OPTIMIZATION OF CHANNEL ASSIGNMENT IN MOBILE COMMUNICATION USING TABU SEARCH

NUR ADIBAH BINTI AMAT ALI

A Report Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Electrical Engineering (Power Industry)

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

DECLARATION

I declare that this thesis entitle “Optimization of Channel Assignment in Mobile Communication using Tabu Search” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……………………………………….

Name : NUR ADIBAH BINTI AMAT ALI

Date : ……………………………………….

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electrical Engineering (Power Industry).

Signature	:	…………………………………
Supervisor Name	:	DR. LOH SER LEE
Date	:	…………………………………

DEDICATION

This project is dedicate to my beloved family especially my father, Amat Ali bin Sapan, my mother, Asparala binti Tarmin, and my siblings. Thank you for all your support and encouragement to me.

ABSTRACT

Mobile communication is a process of exchanging information by connecting between two parties through a wireless network. It is being connected through existing channels in a particular coverage region. Practically, mobile communication system involves a large coverage area that serves a large geographical area. However, there has only limited channels to accommodate all the users. With the increasing number of mobile users recently and the existing channels are limited, it leads to channel assignment problem (CAP). In order to prevent the channel assignment problem from getting worse, an initiative is taken to reuse the existing channels in an efficient way with the minimum interference occur in the channels assignment. This helps to optimize the usage of the channel assignment in mobile communication. Basically the channel is reused in such a way the interference between the assigned channels is minimized. Tabu Search (TS) technique is implemented to solve the CAP by searching for the global optimum solution with the minimum interference cost value. Interference cost value represents the severeness of the interference occurred among the channels assigned. The optimization process is implemented by first determine the initial solution. Then, the chosen initial solution is used to generate the neighbourhood solutions. The reallocated channels are being analysed the value of the penalty cost based on the penalty cost function. The process keeps repeating until the termination criterion is satisfied. Among the available solutions, the solution with lowest penalty cost function is chosen as the final solution. The optimization of the channels assignment is a process of reducing total channels required as the solution is improved. An analytical analysis is carried out to investigate the effect of demand calls and the number of available channels on the cost value. The results are being compared by varying the demand required in the channel assignment with different number of channels used.

ACKNOWLEDGEMENTS

Thank you to Allah SWT, with His willing for giving me the opportunity to complete this project about Final Year Project properly. My supervisor and I worked together, hand in hand to make this project completed in success. So, I am really grateful have chances to cooperate and getting helps from everyone.

First and foremost, I would like to thank and gives my gratitude to my supervisor, Dr. Loh Ser Lee, for all her encouragement, support, understanding and assistance by making sure that my entire project completed successfully. On top of that, thank you for the constant reminders and much needed motivation until I had managed to present this report and submitted. Thank you also for your time and energy that you had spent for me.

Moreover, I would like to thanks my parents for always be there for me and gives motivation to me. Thanks also for the help in financial and inspiration they extended. Last but not least, I would also like to thanks to my friends for sharing information's, assisting me in this project and for all their opinions so that I managed to complete this project successfully. Last but not least, thank you very much to everyone who are involved directly or indirectly in this project.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

RF	-	Radio Frequency
MIN	-	Mobile Identification Number
MSC	-	Mobile Switching Centre
BS	-	Base Station
CAP	-	Channel Assignment Problem
SCA	-	Static Channel Assignment
DCA	-	Dynamic Channel Assignment
HCA	-	Hybrid Channel Assignment
U.S	-	United States
AM	-	Amplitude Modulation
FM	-	Frequency Modulation
GA	-	Genetic Algorithms
NNs	-	Neural Networks
SA	-	Simulated Annealing
TS	-	Tabu Search
AI	-	Artificial Intelligence
FDMA		Frequency-Division Multiple Access
TDMA		Time-Division Multiple Access

CHAPTER 1

INTRODUCTION

1.1 Introduction

In globalization era nowadays, communication is one of the interesting parts for exchanging information among people. Mobile device is widely used for a communication network system. Communication process is propagating through a radio frequency medium. Radio Frequency (RF) is a form of electromagnetic wave frequencies which is in a range extending from around 3 kHz to 300 GHz; including the frequencies that are normally used for communications or radar signals.

Each mobile has its own Mobile Identification Number (MIN). In processing a call request, it involves two main components which are called Mobile Switching Centre (MSC) and Base Station (BS). Generally, wireless networks communication are organised in geographical cells. Each of the cells is consists of a few existing channels that is being controlled by a Base Station (BS) [3]. Meanwhile for the whole BS, it is controlled by MSC. Then, for a user that is making a call request to MSC, it is a process communication between the caller and the receiver.

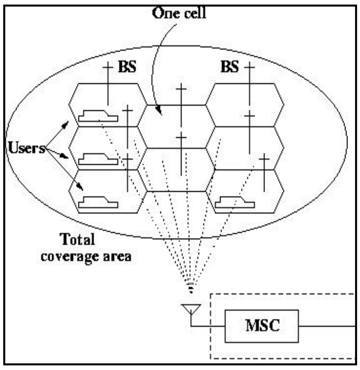

Figure 1.1: Each of base station is represented as one cell [1]

Then, the MIN of receiver will be sent to MSC. The MSC will process the MIN by sending the MIN to all BS which under control of it to find out the location of the call receiver. After that, the respective BS will send signal back to MSC to report that the mobile is within its area of cells. Next, the MSC will order the BS to access the unused voice channel pair to the pair of caller receiver to communicate [1].

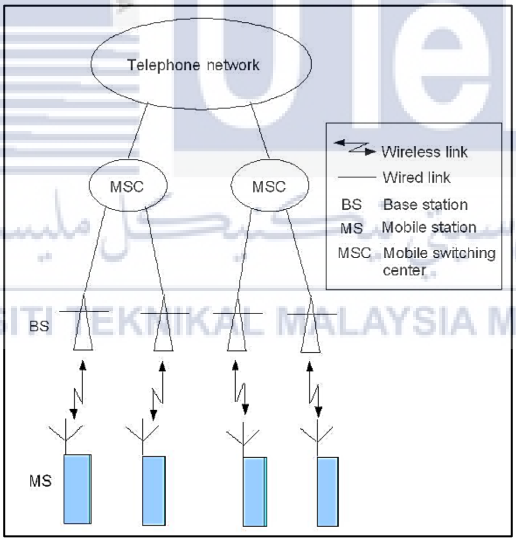

Figure 1.2: Structure of basic mobile communication [1]

Since the number of mobile users is increasing, therefore the need of the channel is increased. Mobile communication needs unused voice channel pair. Basically in communication system, there will be source and destination for the information to be transferred. The information is being transferred from source by using channel to the destination. Channel is a flow path for the information to be transferred. The channel has different range of frequency. Each frequency range has its own purpose of usage.

Therefore, as the amount of consumer increasing recently, the need for the limited existing channel is also increased.

In order to optimize the usage of channel, the strategy of channel reused is needed. Unfortunately, the strategy may lead to interference along the communication process. Interference is categorized into three types. Firstly, co-channel interference is when a communication process is occurred in the same channel is being assigned for a few cells from different BS. Secondly, adjacent-channel interference is when channel is in adjacent positions in radio spectrum is being used in a cell which in adjacent positions too, and thirdly co-site interference is when the adjacent of channel is being assigned in the same cell and they are not separated away from one to another by the smallest distance of separation in frequency range [4].

There are three models in channel assignment problem (CAP) which are Static Channel Assignment (SCA), Dynamic Channel Assignment (DCA) and Hybrid Channel Assignment (HCA). SCA is known as channel assignment that is consists of a fixed number of unused voice channels assigned in each of cell for communication process. Then for DCA is channel assignment that is temporarily being assigned in a cell along the duration of a call is being made. The channel of the call used will be changed based on the nearest location of the BS. The probability of a call is being rejected by using DCA model is reduced. For this project, SCA is used to assign the channels in cells based on the demand.

1.2 Motivation

Mobile device is widely used nowadays. By the development of technologies in mobile is increasing, wireless mobile communication is commonly used. It was caused the users growth rate reached 50 percents per year. It shows that users is increasing day by day which from 8000 users in year of 2000 until it is reached 35000 users by the end of year 2007 [1]. However, due to the limited available channels in mobile communication, it leads to plenty of problems such as call rejection or noise sound effect. Interference between channels in a call occurs when the frequency constraints are not fulfilled in the channel assignment. Hence the optimization of channel assignment is important in order to provide a good quality service to the subscribed mobile users. Since mid-1990s, many methods have been proposed to solve the channel assignment problem [4]. Researches on channel assignment were started to be carried out as the communication problem appeared. Tabu search is a popular optimization technique that can be used to produce an optimal solution of the interference to avoid from any rejected calls due to limited existing channels assignment.

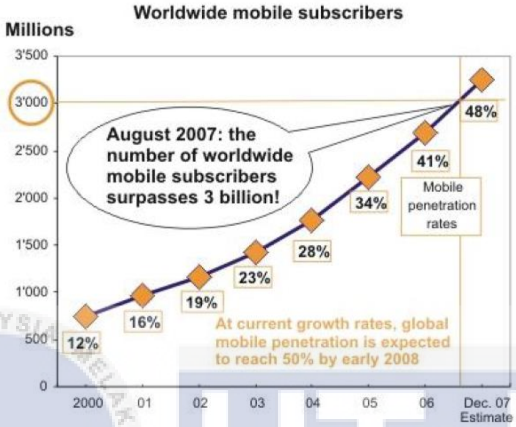

Figure 1.3: Statistics of worldwide mobile subscribers from year 2000 until 2007 [1]

1.3 Problem Statement

Practically a region is divided into N cells and there are M available channels to serve the calls. In a cell, it consists of a certain number of channels. In order to make a successful call, a channel is needed per call. The amount of available channels at a particular time depends on the usage of the line communication. If the calls being made in a time is more than the available channels, the call without getting assigned to a channel will be rejected due to no available channels in that cell. Hence, the channel assignment to call in every cell need to be optimized based on the demand from experience. Meanwhile the amount of interference occurred in the channel during a call need to be reduced.

