Dynamic Channel Allocation Using Simulated Annealing In 802.11 WLANs

Author: Krishna Subramani

Contents

1. Introduction

IEEE 802.11 is a set of media access control and physical layer specifications for implementing wireless local area network (WLAN) communication in 2.4, 3.6, 5 and 60 GHz frequency bands. IEEE 802.11 WLANs are widely deployed in urban area due to the growing demand for wireless data service. 802.11 b/g serves to be the popular choice due to its availability and potential for expansion. Since, there are only three non-overlapping channels in the 2.4 GHz spectrum, providing wireless service to the growing demand is not straight forward. The use of overlapping channels results in adjacent channel interference and hence degradation in the quality of service. In real world scenarios, Access Points (APs) belonging to same or different operators can access the unlicensed ISM band without any restriction and coordination between them. Therefore, it is mandatory to control the interference to improve the quality of service. In scenarios where several APs are deployed, it is necessary to allocate the three available channels in an optimal way such that the interference is minimized.

In this project, a framework using Simulated Annealing (SA) to solve the channel assignment problem in the single – hop multiple AP 802.11 b/g networks is implemented as described in [1]. The objective of the project is to implement the SA algorithm and find the optimal channel assignment for the APs operating in a defined network scenario. The parameters of the SA are chosen in such a way that it minimizes the interference in the system and provides us with the corresponding channel assignment. As explained in [1], the channel assignment by SA algorithm is very close to the known optimum found by Branch and Bound method.

2. Background theory

2.1. Overview on IEEE 802.11 b/g

IEEE 802.11 b/g devices can transmit in the 2.4 GHz band with a total of 14 available channels. In the US, there are only 11 channels legally available and only 13 channels are available in Europe. However, there are only 3 non-overlapping (orthogonal) channels 1, 6 and 11 available in the 2.4 GHz band. Figure 1 depicts the spectrum analyzer view of the frequency space occupied by the 14 available channels.

When designing a wireless LAN, overlapping RF cell coverage is necessary to provide better coverage at the edges of the cell. However, the overlapping cells should not have overlapping frequency space. Therefore, when more than three APs are deployed in a network scenario, only the non-overlapping channels (1, 6 and 11) must be used. This is known as channel reuse pattern. Usage of overlapping channels by the APs causes adjacent channel interference and results in decreased throughput and increased latency.

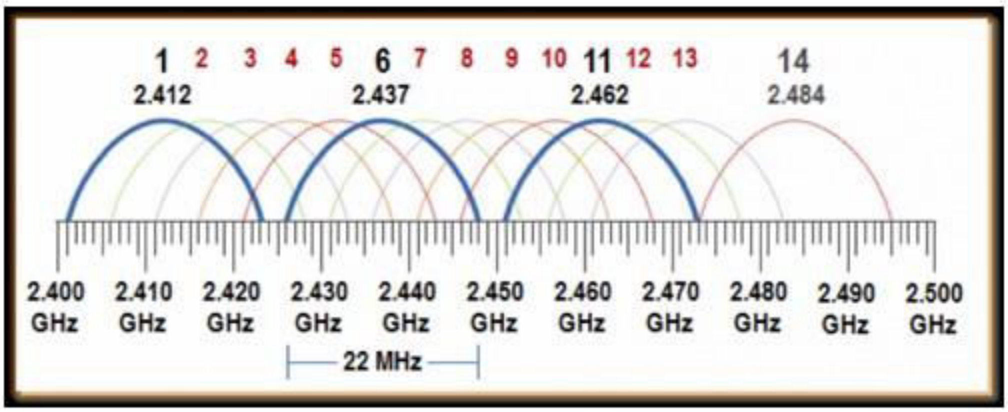

Figure 1 Spectrum analyzer view of 2.4 GHz band

2.2. Channel allocation problem

When there is an increase in the demand for wireless service, several wireless devices are deployed to serve the users. Since only limited bandwidth is available for use by these devices, there is a necessity to continuously reuse the available frequency channels. These available channels must be assigned to the devices in a manner such that the interference between the devices is always maintained at its minimum. This is known as channel allocation problem. The issues which arises due to the allocation of channels in the available frequency space are twofold:

2.2.1. Adjacent channel interference

When the data is transmitted in a specific frequency channel, the receiver is tuned to the same frequency channel for receiving the transmitted data. However, the practical transmitters and receivers do not possess ideal properties. A transmitter can violate the specific frequency band allocated to it and may overshoot into the adjacent channel while transmitting the data. In the same way, the receiver listening to a specific channel may also listen to adjacent channel frequencies and decode the data in it along with the required data. The interference caused by the signals from the adjacent channels is known as adjacent channel interference (ACI). ACI can be avoided by using channels which are well separated. In other words, the channels which are non-overlapping could be used to eliminate adjacent channel interference. This is the reason for using only 3 channels (1, 6 and 11) in the 2.4 GHz band for communication by 802.11 b/g devices.

