorted Solver

Input: A edge colored graph $G=(V,E)$, root vertex r

Output: A spanning tree

```

1  spanning_tree  $\leftarrow \{\emptyset\}$ 
2   $V(\textit{spanning\_tree}) \leftarrow V(\textit{spanning\_tree}) \cup V(G)$ 
3  edge_list  $\leftarrow E(G)$ 
4  sort_with_respect_to_color(edge_list)
5
6  while spanning_tree is not connected do
7       $E(\textit{spanning\_tree})$ 
           $\leftarrow E(\textit{spanning\_tree}) \cup \textit{edge\_list.get\_median}()$ 
8      edge_list.remove\_median()
9
10 return spanning_tree
```

Figure 4.2: MINCSS algorithm for MINCCA.

Unlike algorithm MINCS, algorithm MINCSS does not traverse the graph from root to leaves. Instead, it finds a spanning tree with edge colors that are close to each other in terms of numerical distance. The algorithm first sorts the edges by their numeric color values. Then it constructs a spanning tree by iteratively adding the median edge to the tree by avoiding cycles. Lines 1-3 are initializations. *spanning_tree* is initialized to empty. Then all vertices of the input graph are iteratively added to *spanning_tree*. Then *edge_list* is initialized to all edges of the input graph. In line 4, *edge_list* is sorted by the numeric value of the edge color. In lines 6, 7 and 8 in each iteration of the while loop, the median value of the *edge_list* is removed so that a spanning tree is constructed in the end.

Now we analyze the computational complexity of MINCSS algorithm. Sorting takes $O(|E| \log(|E|))$ time (via an algorithm such as quicksort). Lines 6, 7 and 8 take at most $O(|E| - 1)$ iteration to create a spanning tree. As a result, if we consider a complete graph, the complexity of the MINCSS algorithm is:

- i) $O(|E|) \in O(|V|^2)$
- ii) $\text{MINCSS} \in O(|E| \log(|E|)) + O(|E| - 1) \in O(|V|^2 \log(|V|^2) + O(|V|^2))$
- iii) $\text{MINCSS} \in O(|V|^3)$

4.3. Simulation Results

We evaluate the performance of MINCS and MINCSS algorithms with two types of graphs called Regular Random graphs [24], [25] and Watts Strogatz graphs [23]. Regular Random graph family includes graphs where all vertices have the same degree. On the other hand, Watts Strogatz graph family includes graphs where all vertices are probabilistically connected to k nearest neighbors in a ring topology. This probability is also a parameter for this type of graphs. We analyze the impact of the number of nodes, the degree, and the number of colors.

We use Python programming language [37] and NetworkX [38] graph library to generate graph instances. We implement graph generators and also heuristic algorithms by using NetworkX graph library. NetworkX library has built-in Regular Random and Watts Strogatz graph generators. In order to plot the results, we use matplotlib [39] library, which is also implemented by using Python programming

language. For each test we take the mean of 100 randomly generated instances for each scenario.