On November 2015, United States (U.S) population statistics shown that 87 percent of people those ages 18 years old and above owned mobile phone. It stated that, an increasing percent of people owns smartphone: the survey shown that 77 percent smartphone ownership rate among those with mobile phones is a substantial increase over the 71 percent rate reported in 2014, 61 percent in 2013, 52 percent rate in 2012, and 44 percent rate in 2011 [2].

Based on the past experience, the demand in cell j is recorded and represented by one-dimensional matrix channel demand, D_j. The constraint of the channel assigned between cell i and j is the minimum separation of the frequency, $C_{i,j}$ called non-interference constraint. It is given in a four-dimensional matrix C of row, i and column, j. The set of binary, $X_{j,k}$ shows that channel k is assigned to cell j if $X_{j,k}$ equals to value of one. Otherwise, it will be zero value.

To achieve interference-free assignment, the constraint of $C_{i,j}$ must be satisfied. However, slight interference is allowed to increase the availability of channels. If the minimum separation constraint is violated, an interference will occur and a penalty value will be imposed by cost tensor, $P_{j,i,m}$ where m is distance between channels assigned to cells i and j. The value of the cost tensor shows the degree of the interference occurred.

The penalty cost function shows the severeness of the interference occurred among the assigned channel. In this project the penalty cost function value is called cost value. The problem formulation for the static channel assignment models is stated as followed [5].

The penalty cost function is shown as follows:

$$F(X) = \sum_{j=1}^{N}\sum_{k=1}^{M} X_{j,k} \sum_{i=j}^{N}\sum_{l=k+1}^{M} P_{j,i,(\,|k-l|\,+1\,)} X_{i,l}$$

where $m = |k - l|$.

Next, a set of binary variables is illustrated the assigned channel by:

$$X_{j,k} = \begin{cases} 1 & , \text{ if cell } k \text{ is assign to channel } j \\ 0 & , \text{ otherwise} \end{cases}$$

where $j = 1, 2, 3, \ldots, N$; $k = 1, 2, 3, \ldots, M$.

The relationship between set of binary variables, $X_{j,k}$ and the demand requirement of the channels in each cells, D is shown as below expression:

$$\sum_{k=1}^{M} X_{j,k} = D_j \quad , \quad j = 1,2,3,\ldots,N \; ; \quad k = 1,2,3,\ldots,M$$

The cost tensor, P is stated as below:

$$P_{j,i,m} = \max \{ C_{j,i} - m , 0 \}$$

where $C_{j,i}$ is a matrix C with row j and column i.

A penalty cost or cost value, $F(X)$ is being charged due the interference of the channel occurred. It can be calculated based on the penalty cost function above. The higher the amount of the penalty cost, it shows that, the stronger the interference occurred of the channel among the cell. The penalty cost will be decreased when the channel k and l are far enough from each other. Then, interference will be diminished. The formula purpose is to minimize the total cost of penalty occurred.

1.4 Objectives

There are two main objectives associated with this project, which are:

a) To optimize the channel assignment in mobile communication by minimizing the interference cost value.
b) To implement tabu search in channel assignment problem.
c) To investigate the effect of demand calls and number of available channels on the cost value.

1.5 Scope

Channel assignment in this project targets the application of mobile communication where calls are being made through channels in a cell. The model used is a Static Channel Assignment (SCA) where channels are assigned based on the demands in the region. In this project, tabu search technique is being used. The coding of tabu search is produced by using MATLAB software. A penalty cost function is used to show the severeness of channels interference. As the demand recently in communication network increased, the demand of the channel assignment is varied and the affect towards the interference occurred is analysed.

1.6 Significant of Study

The main concern of this project is the optimal allocation of limited channels in mobile communication. Lower interference gives higher communication quality to mobile users. An optimal solution where the value of channel interference is minimized and at the same time the usage of available channels are maximized, will lead to efficient communication between users.

1.7 Project Outcomes

This project on optimization of channel assignment problem will be solved by running simulation to minimize interference occurred among the existing channels assign. The higher the value of the interference occurred, the lower the quality of the call. In order to get the optimal solution, the process of creating immediate neighbourhood of the current best solution in the hope of finding an improved solution is continue until a prefixed attempt limit is reached. An analysis will be carried out to investigate the effect of demand calls and number of available channels on the cost value.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In this section, literature review on channel assignment problem from the first problem appeared and the development of heuristic techniques are presented. Heuristic techniques are popular in optimization problem and it has been applied by researchers in various types of combinatorial optimization problem, including channel assignment problem.

2.2 Literature Review

The development of mobile is basically from the improvement of the first wireless telephone system. It has been introduced and used in the year of 1877s. Communication medium is started by using Amplitude Modulation (AM) of radio frequency in early 1934. After that, the development of the frequency is improved by using Frequency Modulation (FM) [1]. As the demand of new system is need for better mobile radio network, then FM proved it gives more quality and efficient system compared to AM. It is developed by Edwin Armstrong whose witness many improvement of communication system at that time [1].

Twelve years later, mobile device has been introduced and published. At the beginning, not many people own it and make it as one of their needed [1]. Then, the awareness of people on importance of having mobile as one of the important medium for communication between two parties is increasing, the very first channel assignment problem is occurred in early 1960s [4]. As the users of mobile increasing, the available frequency or channels is limited. It has been found that the first network shows insufficient of the frequency usage when wireless service is being used.

A survey is made to have a clear view on the smartphone usage among U.S people. It is conducted on 2015, which the survey suggested that mobile phone ownership varies slightly by race and ethnicity. It shows that as time goes by from 2011 until 2015, the smartphone usage among the race/ethnicity is increasing rapidly. Adoption of smartphones in the Mobile Survey in 2015 varies in a somewhat more pronounced way: 82 percent of Hispanic mobile phone users have a smartphone, compared to 74 percent of non-Hispanic whites and 76 percent of non-Hispanic blacks. Among those with a mobile phone, smartphone ownership is also higher (88 percent) for the "Other, non- Hispanic" group, which includes respondents who report their race as Asian, American Indian or Alaskan Native, Hawaiian, or Pacific Islander[2].

Race/ethnicity	Smartphone usage				
	2011	2012	2013	2014	2015
White, non-Hispanic	41	50	57	68	74
Black, non-Hispanic	47	54	63	66	76
Other, non-Hispanic	45	54	76	83	88
Hispanic	55	60	72	82	82
2+ races, non-Hispanic	43	59	64	65	79
Total	44	52	61	71	77
Number of respondents	2,002	2,291	2,341	2,603	2,244

Note: Among respondents with a mobile phone.

Figure 2.1: Smartphone usage by race/ethnicity [2]

Until today, various heuristic techniques are being used to solve variety of channel assignment crisis occurred. It is being used for the SCA or DCA models by implement it in Genetic Algorithms (GA) [6], Neural Networks (NNs), Simulated Annealing (SA) [7] [8], Tabu Search (TS) [4] [8] [9], and Artificial Intelligence (AI). The main objective of the CAP is to assign the existing channels to call so that the usage of channels is being maximized, since the channels are limited. Then, interference between the frequencies (channels) is needed to be optimized to minimum value. At the same time, the number of frequencies on each of base station must be large enough to satisfy the demand in the cell.

Based on the previous research, there are many algorithms and problem formulations can be applied in many applications problem. Heuristic techniques solve problems by giving a close-optimal solution at a relevant computational cost. It is non-algorithmic methods that are applied to algorithmically complex channel assignment problems to generate efficient solutions [3].

Besides, a research is being carried out on cellular radio of channel assignment to solve CAP by using simulated annealing technique since the previous project is based on the graph coloring algorithms. The research shows the design neighbourhood structure of channel allocation of cellular radio gives positive impact on quality and efficiency of the channel assignment [6].

Theoretically, simulated annealing is used to solve problem that involve discrete optimization problem examining the equation and the state of systems which is involved temperature. It is one of method that has ability to jump out of local minima to search for a better solution. It is applied on conditions of cooling schedule of the system which involved initial temperature and it will be solved until get the slow cooling process which the temperature is decreased [6].

The optimization problem as a cost function is representing the temperature of the system. It is a solution method that has been developed for obtaining approximate solutions to minimize the temperature as the system requirement [10]. By simulated annealing approach, it guarantees the optimal asymptotically, but the rate of cooling process is rather slow when the process of finding optimum temperature is required [10]. Meanwhile based on the research of Albert Y. Zomaya, eventhough it is solved by using simulated annealing, a problem is arise at the end of the CAP of mobile computing. It cannot prove the quality service is good and optimum allocation schemes, since its main constraint is temperature [11].

Next, a project is being done for a new strategy by applying GA method to the CAP. In that case, rather than channel assignment problem involved temperature, it is being analysed based on system that using frequency-division multiple access (FDMA) without any time-division multiple access (TDMA). It was focused on the optimum call list not necessarily optimize the frequency assignment. Their proposed algorithm produced zero interference and able to achieve local minima during the optimization process for all the problem [6]. Eventhough it is focused on optimization of channel assignment, the usage of the frequency is not efficient. It shows that genetic algorithm only solve the channel assignment problem by allowing the co-site interference conditions.