2.2.2. Co-channel interference

When two APs using the same channel are deployed close together, the transmitted signals from the APs interfere with each other. This interference is known as Co-Channel Interference (CCI). CCI is more problematic when WLANs are deployed to support voice or RFID location tagging. Because, these types of scenarios require denser deployment of APs. Therefore, there is a great potential for adjacent APs to transmit signals in the same frequency channel. CCI can be avoided by preventing the adjacent APs from using the same channel at any point in time. This requires the need for optimal channel assignment to APs in a network such that the interference is minimized.

2.3. Classifications in channel allocation

2.3.1. Fixed channel allocation

In Fixed Channel Allocation (FCA), the channels are pre-allocated to the APs in such a way that the APs using the same channel must be separated by a certain distance known as the reuse distance so that the CCI is minimized. The elements which decide the frequency reuse are the reuse distance and the reuse factor (rate at which the same frequency can be used in the network). This is the simplest form of channel assignment but the utilization of available channels is non-optimal when the traffic is non-uniform.

2.3.2. Dynamic channel allocation

In Dynamic Channel Allocation (DCA), no set relationship exists between the channels and the APs. Instead, the channels are part of a pool of resources. Whenever a channel is needed by an AP, it is assigned with an arbitrary channel under the constraint that the frequency reuse cannot be violated. DCA attempts to alleviate the non-uniform traffic problem faced by FCA. There are two issues with DCA:

- DCA methods typically have a degree of randomness associated with them and this leads to the fact that frequency reuse is often not maximized unlike the case for FCA systems.
- DCA method involve complex algorithms for deciding which available channel is efficient. These algorithms can be computationally intensive and may require large computing resources in order to be real-time.

2.3.3. Hybrid channel allocation

The third category of channel allocation can be explained in terms of cellular communication. It includes systems that are hybrids of fixed and dynamic channel allocation systems. The most straightforward hybrid allocation scheme is Channel

Borrowing. In this, channels are assigned to the cells just as in FCA. If a cell needs a channel in excess of the channels previously assigned to it, that cell may borrow a channel from one of its neighbouring cells given that a channel is available and use of this channel will not violate frequency reuse requirements. The major problem with channel borrowing is that when a cell borrows a channel from a neighbouring cell, other nearby cells are prohibited from using the borrowed channel because of CCI. In WLANs, the APs are assigned with only one channel at any point in time. So, the hybrid channel allocation cannot be used for WLAN systems.

3. Simulated Annealing

Simulated Annealing (SA) is a probabilistic and metaheuristic algorithm for global optimization problems. It is used to find a good approximate of the global optimum of an objective function in a large search space. It was invented by S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi [2] in 1983. Finding an optimal solution for certain optimization problems can be an incredibly difficult task, often practically impossible. This is because when a problem gets sufficiently large, we need to search through an enormous number of possible solutions to find the optimal one. Even with modern computing power there are often too many solutions to consider. In this case, we cannot realistically expect to find the optimal solution within a sensible amount of time. Therefore, SA can be used to find close enough optimal solution within a particular time limit.

The SA algorithm was originally inspired from the physical process of annealing in metallurgy. Annealing involves heating and cooling a material to alter its physical properties. Initially, the material is heated to a specific temperature and then the material is allowed to cool. As the material cools, the atom's random movement decreases and they rearrange to attain the lowest energy state possible, thus forming a crystalline structure. The duration of the cooling process is an important factor as it affects the formation of the crystalline structure. If the cooling is so fast, the atoms reach minimum energy state but are arranged improperly which affects the structural properties of the crystalline material.