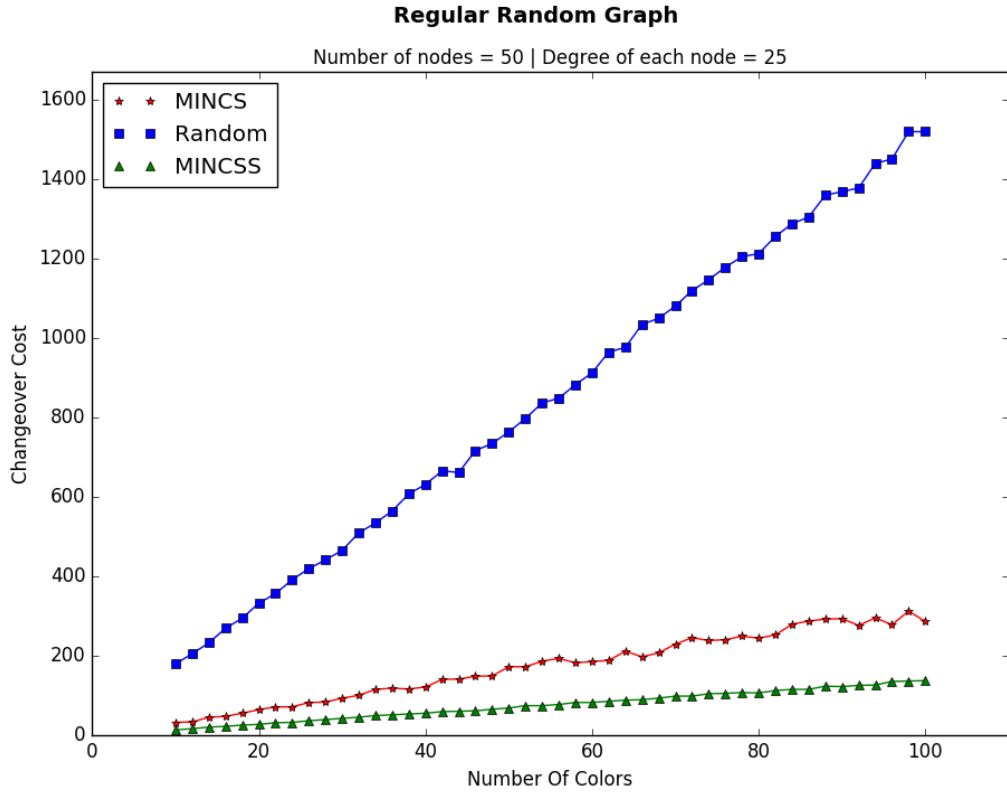


Figure 4.3: Impact of number of color changes for Regular Random graph.

Figure 4.3 shows the impact of the number of colors in Regular Random graph family. We consider a scenario where all graphs have 50 nodes, all vertices for all graph instances have 25 neighbors, and between 10 and 100 edge colors are assigned.

Total changeover cost in all algorithms increase linearly as the number of colors increases. However, MINCS and MINCSS algorithms vastly outperform Random algorithm. Moreover, MINCSS algorithm has more stable and predictable performance compared to MINCS algorithm. In addition, total changeover cost of MINCSS algorithm is non-decreasing as the number of color increase.