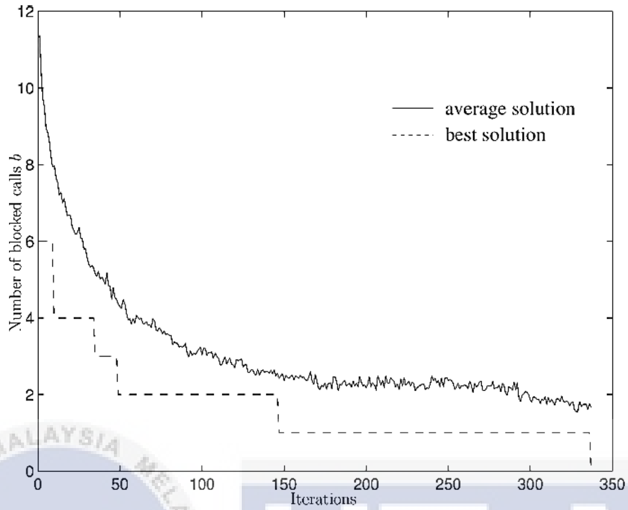

Figure 2.2: Number of blocked calls is reduced as the solution is being improved [7]

Since then, a tabu search techniques is suitable to be used in order to make the channel assignment used efficiently by allowing of any interference occurred, but for searching better solution interference free is preferable. Basically, tabu search was first explored in 1977s [5]. It is consisted of classical tabu search and reactive tabu search. Normally, classical tabu search is used to solve channel assignment problem. It is based on list of neighbourhood data with a set of critical and corresponding components referred to the channels requirement [12].

Tabu search technique is started by having an initial solution to solve the CAP. Then, arrange neighbourhood by tabu move to allocate the channel available to indicate best initial solution which is having the optimum of interference or free interference. In a better way, it is determined by avoiding from any interference occurred [3]. Tabu search is an easy approach and simple technique which can produce a better result. The research paper on tabu search started being produced on tabu search algorithm in channel assignment problem [13].

Research previously proved that tabu search have a better speed in order to generate the best solution compared to simulated annealing and genetic algorithm [14]. It is found that, tabu search technique is much better when the neighbouring allocate is included in long-term memory. Tabu search is shown to give a better result and is proven to be the most effective algorithm for CAP as it produces a better average interference value based on a few problems that is being tested [4].

Recently, a research was conducted to improve tabu search algorithm [15]. Furthermore, eventhough tabu search cannot efficiently solve the problem involving topic optima, this technique can be repeated to solve larger problem in a short time or less iterations [8]. Next, channel assignment problem in mobile communication has been improved by using reactive tabu search techniques.

Basically it has been improved based on classical tabu search which is the repeating process to search for better solution to the problem is clearly state maximum number of iterations required based on the formula of tabu tenure. The research found that, tabu tenures will be greater than the set of maximum number of iteration. It shows that, once the channel assignment is stored as tabu list, the next solution that have the same channel assign will never be considered again [9]. Then, tabu tenure is much clear to be used as reference in term of classical approach, but for better approximation of iteration number reactive approach is much better.

Based on previous research, tabu search technique will be reproduced and validated by doing analytical analysis on the relationship of number available channels and the cost value will be conducted. The repeating process to get the optimum value of interference will be needed by targeting interference free. Besides that, the usage of the existing channel will be increased by using tabu search compared to genetic algorithm.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Overview

In reality, the number of mobile users is continuously increases due to the lifestyle that required communication between each other. As time goes by, the demands required for channel assignment is increased. It is leads to the channel assignment problem occurred. This optimization channel assignment in mobile communication project is solved based on the channel assignment problem occurred nowadays. The project is solved by using tabu search method.

3.2 Implementation of Tabu Search in Channel Assignment Problem

In order to solve the channel assignment problem, tabu search algorithm is coded using MATLAB's software to assign the channels. Then, an initial solution is created based on the parameters required as shown in Section 1.3. In this project, the data of 4 cells and 13 channels are being used. Initially, the number of cell, N in the network is set to be 4 cells and the demand requirement of the channels in each cells, D is assigned one channel each at cell 1, 2 and 3, meanwhile three channels assigned in the cell 4.

In total, the demand requirement of the channel assignment, D is 6 channels. The data for a set of binary variable $X_{j,k}$ is inserted randomly. The set of binary variable represents the assigned channels in each cell. In order to produce an initial solution, the channel is assigned randomly according to the required number of channels in each of the cells. The compatibility matrix of the minimum separation of frequency between cell i and cell j is set. The diagonal terms, $C_{i,i} = 6$ indicate any two channels assigned in the same cell must be at least six frequencies apart. The data set of channel assignment problem is shown as follows:

Number of cells, N = 4

Number of available channels, M = 13

Demand of the channels or number of calls in each of the cells, D_j = [1, 1, 1, 3]

The minimum frequency separation constraint between the cells i and j for zero interference, $C_{i,j} = \begin{bmatrix} 6 & 5 & 0 & 0 \\ 5 & 6 & 0 & 1 \\ 0 & 0 & 6 & 2 \\ 0 & 1 & 2 & 6 \end{bmatrix}$

```
%Optimization of channel assignment problem using Tabu Search technique
N=4; M=13; D=6; %D=[1,1,1,3]
X=zeros(N,M);
%initial solution of set binary variable
X(1,1)=1; X(2,2)=1; X(3,3)=1; X(4,1)=1; X(4,4)=1; X(4,13)=1;
C=[6 5 0 0; 5 6 0 1; 0 0 6 2; 0 1 2 6];
```

Figure 3.1: Data set in MATLAB's code

Then, the set of binary variable is assigned as shown below:

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	●												
2		●											
3			●										
4	●			●									●

Figure 3.2: The initial allocation of assigned channel for the channel assignment problem

As the iteration runs, the cost value is computed and recorded to be observed. The coding of cost value is written in a function of ‘costvalue’ function where the process of calling back function is needed in order to evaluate the cost value. The coding of the channel assignment problem in MATLAB’s is evaluated the cost value as shown in Figure 3.3.

```
function costvalue = COSTVALUE(X,N,M,D) %function to calculate cost value
    C=[6 5 0 0; 5 6 0 1; 0 0 6 2; 0 1 2 6]; %non-interference constraint
    sum2=0; %initialize sum2
    sum3=0; %initialize sum3

    %looping process to calculate the total of interference cost value
    for j=1:N
        for k=1:M
            sum1=0;
            for i=1:N
                for l=1:M
                    m=abs(k-l); %distance between channel k and l
                    P= max((C(j,i)-m),0); %cost tensor
                    %sum1 current+previous interference
                    sum1 = sum1 + (P*X(i,l));
                end;
            end
            %sum2 current+previous interference
            sum2 = sum2 + (X(j,k)*sum1);
        end
    end
    %total interference without interference with itself
    sum3=sum2-D*C(1,1);
    %cost value of interference from the assigned channel
    costvalue = sum3/2;
end
```

Figure 3.3: Coding of computation for cost value

Next, the value of the cost value is passed to the coding that analyse it by using tabu search technique. The important initialized parameters are the current and best binary set and their respective cost value, iteration for number of neighbourhood for each of the cells and the forbidden iteration for forbidden strategy, in order to make sure the optimization process is run properly.

```
%%start to analyze objective function to get the costvalue
Current_CV = 999;    %start the current cost value high value
Best_CV = 999;       %start the best cost value high value
iteration = 3;       %this set data will be inner looping for 3 times
K = 50;              %stopping criteria
p=1;  q=1;
Current_X=X         %let binary setX as initialize for currentX
Current_CV = COSTVALUE(X,N,M,D)%calling COSTVALUE function
temp_X=X;           %set the assigning channel for initial solution
forbidden_iter=5;
Tabu_list=zeros(N,M);
index=0;
```

Figure 3.4: Parameter setup for tabu search algorithm

Then, by runing the algorithm coding on MATLAB's software, a pool of candidate solutions stemmed from the initial solution. The process of creating the first solution from the initial solution is by reallocate any of one assigned channel. It is a process of assigning channels arbitrarily based on the demands needed in each cell. Then process of reallocation of channel is randomly done by replacing a channel used from the current channel with an unused channel in the same cell at a time.

```
%start to analyze objective function to get the best(min) costvalue
Current_CV = 999;   %start the current cost value high value
Best_CV = 999;      %start the best cost value high value
iteration = 3;      %this set data will be inner looping for 3 times
K = 50;             %stopping criteria
p=1;  q=1;
Current_X=X        %let binary setX as initialize for currentX
Current_CV = COSTVALUE(X,N,M,D)%calling COSTVALUE function from other
file
temp_X=X; %set the assigning channel from the best(min) initial solution
forbidden_iter=4;
Tabu_list=zeros(N,M);
index=0;

%starting to create neighbourhood
for r=1:K  %process assigned channel is repeatedly up to K times
    for s=1:iteration %swap between two channel allocate
        for i=1:N
            k=randi(M); %randomly move(assign) channel to unused channels
            for j=k:M
                if (X(i,j)==1)
                    rand_1=j;
                    break
                else
                    for j=1:k
                        if (X(i,j)==1)
                            rand_1=j;
                            index=1;
                            break
                        end
                    end
                    if(index==1)
                        index=0;
                        break
                    end
                end
            end
            for j=1:999
                k=randi(M);
                %avoid assign channel at the used channels
                if (k~=rand_1 && Tabu_list(i,k)<=0 && X(i,k)==0)
                    rand_2=k;
                    break
                end
            end
            temp_X(i,rand_1)=0; %current channel is being moved(swap)
            temp_X(i,rand_2)=1; %unused channels is being assigned(swap)

            CV=COSTVALUE(temp_X,N,M,D);
            temp_X=temp_X;
            if (CV < Current_CV)%the best solution is chosen
                Current_CV = CV;%record current cost value
                Current_X = temp_X;  %record location of assigned channel
                p = i;
```

```
                q = rand_1;
            end
            temp_X=X;  %the setX is being reset to original setX
        end

    end
    Current_CV
    Current_X

    X=Current_X; %update the current solution
    %tabu list strategy (avoid repeating assign the same location)
    Tabu_list(p,q)=forbidden_iter; %%forbidden strategy
    Tabu_list=Tabu_list-1         %freeing strategy (iteration decrease)

    if (Current_CV<Best_CV)   %%verify the best solution
        Best_CV=Current_CV
        Best_X=Current_X
    end
    Current_CV=999;
end
```
Figure 3.5: Coding algorithm of Tabu Search method

As the neighbourhood is created, an assigned channel is randomly chose and being free, and an unused channel will be assigned as a replacement. Meanwhile, the freed channel is prohibited to be assigned in the next few iteration and is recorded in tabu list. In tabu search technique, each movement of the channel is being recorded by using tabu characteristic which is called as tabu list. The channel used in tabu list is forbidden to be reused in a prefixed number of iterations. This aims to avoid creating the same set of binary in the neighbourhood. In other words, all binary set varies from each other in the neighbourhood.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1											●		
2		●											
3			●										
4	●			●									●

Figure 3.6 : The first assigned channel for channel assignment problem

In the process of creating neighbourhood, it produces a pool of solutions in which one call is reassigned with a new channel from the current solution. The process keeps repeating as long as the stopping criterion is not reached. Neighbourhood is created by reallocation of assigned channel in a cell. As a solution is generated, its corresponding cost value is computed using the penalty cost function. The cost value represents the interference occurred among the assigned channels.