SA uses a stochastic approach to direct the search. In other words, it guides the local search in the following way: if S is the current state in the system and $\gamma(S)$ is the corresponding objective function value, then a perturbation to a new neighbouring state, S' is always accepted if it decreases the objective function, i.e. $\Delta\gamma = \gamma(S') - \gamma(S) \leq 0$. If the new state increases the objective function it will be accepted with probability, $P(\Delta\gamma, \mathrm{T}) = exp\left(\frac{-\Delta\gamma}{T}\right)$ where T is an external parameter called the "temperature". The value of T varies from a relatively large value to a small value close to zero. These values are controlled by a cooling schedule which specifies the initial and instant temperature values at each stage as the iteration progresses. When the temperature is high, stochastic influence is strong, but as the temperature decreases the process gradually turns into a deterministic one. It is this feature that helps SA to escape from local optima.

In order to apply the SA to a particular problem, the state space, the neighbouring state transition scheme, the probability transition function and the annealing schedule must be specified. The choices on these parameters have a significant impact on the method's effectiveness.

3.1. Advantage of SA

Let us consider the hill climbing problem as depicted in figure 2. In this problem, there are many local optima. A hill climber algorithm will simply accept neighbouring solutions that are better than the current solution. When the hill climber algorithm cannot find a better neighbour, it stops. A hill climber algorithm starting at a point on the hill (indicated by a red arrow in the left part of the figure 2) works its way to the top of the hill until it reaches a point (indicated by a red arrow in the right part of the figure 2) where it cannot climb any further without descending. Clearly, the hill climbing algorithm is struck at a local maximum which it considers as a global maximum.

Simulated Annealing (SA) works slightly different than the hill climbing algorithm. SA will accept worse solutions (a lesser function value in this problem) based on a probability. It is this feature of SA that helps it to escape from the local optima.

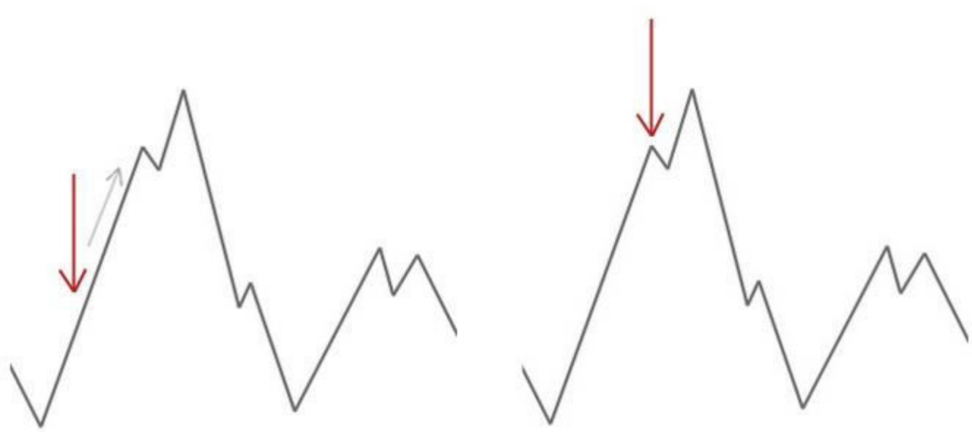

Figure 2 Hill climbing problem

4. Network Topology

A simplified network model used in this work is shown in figure 3. The model consists of five APs randomly deployed in an area. Each AP serves several users that are randomly distributed within a certain communication radius. In figure 3, the APs 1 and 2 are within the communication range of each other. Also, APs 3 and 4 are within each other's communication range. AP 5 does not interfere with any of the other four APs.