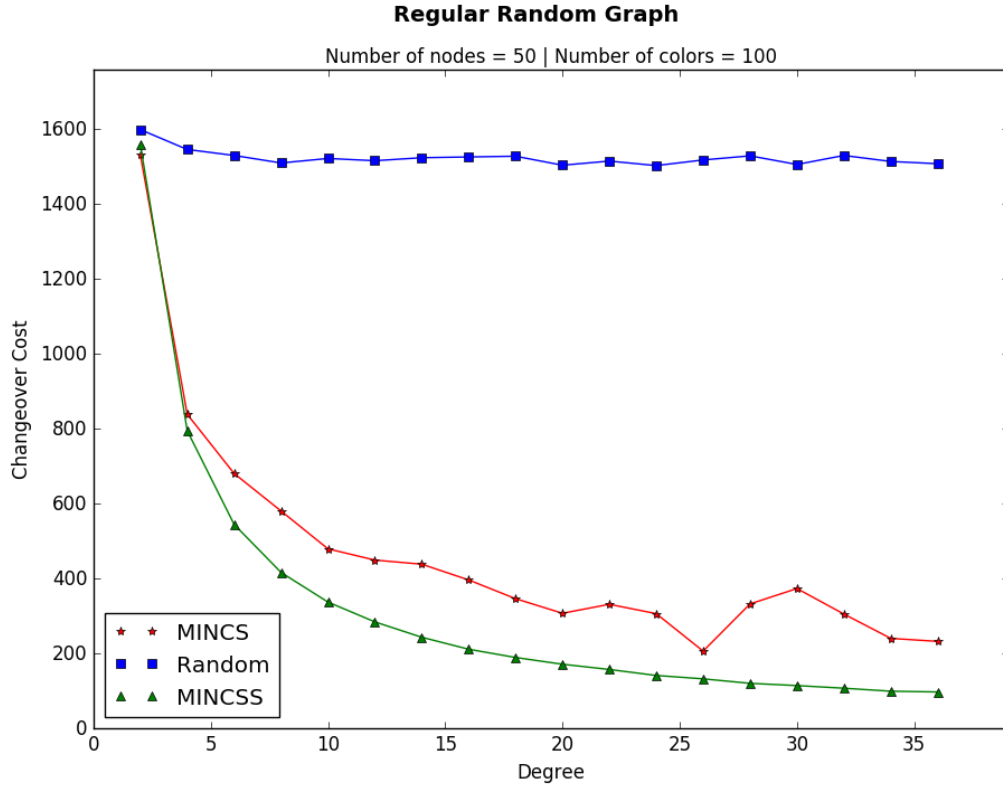


Figure 4.4: Impact of vertex degrees for Regular Random graph.

Figure 4.4 illustrates the performance of MINCS, MINCSS and Random algorithms for increasing number of degree. We choose graphs with 50 nodes, 100 colors, and node degrees ranging from 4 to 38. MINCS and MINCSS algorithms vastly outperform Random algorithm. MINCSS algorithm also outperforms the MINCS algorithm. MINCSS algorithm has non-increasing behavior. On the other hand, the performance of the MINCS algorithm is not stable since the total changeover cost fluctuates for degrees larger than 20.

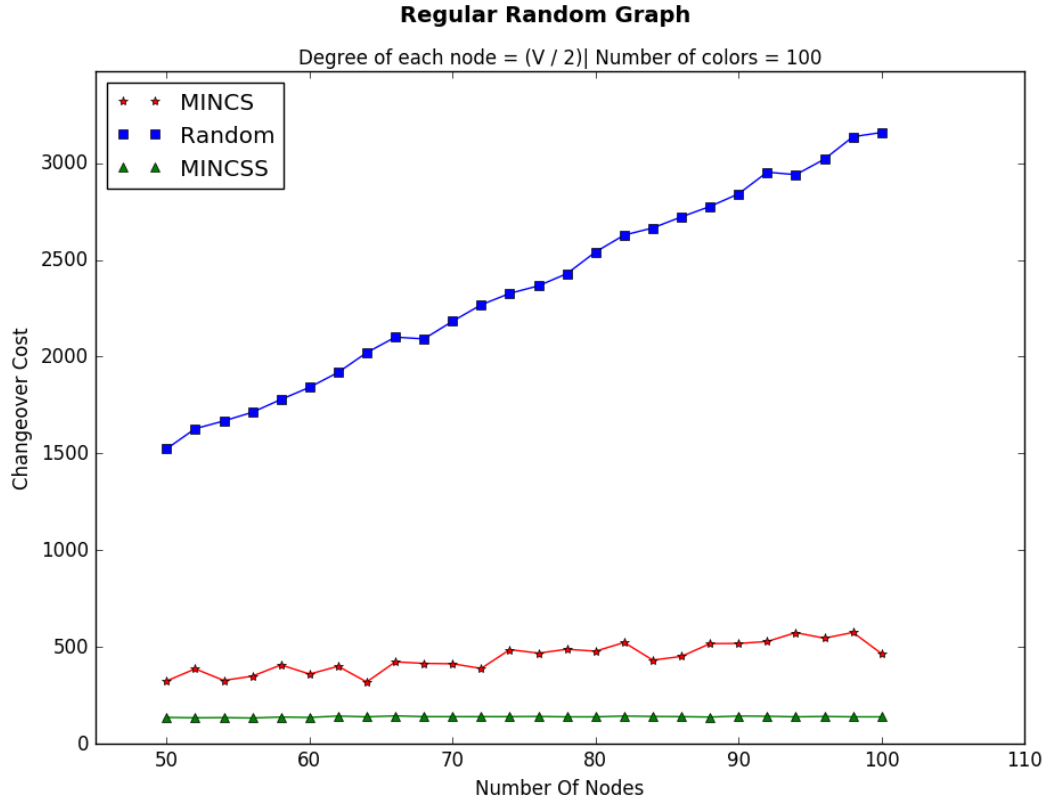


Figure 4.5: Impact of number of node changes for Regular Random graph.