After the initial solution is generated, the channel assignment problem is started to be processed based on tabu search technique and new neighbourhood is being generated. Then, in the pool of solutions in the neighbourhood, the solution with the lowest cost value will be chosen and known as current solution. If the cost value of current solution is lower than the best stored value, the current solution will be updated as the best solution.

Whenever current solution is updated, the previously visited solution is being recorded in memory called tabu list. The moves in tabu list are forbidden moves. This is needed to prevent the production of same solution in neighbourhood. The moves are freeing based on the freeing strategy. However, according to the aspiration criterion, if the tabu moves has a sufficiently attractive evaluation where it may result in the lowest cost value than any of the visited solution, that particular tabu moves will be free from tabu list.

Since the interference contributes to the cost penalty value, the coding algorithm is created for the problem formulation used. In this case, termination criteria used is based on the number of iterations reaches the setting of maximum number of iteration. The process of searching for the best solution will stop when the termination criterion is satisfied and hence the final solution is found. The process implementation of tabu search in this channels assignment is illustrated in a flowchart as shown in Figure 3.7.

In order to investigate the relation between the cost values of interference with the number of available channels, the data is varied based on the available channels, M and the demand requirement, D in each of the calls. Data set is used where the demand is varied in 5, 10, 15, 20 and 25 for the channel assignment problem. The two dimensional compatibility matrix used is shown below and the parameters of channel assignment problem for data set, is set as follows:

Number of cells, N = 4

Number of available channels, M = 25

Size demand of the channels or number of calls required, D = 5, 10, 15, 20, 25

The minimum frequency separation constraint between the cells i and j for zero interference, $C_{i,j}$.

$$C_{i,j} = \begin{bmatrix}
6 & 5 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 6 & 5 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 6 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 6 & 5 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 5 & 6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 6 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 5 & 6 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 5 & 6 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 5 & 6 & 5 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 5 & 6 & 5 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 5 & 6 & 5 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 5 & 6 & 5 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 5 & 6 & 5 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 & 6 & 5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 5 & 6
\end{bmatrix}$$

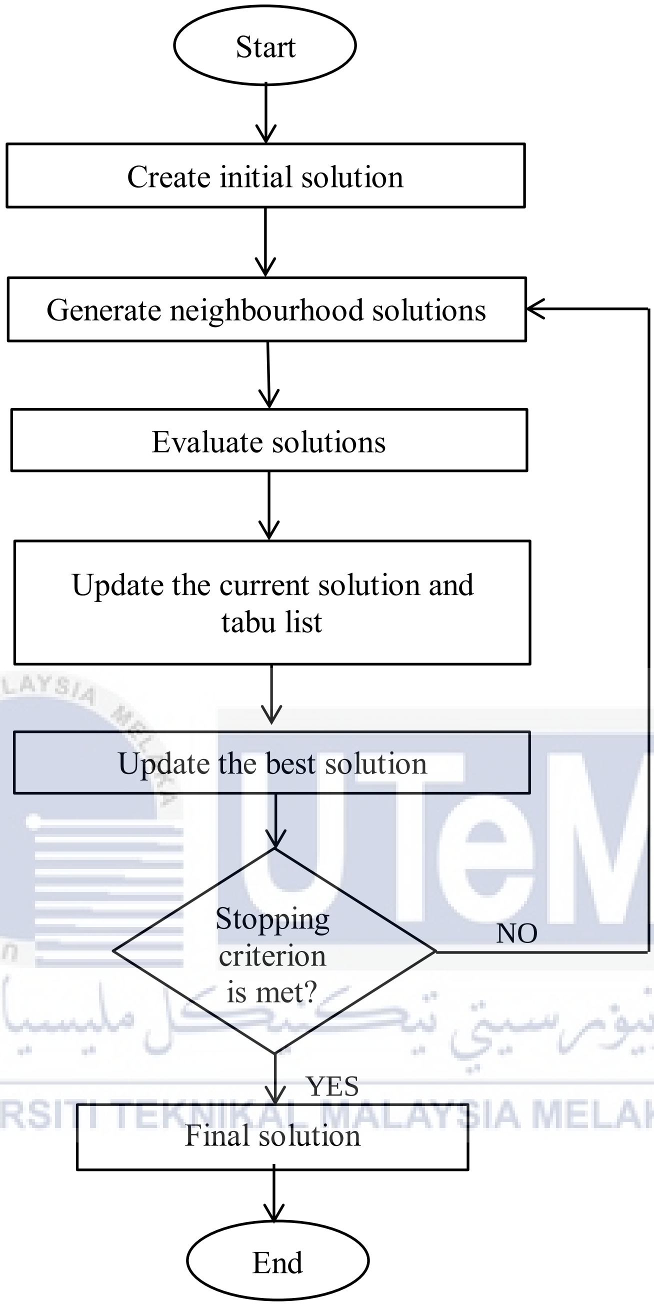

Figure 3.7: Flowchart of tabu search method

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Overview

Channel assignment problem is the main objective to be solved in this project. Problem formulation of penalty cost function is presented in Section 1.3. Basically it is calculated and analysed through a developed coding algorithm using MATLAB software. Simulation modelling on the total cost penalty value is executed. A few solutions were used to illustrate the computation of cost value for a feasible solution and a simple analytical analysis is carried out. The channel assignment problem is formulated as shown in Section 1.3 and it is integrated with a cost penalty element. By referring to the value of the cost penalty in each of the solutions, it shows the severeness of the interference in the calls.

4.2 Results and Discussions

Channel assignment problem is started to be solved by generating an initial solution. The set of binary variable represents the assigned channels of calls based on the demands in each of cells. The required parameters are the number of cells in a region, the number of available channels, the demand requirement or the number of call request for each of the cells and the minimum separation of frequencies between the calls which is stated in the compatibility matrix.

The two dimensional compatibility matrix presents the minimum frequency separation constraint between the two calls in order to get the zero interference among the channel assignment. It shows the minimum requirement of channels assigned in the same cell is needed to be separated by six frequencies from each other. It can be seen from the diagonal element of the compatibility matrix and the parameters are shown as follows:

Number of cells, N = 4

Number of available channels, M = 13

Demand of the channels or number of calls in each of the cells, D_j = [1, 1, 1, 3]

The minimum frequency separation constraint between the cells i and j for zero interference, $C_{i,j} = \begin{bmatrix} 6 & 5 & 0 & 0 \\ 5 & 6 & 0 & 1 \\ 0 & 0 & 6 & 2 \\ 0 & 1 & 2 & 6 \end{bmatrix}$

The data of channel assignment problem is used to run the simulation and is analysed. The assigned channel is randomly assigned as shown in Figure 4.1. From the coding of algorithm in the MATLAB, the data is analysed for 50 iterations which means the coding algorithm is looping for 50 times. In each of the loop, the channels are randomly swapped to produce twelve solutions in a neighbourhood. The initial channel assignment is channel 1 in cell 1, channel 2 in cell 2, channel 3 in cell 3 and channel 1, 4, and 13 in cell 4.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	●												
2		●											
3			●										
4	●			●									●

Figure 4.1: The initial allocation of assigned channel for channel assignment

For the initial channel assigned in the Figure 4.1, the interference occurred is eight. In other words, the current cost value of initial assigned channel is eight. Then, a neighbourhood is formed by swapping the assigned channel to another unused channel. One neighbourhood contains 12 feasible solutions with different binary set. Among the 12 feasible solutions, the binary set with the lowest cost value is set as the current binary set. For example, the allocated channel in cell 1 is reallocated from channel 1 to channel 11.

The other assigned channels remain the same. In this project, only one channel is swapped at a time so that the effects on the interference occurred can be seen clearly. As the channel assigned is changed, it is found that the current cost value evaluated decreases to four. It means that by reassigned the channel from channel 1 to channel 11, the cost value of the interference reduces from eight to four. At the moment, the current cost value holds a value of four for the first generated solution.