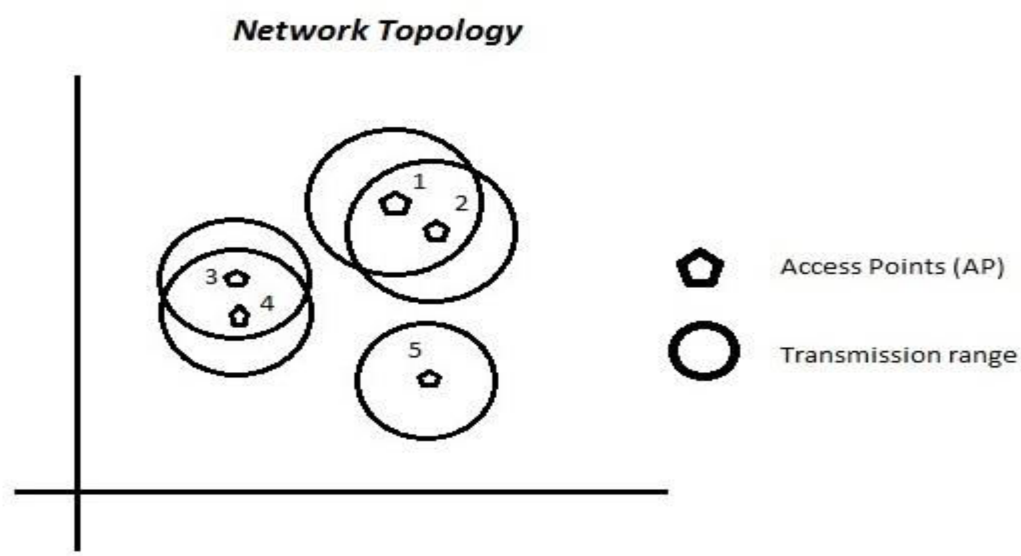

Figure 3 Network topology

5. Path loss calculation

In order to accurately model a wireless environment, the COST 231 Walfisch – Ikegami model [4] has been adopted. It presents a path loss formula that considers diffraction and reflection of urban building groups. The expression is given by $L_{path} = L_{rts} + L_{msd} + 20log_{10}(d)$, where L_{rts} is the rooftop to street diffraction and scatter loss, which takes city structure into account, such as height of the buildings, width of the roads, building separation and road orientation, $L_{msd}$is the multi-screen diffraction loss and d is the distance between the APs.

The rooftop to street diffraction is given by the following equation:

$$L_{rts} = -16.9 - 10log_{10}(w) + 10log_{10}(f) + 20log_{10}\left(h_{roof} - h_{RX}\right) + l_{ori}$$

$$with\ l_{ori} = \begin{Bmatrix} -10 + 0.354\phi\ for\ 0\ \text{deg} \leq \phi\ \leq 35\ deg \\ 2.5 + 0.075(\phi - 35)\ for\ 35\ \text{deg} \leq \phi \leq 55\ deg \\ 4.0 - 0.114(\phi - 35)\ for\ 55\ \text{deg} \leq\ \phi\ \leq 90\ deg \end{Bmatrix}$$

where w is the width of the roads in meters, h_{roof} is the rooftop height in meters,

h_{RX} is the receiver height in meters, φ is the road orientation,

l_{ori} is the orientation loss and f is the frequency of operation in MHz

The multi-screen diffraction loss is given by:

$$L_{msd} = l_{bsh} + k_a + k_d log_{10}(d) + \ k_f log_{10}(f) - 9log_{10}(b)$$

$$with\ l_{bsh} = \begin{Bmatrix} -18\left(1 + \left(h_{Tx} - h_{roof}\right)\right)\ for\ h_{Tx} >\ h_{roof} \\ 0\ for\ h_{Tx} <\ h_{roof} \end{Bmatrix}$$

$$k_a = \begin{Bmatrix} 54\ for\ h_{Tx} > h_{roof} \\ 54 - 0.8\left(h_{Tx} - h_{roof}\right)\ for\ d \geq 0.5km\ and\ h_{Tx} \leq h_{roof} \\ 54 - 0.8\left(\left(h_{Tx} - h_{roof}\right)\frac{d}{0.5}\right)\ for\ d < 0.5km\ and\ h_{Tx} \leq\ h_{roof} \end{Bmatrix}$$

$$k_d = \begin{Bmatrix} 18\ for\ h_{Tx} > h_{roof} \\ 18 - 15\left(\frac{h_{Tx} - h_{roof}}{h_{roof} - h_{Rx}}\right)\ for\ h_{Tx} <\ h_{roof} \end{Bmatrix}$$

$$k_f = \ -4 + \begin{Bmatrix} 0.7\left(\frac{f}{925} - 1\right)\ for\ medium\ sized\ city\ and\ suburban\ centers \\ 1.5\left(\frac{f}{925} - 1\right)\ for\ metropolitan\ centers \end{Bmatrix}$$

where the factors k_a and k_f control the dependence of the multi − screen diffraction loss versus the distance and the radio frequency. The factor k_a indicates the increase of the path loss for base stations below the rooftop.