Afterwards, we analyze the impact of the number of nodes in Regular Random graphs. We create graphs with 100 colors. Every vertex in the graph is connected to half of all its neighbors. The number of nodes changes between 50 and 100. The performance of the MINCSS algorithm is not affected by the change in the number of nodes and it outperforms MINCS and Random algorithms. However, Random algorithm has a linear increase for increasing number of nodes. On the other hand, MINCS algorithm has better performance compared to Random algorithm. MINCS algorithm also does not have a stable performance compared to MINCSS algorithm.

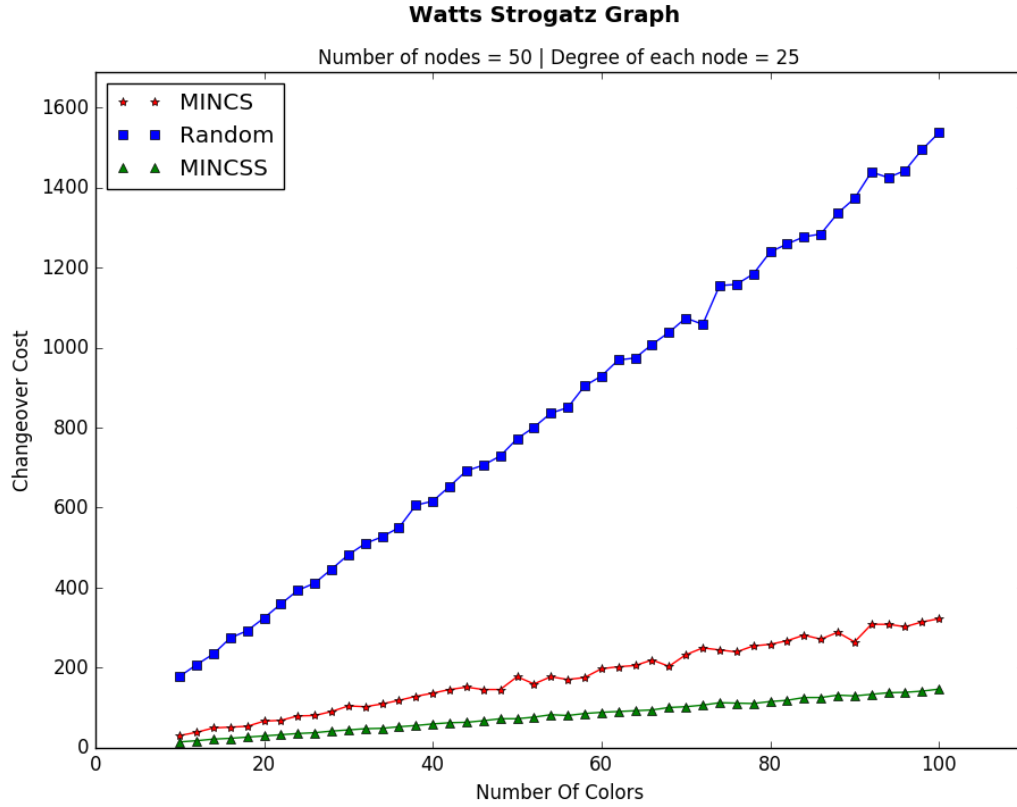


Figure 4.6: Impact of number of color changes for Watts Strogatz Graph.