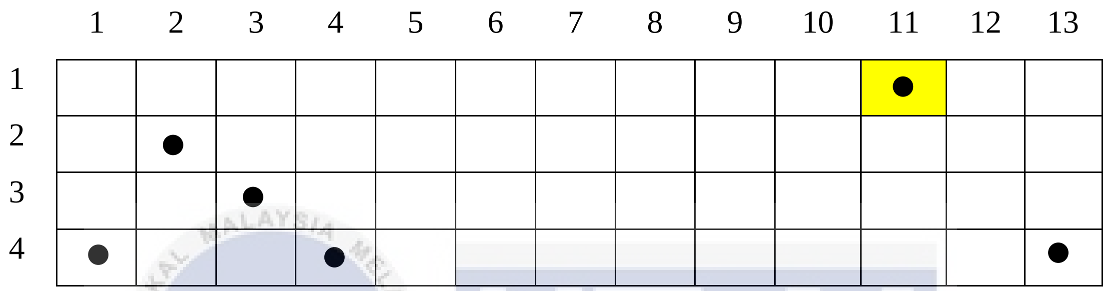

Figure 4.2: The first assigned channel for channel assignment

Reallocation of the channel assigned is repeated. The channel assigned is moved in cell 3 from channel 3 to channel 6. Then, the cost value is evaluated again. The result of the second assigned channel is three. As shown in the Figure 4.3, the cost value of interference is reduced from four to three.

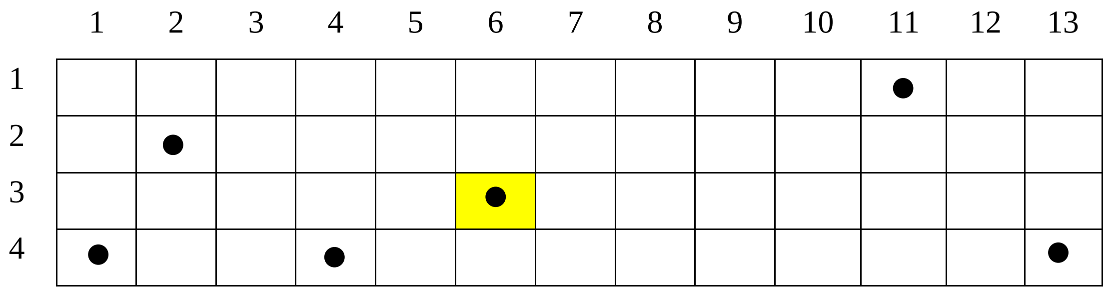

Figure 4.3: The second assigned channel for channel assignment

As the new value binary set is produced and its cost value is evaluated, the best cost value is revised if the current binary set produces a lower cost value. Since the number of channel is needed to be optimized, the lowest cost value is the best solution. The updated best cost value after two loops is three. Then, reallocated process of the channel assigned is repeated.

In the third neighbourhood, the most promising binary set is shown in Figure 4.4. The channel assigned in cell 3 is moved from channel 6 to channel 9. The cost value is evaluated again and it is the same as before which is three overlapped eventhough the channel is moved to the unused channel. Since the current cost value the same as the best cost value obtained, there is no change to the best solution. However, the move in the third current solution is being kept in tabu list.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1											●		
2		●											
3									●				
4	●			●									●

Figure 4.4: The third assigned channel for channel assignment

The process of channel assignment continues to reallocate the channel and the cost value is found lower than the best solution at the thirteenth neighbourhood/loops with the cost value of two. Then, this new current solution is stored as the new best solution for the channel assignment problem.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1									●				
2	●												
3						●							
4		●							●				●

Figure 4.5: The thirteenth assigned channel for channel assignment

The process continues and the binary set with a cost value lower than two is found at the twenty-sixth neighbourhood which the cost value of one. The best solution is then updated and its corresponding best cost value is changed from two to one.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1		●											
2												●	
3				●									
4		●						●					●

Figure 4.6: The twenty-sixth assigned channel for channel assignment

Every time the current solution is evaluated, the movement of channel being assigned is recorded in memory called as tabu list. The process repeats iteratively until stopping criteria is met, where the termination criteria used a fixed number of iteration. The process of searching for the best solution will stop when the termination criterion is satisfied and hence final solution is found. From the channel assignment problem stated, the best solution is shown in Figure 4.6 with the cost value of one.

In other words, the interference occurred among the assigned channel in the best solution is cost value of one. In real life application, the cost value represents interference or noise in a call. In other words, the cost value is a parameter that shows the severeness of frequencies overlapping in the frequency coverage area.

In the long run, every move in the current solution from a neighbourhood is recorded in tabu list. The tabu list is in a form of two-dimensional array that represents the forbidden moves for all positive entries. After each of the iteration is run, every element in the tabu list is decreased by one until it reaches freeing criterion which is zero. The tabu list is calculated from value of five as stated in the coding which is called forbidden iteration.

Originally the tabu list is holding zero values for all the entries. Since the forbidden move is 4, it means the visited move which is discarded in initial solution is set to be 4 in tabu list as shown in Figure 4.7. From the initial solution, it shows that, there is movement on cell 1.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**4**	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.7: The first tabu list of assigned channel for channel assignment problem

Then it can be seen that from the first generated solution, another channel is being moved again to generate the next current solution which is from cell 3 of channel 3 to another unused channel. Then, after second iteration is run the value of forbidden iteration on the cell 3 of channel 3 is holding the value of four. At the moment, the value of the tabu list on the cell 1 of channel 1 is continuously decreased.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**3**	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	**4**	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.8: The second tabu list of assigned channel for channel assignment problem

As the iteration continues to run, it is notice that from the second current solution, another channel allocation is being swapped again to generate new neighbourhood which is from cell 3 of channel 6 to another unused channel. Then, after third iteration is run the value of forbidden iteration on the cell 3 of channel 6 is holding the value of four. At the moment, the values in the tabu list for the cell 1 of channel 1 and cell 3 of channel 3 are continuously decreased to two and three, respectively.

As shown in Figure 4.4, the channel allocation in cell 3 is moved from channel 6 to channel 9. In Figure 4.9, it can be seen that the movement of the channel is already happened on the cell 1 of channel 1 and cell 3 of channel 3 and 6.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**2**	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	**3**	0	0	**4**	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.9: The third tabu list of assigned channel for channel assignment problem

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	4	-9	-6	-4	0	0	0	-31	-38	-42	-23	0
2	-30	-33	-29	-25	-28	-27	-24	-21	-1	-3	-12	3	-20
3	2	0	-44	1	0	0	0	-36	-41	-37	0	0	0
4	-35	0	0	-34	0	0	0	0	-22	0	0	0	0

Figure 4.10: The last one (fifty) tabu list of assigned channel for channel assignment problem

From the tabu list, it can be seen that every movement of the channel assigned is being kept in the memory where it is the properties of tabu search. Tabu list is used to prevent a visited solution being reused in recent iterations. Meanwhile, from the coding algorithm of the tabu search, it is written to keep the current and the best cost value until termination criterion is met. In this case, the termination criterion chosen is by setting the iteration up to a maximum of 50 loops.

The result of cost value for each of the iterations is recorded and it is represented in Figure 4.11. It shows that as the number of iteration increases, the more of new neighbourhood is being generated. The solution keeps improved as the iteration goes. When the termination criterion is reached, the binary set that gives the lowest cost value is treated as the optimal solution.

The lesser is the cost value, the lesser is the severeness of the interference occurred in the call. It is due to the evaluation of cost value takes into consideration of the interference between channels assigned. The linear line of the graph shows the cost value and the number of iteration are inversely proportional to each other.

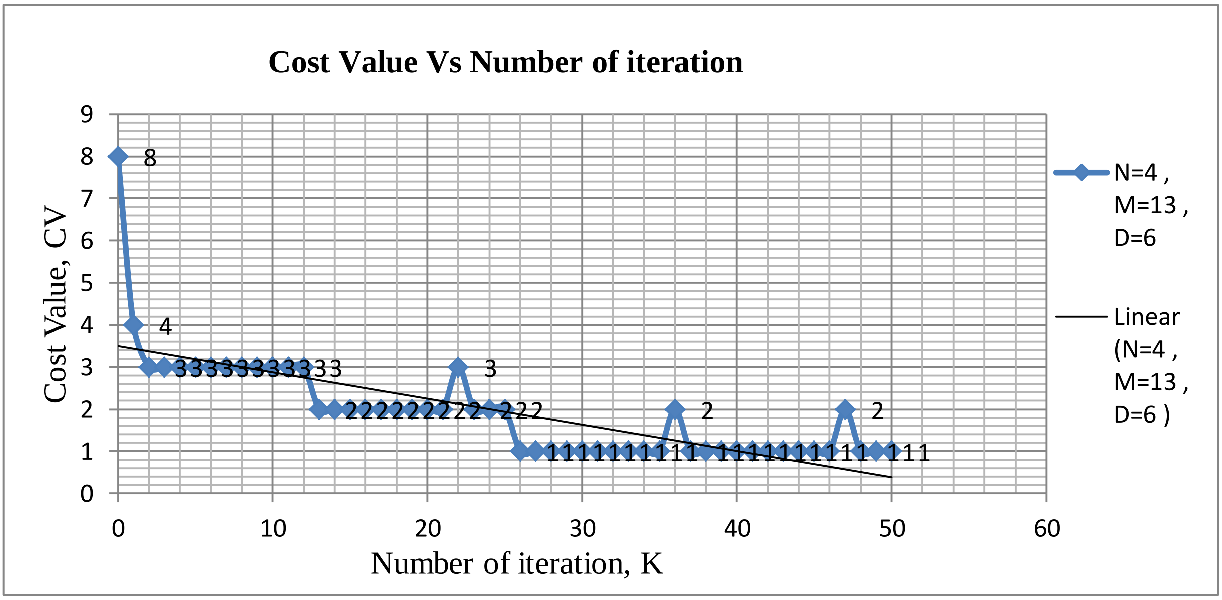

Figure 4.11: Optimization process of the channel assignment problem

Further research is carried out to investigate the effect on cost value which different number of channels and demands of channels. The result of channel assignment problem with different parameters is analysed. By varying the demand size of the channel assignment, *D*, the effect is analysed to see the lowest cost value that the solution may get. At the same time, it shows the severeness of interference occurred in the channel assignment problem. The number of cells is fixed to four. Then, the size of demand of the channels or number of calls required is varied in 5, 10, 15, 20 and 25 of calls at a time.