6. Problem formulation

The set of all APs is denoted by A, and the set of all available channels is denoted by C. Matrix $\rho : C \times A \longrightarrow [0,1]$ is defined in the following way,

$$\rho(i,a) = \begin{Bmatrix} 1 \; if \; AP \; a \; uses \; channel \; i \\ 0 \; if \; AP \; a \; does \; not \; use \; channel \; i \end{Bmatrix}$$

$$where \; i \in \{1,2 \ldots . |C|\}, a \in \{1,2 \ldots . |A|\}$$

As each AP can operate in one channel at a time, each column of ρ contains only one '1' and $|C| - 1$ '0'. $\Omega_{|A|\times|A|}$ is an interference coefficient matrix. $\Omega_{a,b} = 1$ if AP a and AP b uses same channel and they are within each other's communication range, otherwise it is set to 0. The path loss $L_{path}(dB)$ is converted into link gain $G(mW)$ that captures the power loss on the path between the two APs.

Hence, the interference experienced by AP a, operating on channel i, from its neighbouring APs b with transmit power P_t (set to 1 for all APs) is given by

$$I_a(i) = \sum_{b \in NB(a)} \Omega_{a,b} G_{ab} P_t + \eta_a$$

$$where \; NB(a) \; defines \; a \; local \; environment \; around \; the \; AP \; a$$

$$but \; not \; including \; AP \; a \; itself,$$

$$\eta_a \; is \; the \; background \; noise \; experienced \; by \; AP \; a.$$

The objective of the channel allocation problem is to find a proper channel assignment, $S = (s_1, s_2 \ldots \ldots s_{|A|})$ such that the total interference $\gamma(S)$ is minimized, while each component in vector S can take any channel number (1, 6 or 11).

The objective function is formulated as follows:

$$Minimize \; \gamma[S] = \sum_{a \in A} I_a(s_a) = \sum_{a \in A} \sum_{b \in NB(a)} \Omega_{a,b} G_{ab} P_t + \eta_a$$

$$\text{Subject to } \sum_{a=1}^{|A|} \sum_{i=1}^{|C|} \rho(i, a) = |A|$$

7. Algorithm implementation

The quality of the results obtained by SA depends on several parameters. The following subsections will shed light on the selection of parameters for SA to solve the channel allocation problem in IEEE 802.11 b/g WLANs.

7.1. Neighbouring state transition

A neighbouring state is defined as a new channel assignment produced from its previous assignment by one or several APs selecting new channels. During, the transition, the target AP could switch to a channel randomly or to a channel that has not been used by its neighbouring APs if possible. In this work, the APs are allowed to choose a channel randomly in all transitions.

The details of the algorithm are described as follows:

a) Initialize each AP randomly with a channel.
b) Calculate the total interference in the system for the specific channel assignment.
c) Find a neighbouring state by allowing all the APs to choose a channel randomly.
d) Calculate the total interference after the transition.
    - If $new\ \gamma(S) < old\ \gamma(S)$, then keep the channel assignment
    - Else if $new\ \gamma(S) \geq old\ \gamma(S)$, then accept the channel assignment with probability $min\left(1, exp\left(\frac{-\Delta\gamma}{T}\right)\right)$
e) Update T according to the cooling schedule.
f) Repeat steps c - e until an acceptable channel assignment is found or a maximum number of iterations has been reached.

7.2. Temperature initialization

For better optimization, when initializing the temperature variable we should select a temperature that will initially allow for practically any move against the current solution. This gives the SA algorithm the ability to better explore the entire search space before cooling and settling in a more focused region.

7.3. Cooling schedule

Cooling schedule control the way in which the temperature is decreased. It has a major impact on convergence rate and solution quality. Generally, if the temperature is decreased quickly, then the algorithm converges fast, but the final solutions will tend to get worse. On the other hand, slow cooling will make the algorithm slow but will give better solutions. If a system of 50 APs and 3 channels is taken, there are 3^{50} channel assignments possible and each channel assignment has total interference level. Therefore, slower cooling can provide more opportunities to hit good channel assignment. The effect of cooling schedule on the convergence rate for 5 AP, 3-Channel network system will be discussed in the results section. The cooling schedule considered in this work is given by

$$T = T \times alpha \text{ } where \text{ } alpha \in (0,1) \text{ } is \text{ } the \text{ } cooling \text{ } factor$$