Figure 4.6 shows that the impact of the number of colors for Watts Strogatz graph is almost identical to the one for Regular Random graphs. We create Watts Strogatz graphs that have 50 nodes and each node has 25 neighbors. The number of colors varies between 10 to 100. MINCSS algorithm has better and more stable performance compared to MINCS and Random algorithms. MINCS algorithm has also better performance than Random; however, unlike MINCSS, MINCS algorithm does not have a stable performance.

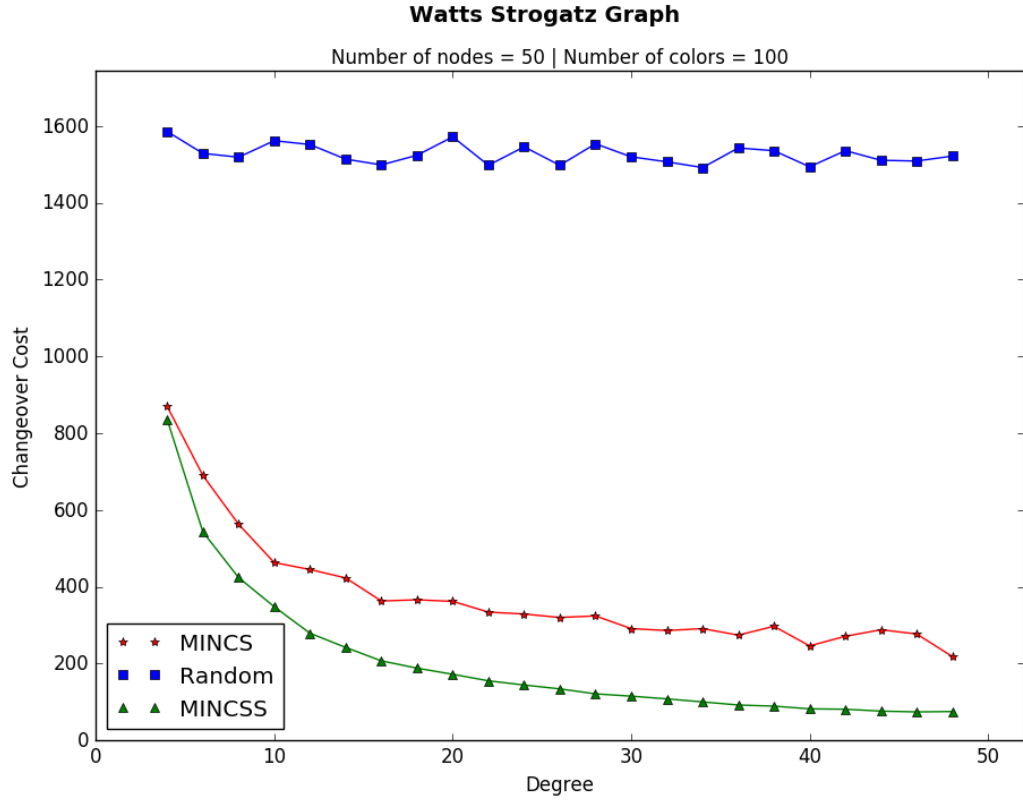


Figure 4.7: Impact of vertex degrees for Watts Strogatz Graph.

Figure 4.7 illustrates the impact of the degrees for Watts Strogatz graphs. We consider graphs with 50 nodes and 100 colors. Number of degree for each node varies between 4 and 48 for this scenario. Random algorithm continuously yields high total changeover cost. On the other hand, for lower degrees, MINCS and MINCSS algorithms result in lower changeover cost compared to Regular Random graph scenario. MINCSS algorithm has the best performance also for Watts Strogatz graph case. MINCSS algorithm has non-increasing behavior while the degree increases. Moreover, MINCS algorithm has better performance compared to Regular Random scenario.