For the first case which is five demands in the channel assignment, the number of available channel is tested with step size of 5, from 5 to 25 available channels. It is found that the cost value evaluated is from one, then being minimized to zero cost value when the number of available channels increases. It means that, the channel assigned reaches the level of free-interference as the available channel increases. In other words, the constraint of the compatibility matrix is fully satisfied and the distance of the frequencies between the assigned channels is far enough to each other.

Then, the demand is increased to ten in case 2 in the channel assignment. The cost value is found optimized from 70 to 4 as the availability of channels increases. As the termination criterion reaches the maximum limit, the best cost value found along the neighbourhood process is four with 25 channels used. As shown in Figure 4.12, as the number of demand increases, the cost value of each of the categories that represents the interference increases too. This is due to the higher demand served by the same number of channels.

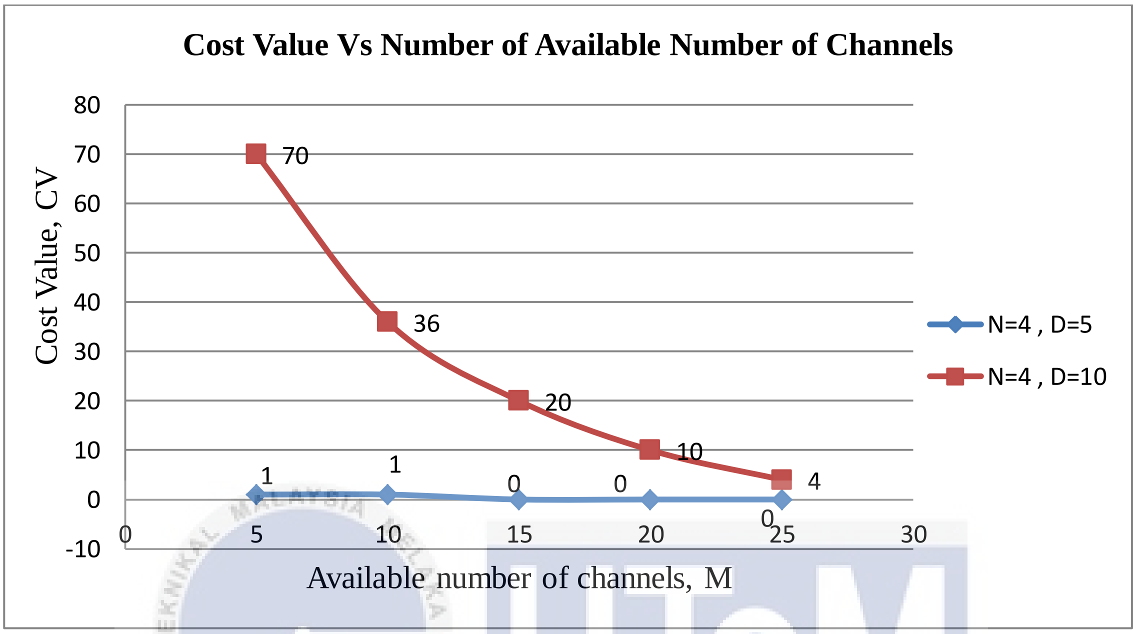

Figure 4.12: Amount of blocked calls for small demands required in the channel assignment

The experiment is further run with the demands of 15, 20 and 25. It is found that the cost values decreases from 102, 207, and 328 to 29, 70 and 123, respectively, as the number of channels increases from 10 to 25 with a step size of 5. None of the channel assigned reaches the stage of interference free as the demands of the channel increases. In other words, the constraint of the compatibility matrix is not satisfied and there is interference occurred between the assigned channels.

Then, by using a fixed number of channels, for example M=15, cost values of the demands shows of 15, 20 and 25 are 65, 135 and 229, respectively. This is due to the higher number of demands requiring more channels for the calls. By using the same number of available channels, the channels assigned will be closer to each other in the case of higher demands, and hence, the cost value increases as more penalty is imposed on the violation of minimal separation of channels.

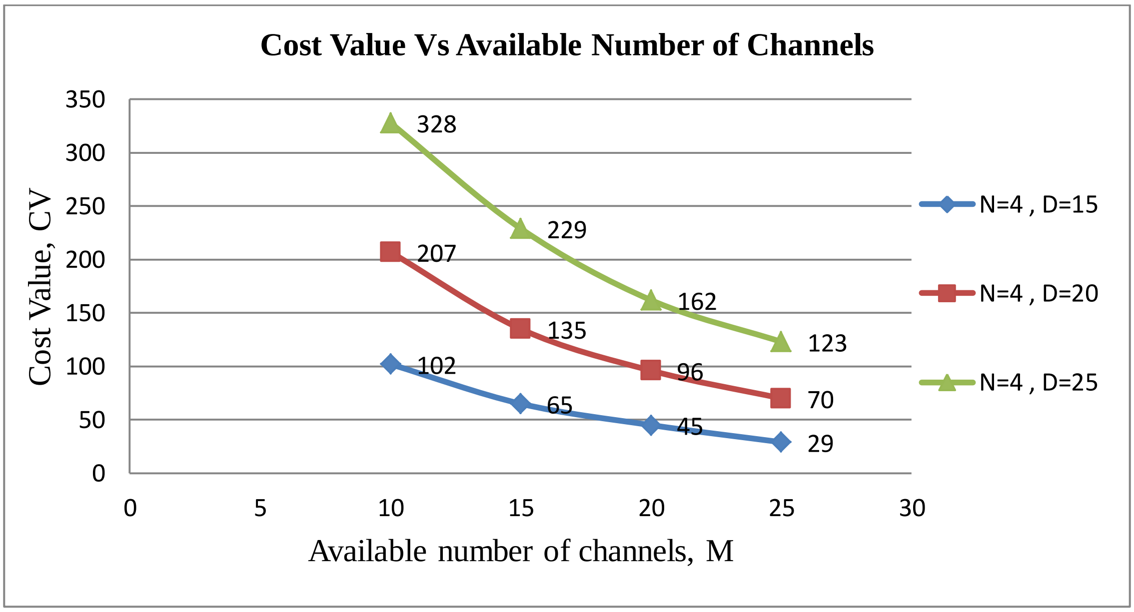

Figure 4.13: Cost value for higher demands required in the channel assignment

The results show the cost value depends on the demand sizes and the number of available channels in the network. The larger is the demand, the more channels are needed to keep the cost value low. The higher is the cost value, the more number of calls may be blocked and the quality of mobile communication decreased. Hence, in channel assignment problem, channels need to be assigned optimally with lower cost value to provide higher quality services in mobile communication.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion, this project presents the implementation of tabu search in solving channel assignment problem in mobile communication. Interference is allowed at the minimum level. Cost penalty value is computed based on the problem formulation for each of the obtained candidate solutions. The lower cost penalty represents the lower interference occurs.

A coding algorithm based on the tabu search technique is developed using MATLAB software to solve channel assignment problem search. The optimization of channel assignment is achieved as the optimal solution is found based on the best solution which is the lowest cost value as termination criterion is met. The optimal solution is obtained by carrying out exhaustive search using the developed coding algorithm where new neighbourhood is generated from current solution in each of the iterations.

A neighbourhood contains a pool of feasible solutions. Then, the cost value is evaluated and the lowest from the neighbourhood is being chosen as current solution. One neighbourhood gives a current solution, and from all the current solutions the best one is the optimal solution. Based on the result, it found that, the larger the iteration counters is chosen, the lower the cost value may be produced.

Practically, the demand of the channel assignment is varied in order to see the effect of the demands on the cost value. It found that, for a fixed available channel, as the number of the demands increases, the cost value increases too. In order to keep the cost value low, more channels are needed to satisfy the demand calls.

5.2 Recommendation

To maximize the utilization of channels, channel borrowing strategy may be applied in fixed channel assignment model. In real time, certain cells may have idle channels while some other cells may need extra channels. The channel borrowing strategy may allow idle channels from a cell be borrowed by another cell which is in need. Future investigation is suggesting implementing the channel assignment problem by using Dynamic Channel Assignment (DCA) instead of Static Channel Assignment (SCA).

This is due to the DCA can be reused back as the call is complete or end. In other words, the channel used for a call is being released when a call is end since it is do not own any particular channel for the demand calls. Therefore, the available channel can be used more effectively and efficiently.

REFERENCES

[1] M. Abhijit, "Lecture Notes on Mobile Communication," *Indian Inst. Technol. Guwahati*, no. November, pp. 1–37, 2009.

[2] M. A. Bentil, A. A. Boyd, et al., "Consumers and Mobile Financial Services 2016," *Consumer and Community Development Research Section of the Federal Reserve Board's Division of Consumer and Community Affairs (DCCA)*, no. March, 2016.

[3] J. Cheeneebash, H. Coomar, and S. Rughooputh, "A comprehensive review of methods for the channel allocation problem," *African Minds*, pp. 1–54, 2014.

[4] Y. Peng, L. Wang, and B. H. Soong, "Optimal Channel Assignment in Cellular Systems using Tabu Search," *IEEE*, pp. 31–35, 2003.

[5] K. Smith and M. Palaniswami, "Static and Dynamic Channel Assignment Using Neural Networks," *IEEE J. Sel. AREAS Commun.*, vol. 15, no. 2, pp. 238–249, 1997.

[6] L. Wang, S. Li, S. C. Lay, W. H. Yu, and C. Wan, "Genetic Algorithms for Optimal Channel Assignments in Mobile Communications," *Nanyang Technol. Univ.*, pp. 1–33.

[7] M. Duque-anton, D. Kunz, and and B. R. Riiber, "Channel Assignment for Cellular Radio Using Simulated Annealing," *IEEE Trans. Veh. Technol.*, vol. 42, no. 1, pp. 14–21, 1993.