7.4. MATLAB code

The MATLAB R2014b code for solving the channel assignment problem in IEEE 802.11 b/g WLANs by implementing the SA algorithm is given below:

```
function [bestsol,fmin] = Final(alpha)
clc
clear
close all
if nargin<1
    alpha = 0.95;% Cooling factor
end
% Initializing parameters and settings
T_init= 1; % Initial temperature
T = T_init; % Temp variable
T_min = 1e-10; % Minimum temperature
max_rej = 2500; % Max # of rejections
max_run = 500; % Max # of runs
max_accept = 250; % Max # of acceptances
s= randi(3,1,5); % Channels
i= 0;j=0; accept = 0;
G = rand (5,5);
G(logical(eye(size(G)))) = 0;
G = G./100;
E_init = fun(s,G);
E_old = E_init; count = 0;
best = s; fminimum = []; counter = [];
while ((T>T_min) || (j<=max_rej))
    i = i+1;
    %Check if max numbers of run/accept are met
    if(i>=max_run) || (accept >=max_accept)
        %reset the counters
        i=1; accept =1;
        %Cooling according to a cooling schedule
        T = cooling(alpha,T);
```

```
    end
    %new solution for channel assignment for all APs
    ns=randi(3,1,5);
    E_new = fun(ns,G);
    % Decide to accept new solution
    %Accept if improved
    DeltaE = E_new - E_old;
    if(E_new < E_old)
        best = ns; E_old = E_new;
        accept = accept + 1; j=0;
    end
    p= min(1,exp(-DeltaE/T)); % Probability of
accepting new solution
    %Accept with a small probability p if not improved
    if ((E_new >= E_old) && (p > 0))
        best = ns; E_old = E_new;
        accept = accept + 1;
    else
        j=j+1;
    end
%Update the estimated optimal solution
fmin = E_old;
count = count + 1;
fminimum(count) = fmin; %#ok<AGROW>
counter(count) = count;  %#ok<AGROW>
%display(fmin)

end
plot (counter,fminimum)
xlabel ('Iteration');
ylabel ('Interference');
title ('Variation in interference')
display(count)
display(E_init)
bestsol = best;
bestfunctionvalue = fmin;
display(bestfunctionvalue)
end

function I=fun(s,G)
E_norm = 25; % Normalization
omega = zeros([5 5]);
if (s(1,1) == s(1,2))
    omega(1,2) = 1;
    omega(2,1) = 1;
end
if(s(1,3)==s(1,4))
```

```
        omega(3,4)=1;
        omega(4,3)=1;
    end
    rho = zeros([3 5]);
    for y=1:5
        switch s(1,y)
            case 1
                rho(1,y)=1;
            case 2
                rho(2,y)=1;
            case 3
                rho(3,y)=1;
        end
    end
    noise = rand(1,5);
    noise= sum(noise);
    I=omega.*G + noise;
    I(logical(eye(size(I)))) = 0;
    I= sum(I);
    I=sum(I,2)/E_norm;
    end

    function T=cooling(alpha,T)
    T=alpha*T;
    end
```

The code consists of three functions namely Final (alpha), fun(s, G) and cooling (alpha, T). The function Final (alpha) contains the code for initializing the parameter variables by assigning values to it, the Simulated Annealing algorithm and the function calls for other two functions. The function fun (s, G) contains the code for calculating the total interference in the system along with the equality constraint. The function cooling (alpha, T) contains the code for defining the cooling schedule. The function fun (s, G) is called for the first time to calculate the initial total interference level and also when a new channel assignment occurs to calculate the new total interference level. The function cooling (alpha, T) is called when the value of the temperature variable T has to be reduced.

8. Results

The SA algorithm was implemented for the network topology defined in section 4. It has been found that the algorithm provides optimal channel assignment in most number of trials (each trial has different link gain values). The command window output for two trials when $alpha = 0.95$ are shown below:

```
count = 178353

E_init = 0.9267

bestfunctionvalue = 0.1150

ans =

     2   1   3   1   2
```

Figure 4 Optimal channel assignment

```
count =  178272

E_init =   2.2049

bestfunctionvalue = 0.1677

ans =

     3   1   1   1   3
```

Figure 5 Sub-optimal channel assignment

In figure 4 and figure 5, the “count” represents the number of iterations taken to arrive at the respective solutions. “E_init” represents the value of the total interference power calculated after the initial channel assignment. “bestfunctionvalue” represents the final total interference power obtained for the respective final channel assignments. “ans” represents the channel vector S which contains the final channel assignments for the 5 APs.