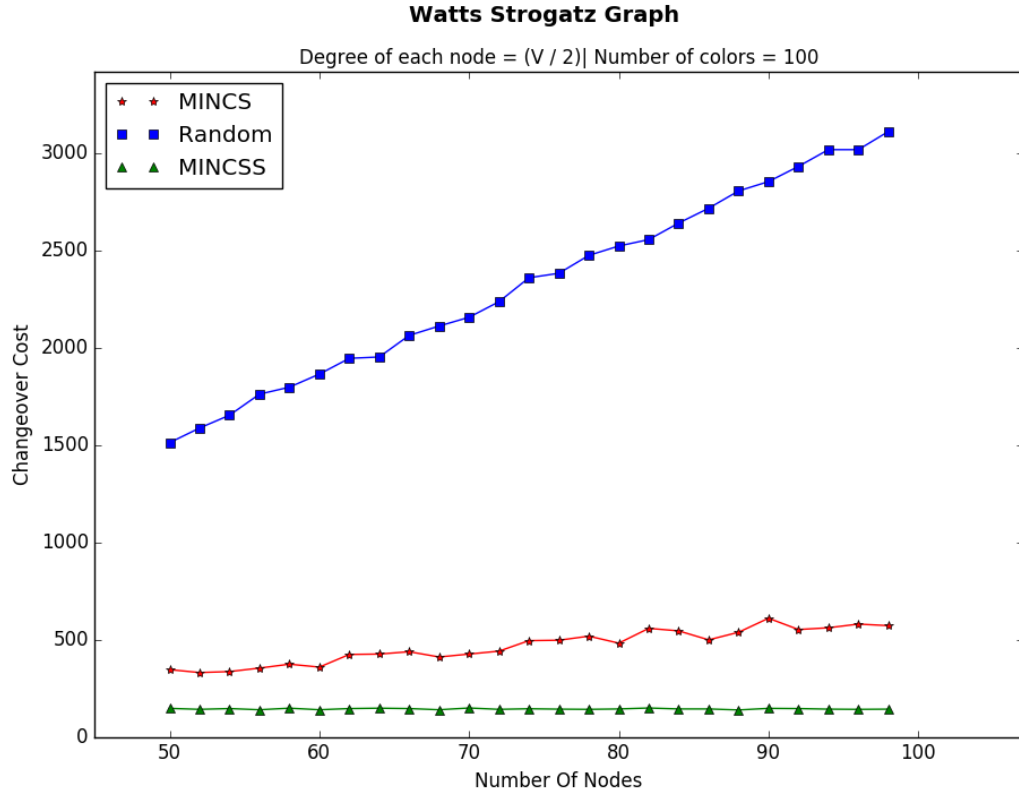


Figure 4.8: Impact of number of node changes for Watts Strogatz Graph.

Figure 4.8 shows the impact of increasing number of nodes for Watts Strogatz graph family. We create graphs with 100 colors. Each node in a graph has a connection with half of the nodes in the graph. Number of nodes changes between 50 and 100. MINCSS algorithm outperforms MINCS and Random algorithms. MINCS algorithm also has stable and non-increasing performance. In addition, MINCS algorithm performs better than Random algorithm. However, unlike MINCSS, MINCS algorithm does not have a stable performance.

5. CONCLUSION

In this thesis, we present an optimization model in the form of an integer linear programming formulation that makes frequency and time slot allocation for ad hoc cognitive radio networks with a single transceiver. Our formulation aims to minimize the energy consumption due to frequency switching by taking into account the fact that energy consumption due to frequency switching depends on the distance between the previous and current frequencies. Our simulation results indicate that our idea of taking into account the different energy consumption that occurs during switching to different frequency bands is essential for ad hoc cognitive radio networks especially when there are a high number of primary users in the vicinity of the secondary users.

We also present two heuristic algorithms for the MINCCA problem. Simulation results demonstrate that our proposed heuristic algorithms have superior performance over the random algorithm.

For future work, an ILP formulation of MINCCA problem can be introduced in order to better analyze the performance of the proposed heuristic algorithms.