[8] M. S. Kamboj and J. Sengupta, "Comparative Analysis of Simulated Annealing and Tabu Search Channel Allocation Algorithms," *Int. J. Comput. Theory Eng.*, vol. 1, no. 5, pp. 588–591, 2009.

[9] G. Didem and G. Genc, "Channel Assignment Problem in Cellular Networks : A Reactive Tabu Search Approach," *IEEE*, 2009.

[10] M. P. Vecchi, S. Kirkpatrick, C. D. Gelatt, "Optimization by Simulated Annealing," *Sci. Publ.*, vol. 220, no. 4598, pp. 671–680, 2007.

[11] L. Wang, S. Ng, S. Lee, and W. Y. Hing, "Solving channel assignment problems using local search methods and simulated annealing," *Nanyang Technol. Univ.*, pp. 1–8.

[12] F. Glover and M. Laguna, "Principles of Tabu Search," *Univ. Color.*, 1977.

[13] D. J. Castelino, S. Hurley, and N. M. Stephens, “A tabu search algorithm for frequency assignment,” *Sci. Publ.*, vol. 63, pp. 301–319, 1996.

[14] A. Capone and M. Trubian, “Channel Assignment Problem in Cellular Systems : A New Model and a Tabu Search Algorithm,” *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1252–1260, 1999.

[15] and D. H. S. R. Montemanni, J. N. J. Moon, “An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency-Assignment Problem,” *IEEE Trans. Veh. Technol.*, vol. 52, no. 3, pp. 891–901, 2003.

APPENDICES

Appendix A: Project Gantt Chart

No.	Activities	Duration (weeks)																											
		Semester 1														Semester 2													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Briefing final year project	■																											
2	Project title selection		■																										
3	Identify project objective and problem		■	■	■																								
4	Literature review			■	■	■	■	■	■	■	■	■	■	■		■	■	■	■	■	■	■	■	■	■	■	■	■	
5	Understand problem formulation				■	■	■	■	■																				
6	PSM 1 report								■	■	■	■	■	■															
7	Submission PSM 1													■															
8	Presentation PSM 1														■														
9	Simulation of problem formulation															■	■	■	■	■									
10	Analysis of simulation																			■	■	■	■						
11	Findings and conclusion																						■	■	■				
12	PSM 2 report																							■	■	■	■	■	
13	Submission PSM 2																											■	
14	Presentation PSM 2																												■

Appendix B: Data Set and Result of Channel Assignment Problem

N=4; M=13; D=6;
X=zeros (N, M);
X(1,1)=1; X(2,2)=1; X(3,3)=1; X(4,1)=1; X(4,4)=1; X(4,13)=1;

$$C_{i,j} = \begin{bmatrix} 6 & 5 & 0 & 0 \\ 5 & 6 & 0 & 1 \\ 0 & 0 & 6 & 2 \\ 0 & 1 & 2 & 6 \end{bmatrix}$$

```
>> TABUSEARCH

Current_CV =     8
Current_X =

   1   0   0   0   0   0   0   0   0   0   0   0   0
   0   1   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0   0
   1   0   0   1   0   0   0   0   0   0   0   0   1

1ST
Current_CV =     4
Current_X =

   0   0   0   0   0   0   0   0   0   0   1   0   0
   0   1   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0   0
   1   0   0   1   0   0   0   0   0   0   0   0   1

Tabu_list =

   4   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0

Best_CV =     4
Best_X =

   0   0   0   0   0   0   0   0   0   0   1   0   0
   0   1   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0   0
   1   0   0   1   0   0   0   0   0   0   0   0   1

2ND
Current_CV =     3
Current_X =

   0   0   0   0   0   0   0   0   0   0   1   0   0
   0   1   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   0   0   0   0   0   0
   1   0   0   1   0   0   0   0   0   0   0   0   1
```

Tabu_list =

3	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Best_CV = 3

Best_X =

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

3RD

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1

Tabu_list =

2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

4TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1

Tabu_list =

1	0	0	0	0	0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

5TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1

```
Tabu_list =

     0    0    0    0    0    0    0    0    0    0    3    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    1    0    0    2    0    0    4    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0
```

6TH

```
Current_CV =     3
Current_X =

     0    0    0    0    0    0    0    0    0    1    0    0    0
     0    1    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    1    0    0    0
     1    0    0    1    0    0    0    0    0    0    0    0    1

Tabu_list =

    -1    0    0    0    0    0    0    0    0    0    2    4    0
     0    1    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    1    0    0    3    0    0    0    0
     0    1    0    0    0    0    0    0    0    0    0    0    0
```

7TH

```
Current_CV =     3
Current_X =

     0    0    0    0    0    0    0    0    0    1    0    0    0
     0    0    0    0    1    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    1    0    0    0
     1    0    0    1    0    0    0    0    0    0    0    0    1

Tabu_list =

    -2    0    0    0    0    0    0    0    0    0    1    3    0
     0    4    0    0    0    0    0    0    0    0    0    0    0
     0    0   -1    0    0    0    0    0    2    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0
```

8TH

```
Current_CV =     3
Current_X =

     0    0    0    0    0    0    0    0    1    0    0    0    0
     0    1    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    1    0    0    0
     1    0    0    1    0    0    0    0    0    0    0    0    1

Tabu_list =

    -3    0    0    0    0    0    0    0    0    4    0    2    0
     0    3    0    0    0    0    0    0    0    0    0    0    0
     0    0   -2    0    0   -1    0    0    1    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0
```

9TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1

Tabu_list =

-4	0	0	0	0	0	0	0	0	3	-1	1	0
0	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-2	0	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

10TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1

Tabu_list =

-5	0	0	0	0	0	0	0	0	2	-2	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	-4	0	0	-3	0	4	-1	3	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

11TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0	0	0	1

Tabu_list =

-6	0	0	0	0	0	0	0	0	1	-3	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-5	0	0	-4	0	3	-2	2	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0

12TH

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-7	0	0	0	0	0	0	0	0	0	-4	-2	-12
0	-1	0	0	0	0	0	0	0	0	0	0	0
0	0	-6	0	0	-5	0	2	-3	1	0	0	0
3	0	0	4	0	0	0	0	0	0	0	0	0

13TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-8	0	0	0	0	0	0	0	0	-1	-5	-3	0
0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	-7	0	0	-6	0	1	-4	0	0	0	0
2	0	0	3	0	0	0	0	0	0	0	0	0

Best_CV = 2
Best_X =

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**

14TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-9	0	0	0	0	0	0	0	0	-2	-6	-4	0
0	3	0	0	0	0	0	0	0	0	0	0	0
0	0	-8	0	0	4	0	0	-5	-1	0	0	0
1	0	0	2	0	0	0	0	0	0	0	0	0

15TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-10	0	0	0	0	0	0	0	4	-3	-7	-5	0
0	2	0	0	0	0	0	0	0	0	0	0	0
0	0	-9	0	0	3	0	-1	-6	-2	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0

16TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-11	0	0	0	0	0	0	0	3	-4	-8	-6	0
4	1	0	0	0	0	0	0	0	0	0	0	0
0	0	-10	0	0	2	0	-2	-7	-3	0	0	0
-1	0	0	0	0	0	0	0	0	0	0	0	0

17TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-12	0	0	0	0	0	0	0	2	-5	-9	-7	0
3	0	4	0	0	0	0	0	0	0	0	0	0
0	0	-11	0	0	1	0	-3	-8	-4	0	0	0
-2	0	0	-1	0	0	0	0	0	0	0	0	0

18TH
Current_CV = 2
Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-13	0	0	0	0	0	0	0	1	-6	-10	-8	0
2	-1	3	0	4	0	0	0	0	0	0	0	0
0	0	-12	0	0	0	0	-4	-9	-5	0	0	0
-3	0	0	-2	0	0	0	0	0	0	0	0	0

19^TH^

Current_CV = 2

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-14	0	0	0	0	0	0	0	0	-7	-11	-9	0
1	-2	2	0	3	4	0	0	0	0	0	0	0
0	0	-13	0	0	-1	0	-5	-10	-6	0	0	0
-4	0	0	-3	0	0	0	0	0	0	0	0	0

20^TH^

Current_CV = 2

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-15	0	0	0	0	0	0	0	-1	-8	-12	-10	0
0	-3	1	0	2	3	0	0	0	0	0	0	0
0	0	-14	0	0	4	0	-6	-11	-7	0	0	0
-5	0	0	-4	0	0	0	0	0	0	0	0	0

21^ST^

Current_CV = 2

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-16	0	0	0	0	0	0	0	-2	-9	-13	-11	0
-1	-4	0	4	1	2	0	0	0	0	0	0	0
0	0	-15	0	0	3	0	-7	-12	-8	0	0	0
-6	0	0	-5	0	0	0	0	0	0	0	0	0

22^ND^

Current_CV = 3

Current_X =

0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-17	0	0	0	0	0	0	0	-3	-10	-14	-12	0
-2	-5	-1	3	0	1	4	0	0	0	0	0	0
0	0	-16	0	0	2	0	-8	-13	-9	0	0	0
-7	0	0	-6	0	0	0	0	0	0	0	0	0

23RD

Current_CV = 2

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	1

Tabu_list =

-18	0	0	0	0	0	0	0	-4	-11	-15	4	0
-3	-6	-2	2	-1	0	3	0	0	0	0	0	0
0	0	-17	0	0	1	0	-9	-14	-10	0	0	0
-8	0	0	-7	0	0	0	0	0	0	0	0	0

24TH

Current_CV = 2

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-19	0	0	0	0	0	0	0	-5	-12	-16	3	0
-4	-7	-3	1	-2	-1	2	0	0	0	0	0	0
0	0	-18	0	0	0	0	-10	-15	-11	0	0	0
-9	0	0	-8	0	0	0	0	4	0	0	0	0