The network topology is designed in such a way that the AP 1 and AP 2 are within each other’s communication range and also AP 3 and AP 4 are within each other’s communication range. Therefore, the first and second entries in “ans” should not have the same channel number. Likewise, the third and fourth entries in “ans” should not have the same channel number. If the above is true, then the SA algorithm has provided an optimal solution.

In figure 4, the channels assigned to the APs 1 – 5 are 2, 1, 3, 1 and 2 respectively. Therefore, the channel assignment is optimal. However, in a different trial, as depicted in figure 5, the channels assigned to the APs 1 – 5 are 3, 1, 1, 1 and 3 respectively. The APs 3 and 4 are assigned the same channel. This channel assignment is sub-optimal. This proves the fact that the SA algorithm cannot provide optimal solution in all the trials.

It is shown in [1] that the SA algorithm can provide close to optimal solutions when it is tested for 1000 different network scenarios (figure 6). The optimal solutions are obtained using Branch and Bound algorithm. The SA algorithm can be used to find good channel assignments for network scenarios up to 40 APs. However, for scenarios which involves more than 40 APs, finding the global optimum for 3 channels within reasonable time does not seem to be possible. This is because, the computational time increases exponentially with the increase in the size of optimization problem.

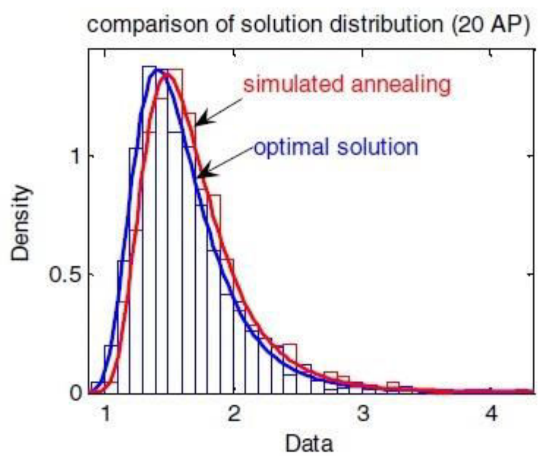

Figure 6 Comparison between SA and Branch and Bound algorithm (reproduced from [1])

8.1. Effect of cooling factor

In this subsection, the effect of cooling factor $alpha$ on the SA algorithm is explained. The SA algorithm is tested for two values of $alpha$ – 0.8 and 0.95. The rate of convergence of the algorithm and the solution quality depends on the cooling factor. If the value of cooling factor is low, the cooling process will be fast and vice-versa. The convergence rate of the algorithm will be fast when the value of $alpha$ is low. However, the quality of the solution is not guaranteed to be good. The convergence rate will be slow when the value of $alpha$ is high. Therefore, the quality of the solution will be good. The above described effect is shown in figures 7 and 8. It can be seen in figure 7 ($alpha = 0.95$) that the total interference converges to a value of 0.1341 when the number of iteration is close to 12E4. In figure 8 ($alpha = 0.8$), the total interference converges to a higher value of 0.2316 when the number of iteration is 2.6E4.

8.2. Future work

As future work, the SA algorithm can be tested for denser network scenarios (number of APs can be 10, 20, 30 and 40). Also, the algorithm can be tested using IEEE 802.11a devices which can use any one channel from a set of 12 non-overlapping channels.