REFERENCES

- [1] Akyildiz I., Lee W., Vuran M., Mohanty S., (2006), “Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey”, *Computer Networks*, 50 (13), 2127–2159.
- [2] Bayhan S., Alagoz F., (2013), “Scheduling in centralized cognitive radio networks for energy efficiency”, *IEEE Transactions on Vehicular Technology*, 62 (2), 582–595.
- [3] Gozupek D., Buhari S., Alagoz F., (2013), “A spectrum switching delay-aware scheduling algorithm for centralized cognitive radio networks”, *IEEE Transactions on Mobile Computing*, 12 (7), 1270–1280.
- [4] Gur G., Alagoz F., (2011), “Green wireless communications via cognitive dimension: an overview”, *IEEE Network Magazine*, 25 (2), 50–56.
- [5] Cheng G., Liu W., Li Y., Cheng W., (2007), “Spectrum aware on-demand routing in cognitive radio networks”, *IEEE DySPAN*.
- [6] Cheng G., Liu W., Li Y., Cheng W., (2007), “Joint on-demand routing and spectrum assignment in cognitive radio networks”, *IEEE ICC*, 6499–6503, Glasgow, UK, 24–28 June.
- [7] Filippini I., Ekici E., Cesana M., (2009), “Minimum maintenance cost routing in Cognitive Radio Networks”, *IEEE MASS*.
- [8] Chowdhury K., Felice M., (2009), “SEARCH: a routing protocol for mobile cognitive radio ad-hoc networks”, *Computer Communications*, 32 (18), 1983–1997.
- [9] Chen J., Li H., Wu J., Zhang R., (2009), “STARP: a novel routing protocol for multihop dynamic spectrum access networks”, *ACM Workshop on Mobile Internet Through Cellular Networks*.
- [10] Krishnamurthy S., Thoppian M., Venkatesan S., Prakash R., (2005), “Control channel based MAC-layer configuration, routing and situation awareness for cognitive radio networks”, *IEEE MILCOM*.
- [11] Arkoulis S., Anifantis E., Karyotis V., Papavassiliou S., Mitrou N., (2013), “On the optimal, fair and channel-aware cognitive radio network reconfiguration”, *Computer Networks*, 57 (8), 1739–1757.
- [12] Chen K., Cetin B., Peng Y., Prasad N., Wang J., Lee S., (2010), “Routing for cognitive radio networks consisting of opportunistic links”, *Wireless Communications and Mobile Computing*, 10 (4), 451–466.
- [13] C. W. Pyo, M. Hasegawa, (2007), “Minimum weight routing based on a common link control radio for cognitive wireless ad hoc networks”, *ACM International*

Conference on Wireless Communications and Mobile Computing, 399–404, Honolulu, Hawaii, USA, 12–16 August.

- [14] Zhu G., Akyildiz I., Kuo G., (2008), “Stod-rp: a spectrum-tree based on demand routing protocol for multi-hop cognitive radio networks”, IEEE Global Telecommunications Conference GLOBECOM, 1–5, New Orleans, LO, USA, 1–4 December.
- [15] Wang X., Kwon T., Choi Y., (2009), “A multipath routing and spectrum access (mrta) framework for cognitive radio systems in multi-radio mesh networks”, ACM Workshop on Cognitive Radio Networks, 55–60, Beijing, China, 21 September.
- [16] Ma H., Zheng L., Ma X., (2008), “Spectrum aware routing for multi-hop cognitive radio networks with a single transceiver”, IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications CrownCom, 1–6, Singapore, Singapore, 15–17 May.
- [17] Zeeshan M., Sattar K., Shah Z., Ullah I., (2013), “Routing and spectrum decision in single transceiver cognitive radio networks”, IEEE Vehicular Technology Conference VTC, 1–5, Las Vegas, NV, USA, 2–5 September.
- [18] Kamruzzaman S., Kim E., Jeong D., (2011), “An energy efficient qos routing protocol for cognitive radio ad hoc networks”, IEEE International Conference on Advanced Communication Technology ICACT, 344–349, Seoul, South Korea, 13–16 February.
- [19] Web 1, (2014), <http://www.ilog.com/products/cplex>, (Eriřim Tarihi: 13/02/2018)
- [20] Web 2, (2010), <https://www.fcc.gov/public/attachmatch/FCC-10-174A1.pdf>, (Eriřim Tarihi: 13/02/2018)
- [21] Gözüpek D., Shalomb M., Voloshinc A., Zaks S., (2014), “On the complexity of constructing minimum changeover cost arborescences”, 540 (541), 40–52.
- [22] Amaldi E., Galbiati G., Maffioli F., (2011), “On minimum reload cost paths, tours, and flows”, 57 (3), 254–260.
- [23] Steger A., Wormald N., (1999), “Generating random regular graphs quickly”, Probability and Computing, 8 (4), 377–396.
- [24] Kim J., Vu V. H., (2003), “Generating random regular graphs”, Proceedings of the thirty-fifth ACM symposium on Theory of computing, 213–222, San Diego, CA, USA, 9–11 June.
- [25] Watts D. J., Strogatz S. H., (1998), “Collective dynamics of small-world networks”, Nature, 393 (1), 440–442.