25TH

Current_CV = 2

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-20	0	0	0	0	0	0	0	-6	-13	-17	2	0
-5	-8	-4	0	-3	-2	1	4	0	0	0	0	0
0	0	-19	0	0	-1	0	-11	-16	-12	0	0	0
-10	0	0	-9	0	0	0	0	3	0	0	0	0

26TH

Current_CV = 1

Current_X =

```
0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1
```

Tabu_list =

```
-21 0 0 0 0 0 0 0 -7 -14 -18 1 0
-6 -9 -5 -1 -4 -3 0 3 0 0 0 0 4
0 0 -20 0 0 -2 0 -12 -17 -13 0 0 0
-11 0 0 -10 0 0 0 0 2 0 0 0 0
```

Best_CV = 1

Best_X =

```
0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1
```

27TH

Current_CV = 1

Current_X =

```
0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1
```


Tabu_list =

```
-22 0 0 0 0 0 0 0 -8 -15 -19 0 0
-7 -10 -6 -2 -5 -4 -1 2 0 0 0 4 3
0 0 -21 0 0 -3 0 -13 -18 -14 0 0 0
-12 0 0 -11 0 0 0 0 1 0 0 0 0
```

28TH

Current_CV = 1

Current_X =

```
0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1
```

Tabu_list =

```
-23 4 0 0 0 0 0 0 -9 -16 -20 -1 0
-8 -11 -7 -3 -6 -5 -2 1 0 0 0 3 2
0 0 -22 0 0 -4 0 -14 -19 -15 0 0 0
-13 0 0 -12 0 0 0 0 0 0 0 0 0
```

29TH

Current_CV = 1

Current_X =

0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-24	3	0	4	0	0	0	0	-10	-17	-21	-2	0
-9	-12	-8	-4	-7	-6	-3	0	0	0	0	2	1
0	0	-23	0	0	-5	0	-15	-20	-16	0	0	0
-14	0	0	-13	0	0	0	0	-1	0	0	0	0

30TH

Current_CV = 1

Current_X =

0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-25	2	0	3	0	0	0	0	-11	-18	-22	-3	0
-10	-13	-9	-5	-8	-7	-4	-1	0	0	0	1	0
0	0	-24	4	0	-6	0	-16	-21	-17	0	0	0
-15	0	0	-14	0	0	0	0	-2	0	0	0	0

31ST

Current_CV = 1

Current_X =

1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-26	1	0	2	4	0	0	0	-12	-19	-23	-4	0
-11	-14	-10	-6	-9	-8	-5	-2	0	0	0	0	-1
0	0	-25	3	0	-7	0	-17	-22	-18	0	0	0
-16	0	0	-15	0	0	0	0	-3	0	0	0	0

32ND

Current_CV = 1

Current_X =

1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-27	0	0	1	3	0	0	0	-13	-20	-24	-5	0
-12	-15	-11	-7	-10	-9	-6	-3	0	0	4	-1	-2
0	0	-26	2	0	-8	0	-18	-23	-19	0	0	0
-17	0	0	-16	0	0	0	0	-4	0	0	0	0

33RD

Current_CV = 1

Current_X =

0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

4	-1	0	0	2	0	0	0	-14	-21	-25	-6	0
-13	-16	-12	-8	-11	-10	-7	-4	0	0	3	-2	-3
0	0	-27	1	0	-9	0	-19	-24	-20	0	0	0
-18	0	0	-17	0	0	0	0	-5	0	0	0	0

34TH

Current_CV = 1

Current_X =

0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

3	-2	0	-1	1	0	0	0	-15	-22	-26	-7	0
-14	-17	-13	-9	-12	-11	-8	-5	0	0	4	-3	-4
0	0	-28	0	0	-10	0	-20	-25	-21	0	0	0
-19	0	0	-18	0	0	0	0	-6	0	0	0	0

35TH

Current_CV = 1

Current_X =

0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

2	-3	0	-2	0	0	0	0	-16	-23	-27	-8	0
-15	-18	-14	-10	-13	-12	-9	-6	4	0	3	-4	-5
0	0	-29	-1	0	-11	0	-21	-26	-22	0	0	0
-20	0	0	-19	0	0	0	0	-7	0	0	0	0

36TH

Current_CV = 2

Current_X =

0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

1	-4	0	-3	-1	0	0	0	-17	-24	-28	-9	0
-16	-19	-15	-11	-14	-13	-10	-7	3	4	2	-5	-6
0	0	-30	-2	0	-12	0	-22	-27	-23	0	0	0
-21	0	0	-20	0	0	0	0	-8	0	0	0	0

37TH

Current_CV = 1

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

0	-5	4	-4	-2	0	0	0	-18	-25	-29	-10	0
-17	-20	-16	-12	-15	-14	-11	-8	2	3	1	-6	-7
0	0	-31	-3	0	-13	0	-23	-28	-24	0	0	0
-22	0	0	-21	0	0	0	0	-9	0	0	0	0

38TH

Current_CV = 1

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-1	-6	3	-5	-3	0	0	0	-19	-26	-30	-11	0
-18	-21	-17	-13	-16	-15	-12	-9	1	2	0	-7	-8
0	0	-32	4	0	-14	0	-24	-29	-25	0	0	0
-23	0	0	-22	0	0	0	0	-10	0	0	0	0

39TH

Current_CV = 1

Current_X =

0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-2	4	2	-6	-4	0	0	0	-20	-27	-31	-12	0
-19	-22	-18	-14	-17	-16	-13	-10	0	1	-1	-8	-9
0	0	-33	3	0	-15	0	-25	-30	-26	0	0	0
-24	0	0	-23	0	0	0	0	-11	0	0	0	0

40TH

Current_CV = 1

Current_X =

0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-3	3	1	4	-5	0	0	0	-21	-28	-32	-13	0
-20	-23	-19	-15	-18	-17	-14	-11	-1	0	-2	-9	-10
0	0	-34	2	0	-16	0	-26	-31	-27	0	0	0
-25	0	0	-24	0	0	0	0	-12	0	0	0	0

41ST

Current_CV = 1

Current_X =

0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-4	2	0	3	-6	0	0	0	-22	-29	-33	-14	0
-21	-24	-20	-16	-19	-18	-15	-12	-2	-1	-3	-10	-11
0	0	-35	4	0	-17	0	-27	-32	-28	0	0	0
-26	0	0	-25	0	0	0	0	-13	0	0	0	0

42ND

Current_CV = 1

Current_X =

1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-5	1	-1	2	4	0	0	0	-23	-30	-34	-15	0
-22	-25	-21	-17	-20	-19	-16	-13	-3	-2	-4	-11	-12
0	0	-36	3	0	-18	0	-28	-33	-29	0	0	0
-27	0	0	-26	0	0	0	0	-14	0	0	0	0

43[RD]

Current_CV = 1

Current_X =

1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-6	0	-2	1	3	0	0	0	-24	-31	-35	-16	0
-23	-26	-22	-18	-21	-20	-17	-14	-4	4	-5	-12	-13
0	0	-37	2	0	-19	0	-29	-34	-30	0	0	0
-28	0	0	-27	0	0	0	0	-15	0	0	0	0

44[TH]

Current_CV = 1

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

4	-1	-3	0	2	0	0	0	-25	-32	-36	-17	0
-24	-27	-23	-19	-22	-21	-18	-15	-5	3	-6	-13	-14
0	0	-38	1	0	-20	0	-30	-35	-31	0	0	0
-29	0	0	-28	0	0	0	0	-16	0	0	0	0

45[TH]

Current_CV = 1

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

3	-2	-4	-1	1	0	0	0	-26	-33	-37	-18	0
-25	-28	-24	-20	-23	-22	-19	-16	4	2	-7	-14	-15
0	0	-39	0	0	-21	0	-31	-36	-32	0	0	0
-30	0	0	-29	0	0	0	0	-17	0	0	0	0

46[TH]

Current_CV = 1

Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

2	-3	-5	-2	0	0	0	0	-27	-34	-38	-19	0
-26	-29	-25	-21	-24	-23	-20	-17	3	1	-8	-15	-16
0	0	-40	-1	0	4	0	-32	-37	-33	0	0	0
-31	0	0	-30	0	0	0	0	-18	0	0	0	0

47TH
Current_CV = 2
Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

1	-4	-6	-3	-1	0	0	0	-28	-35	-39	-20	0
-27	-30	-26	-22	-25	-24	-21	-18	2	0	-9	-16	-17
0	0	-41	4	0	3	0	-33	-38	-34	0	0	0
-32	0	0	-31	0	0	0	0	-19	0	0	0	0

48TH
Current_CV = 1
Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

0	-5	-7	-4	-2	0	0	0	-29	-36	-40	-21	0
-28	-31	-27	-23	-26	-25	-22	-19	1	-1	-10	-17	-18
4	0	-42	3	0	2	0	-34	-39	-35	0	0	0
-33	0	0	-32	0	0	0	0	-20	0	0	0	0

49TH
Current_CV = 1
Current_X =

0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-1	-6	-8	-5	-3	0	0	0	-30	-37	-41	-22	0
-29	-32	-28	-24	-27	-26	-23	-20	0	-2	-11	4	-19
3	0	-43	2	0	1	0	-35	-40	-36	0	0	0
-34	0	0	-33	0	0	0	0	-21	0	0	0	0

50TH

Current_CV = 1

Current_X =

0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1

Tabu_list =

-2	4	-9	-6	-4	0	0	0	-31	-38	-42	-23	0
-30	-33	-29	-25	-28	-27	-24	-21	-1	-3	-12	3	-20
2	0	-44	1	0	0	0	-36	-41	-37	0	0	0
-35	0	0	-34	0	0	0	0	-22	0	0	0	0

Best_CV = 1

Best_X =

0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**