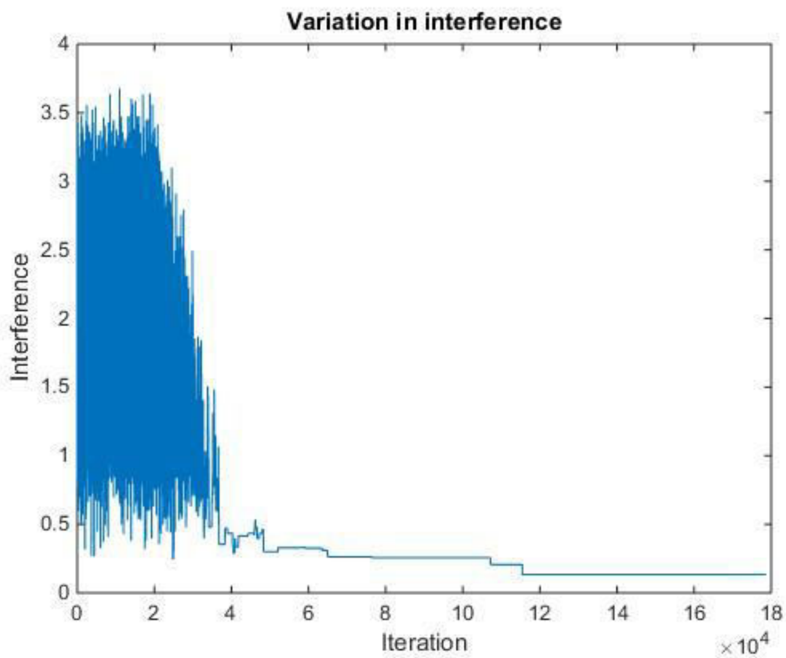

Figure 7 Fluctuation in total interference (alpha = 0.95)

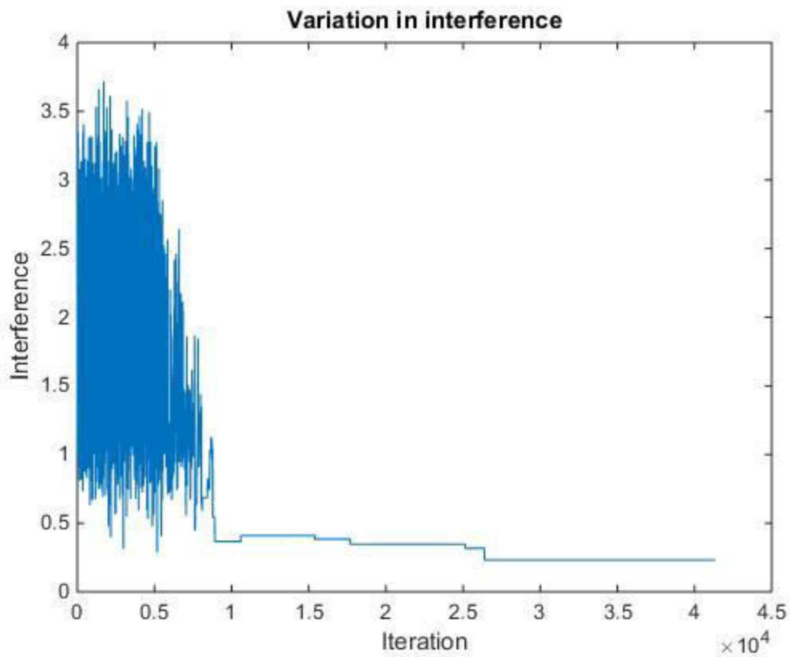

Figure 8 Fluctuation in total interference (alpha = 0.8)

9. Conclusion

In this project, the channel assignment problem in 802.11 b/g WLANs is solved by using Simulated Annealing algorithm. The network topology, the parameters such as the network state transition and cooling schedule required for SA implementation have been defined. The simulations are carried out and the channels are assigned to the APs such that the total interference in the system is minimized.

References

[1] Chen, J., Olafsson, S. and Gu, X. , ”Observations on Using Simulated Annealing for Dynamic Channel Allocation in 802.11 WLANs,” IEEE Vehicular Technology Conference, ISSN :1550-2252, E-ISBN : 978-1-4244-1645-5, Print ISBN: 978-1-4244-1644-8, pp:1801 - 1805, Singapore, May 2008.

[2] Kirkpatrick, S., Gelatt, C.D. and Vecchi, P. M., “Optimization by Simulated Annealing,” Science, Vol. 220, pp. 610-680, 1983.

[3] Laarhoven, P.J.M.V. and Aarts, E.H.L., ”Simulated Annealing: Theory and Applications,” ISBN 978-90-481-8438-5, E-ISBN 978-94-015-7744-1.

[4] Cost 231, Final Report, 1999.

[5] Rutenbar, R.A., “Simulated Annealing Algorithms : An Overview,” IEEE Circuits and Devices Magazine, Vol. 5, ISSN : 8755 – 3996, pp: 19 – 26, January 1989.