- [26] Badruddoza A., Namboodiri V., Jaggi N., (2014), “Does cognition come at a net energy cost in ad hoc wireless LANs?”, *Computer Communications*, 43 (1), 43–54
- [27] Namboodiri V., Gao L., (2010), “Energy-Efficient VoIP over Wireless LANs”, *IEEE Transactions on Mobile Computing*, 9 (4), 566–581.
- [28] Wirth H. C., Steffan J., (2001), “Reload cost problems: minimum diameter spanning tree”, *Discrete Appl. Math.*, 113 (1) 73–85.
- [29] Galbiati G., (2008), “The complexity of a minimum reload cost diameter problem”, *Discrete Appl. Math.*, 156 (18) 3494–3497.
- [30] Galbiati G., Gualandi S., Maffioli F., (2014), “On minimum reload cost cycle cover”, *Discrete Appl. Math.*, 164 (1), 112–120.
- [31] Galbiati G., Gualandi S., Maffioli F., (2011), “On minimum changeover cost arborescences”, 10th International Symposium on Experimental Algorithms, 112–123, Kolimpari, Greece, 5–7 May.
- [32] Gözüpek D., Shalom M., Voloshin A., Zaks S., (2014), “On the complexity of constructing minimum changeover cost arborescences”, *Theoretical Computer Science*, 540 (5), 40–52.
- [33] Cygan M., Fomin F. V., Kowalik L., Lokshtanov D., Marx D., Pilipczuk M., Saurabh S., (2015), “Parameterized Algorithms”, Springer.
- [34] Downey R. G., Fellows M. R., (2013), “Fundamentals of Parameterized Complexity”, *Texts in Computer Science*, Springer.
- [35] Flum J., Grohe M., (2006), “Parameterized Complexity Theory”, *Texts in Theoretical Computer Science*, Springer.
- [36] Niedermeier R., (2006), “Invitation to Fixed-Parameter Algorithms”, Oxford University Press.
- [37] Web 3, (2001), <https://www.python.org/>, (Erişim Tarihi: 13/02/2018)
- [38] Web 3, (2014), <https://networkx.github.io/>, (Erişim Tarihi: 13/02/2018)
- [39] Web 3, (2012ss), <https://matplotlib.org/>, (Erişim Tarihi: 13/02/2018)

BIOGRAPHY

My name is Mustafa Çamurlı. I was born in Rize, Turkey. My address is Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey and my electronic mail address is mcamurli@gtu.edu.tr. I completed bachelor degree in Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey. I completed master degree in Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey. My publication on the thesis is M.Çamurlı, D.Gözüpek, "Channel switching cost-aware resource allocation for multi-hop cognitive radio networks with a single transceiver", International Conference on Ad hoc Networks (ADHOCNETS), Rhodes, Greece, 2014.