| | |
|---|---|
| Title | New graph colouring algorithm for resource allocation in large-scale wireless networks |
| Author(s) | Abdullah, Labeeb Mohsin; Baba, Mohd Dani; Ali, Sinan Ghassan Abid; Lim, Azman Osman; Tan, Yasuo |
| Citation | 2014 IEEE 5th Control and System Graduate Research Colloquium (ICSGRC): 233-238 |
| Issue Date | 2014-08 |
| Type | Conference Paper |
| Text version | author |
| URL | http://hdl.handle.net/10119/13480 |
| Rights | This is the author's version of the work. Copyright (C) 2014 IEEE. 2014 IEEE 5th Control and System Graduate Research Colloquium (ICSGRC), 2014, 233-238. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. |
| Description | |

# New Graph Colouring Algorithm for Resource Allocation in Large-scale Wireless Networks

Labeeb Mohsin Abdullah, Mohd Dani Baba, Sinan Ghassan Abid Ali
Centre for Computer Engineering Studies
Faculty of Electrical Engineering
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
labeeb966@gmail.com, mdani074@salam.uitm.edu.my, sinan79alnasir@gmail.com
Azman Osman Lim, Yasuo Tan
School of Information Science
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan
{aolim, ytan}@jaist.ac.jp

***Abstract*—The vertex-colouring problem is a well-known classical problem in graph theory in which a colour is assigned to each vertex of the graph such that no two adjacent vertices have the same colour. The minimum vertex-colouring problem is known as NP-hard problem in an arbitrary graph. In this paper a graph colouring algorithm based on modified incidence matrix is proposed for resolving Physical Cell ID (PCI) allocation for large-scale femtocell deployment in LTE Telecommunication Networks. The proposed algorithm is not specified for neighbours only, but additionally can deal with neighbours of neighbours' objects due to telecommunication requirements. Our results show that by applying proper searching and assigning methods it is possible to achieve satisfactory results for resource allocation in large and complex networks such as resolving PCI allocation and conflict for large femtocells deployment in LTE Networks.**

***IndexTerms*— Graph colouring, PCI, collision, confusion, LTE.**

## I. INTRODUCTION

The vertex-colouring problem (VCP) is a well-known combinatorial optimization problem in graph theory. A legal vertex colouring of graph G=(V,E),where V(G) is the set of |V|=n vertices and E(G) is the edge set including |E|= m edges, is a function f :V → C from the vertices of the graph G to the colour-set C={c1,c2,…, cp} such that $f(u) \neq f(v)$ for all edges $(u, v) \in E$. That is a legal vertex colouring of G is assigning one of the *p* distinct colours to each vertex of the graph in such a way that no two endpoints of any edge are given the same colours. Formally the vertex-colouring problem can be considered either as an optimization problem or as a decision problem [1].The development of graph theory is very similar to the development of probability theory where the large portions of graph theory have been motivated by the study of games and recreational mathematics.

Since a graph is a very convenient and natural way of representing the relationships between objects it used in many application .Examples of such applications; chemical molecule, map colouring, signal-flow graphs, Konigsberg bridges, tracing maze etc. [2]. For networks the graph can be formulated as mathematical representation to reflect the network connections and objects. Each object of the graph is called a node, vertex or simply a hop, while the corresponding connections in a network represent edges (or links) in a graph. Each edge in a graph joins two distinct nodes [3].In Telecommunication Networks applying graph theory algorithms is most needed, thus in this work we use Telecom network as a case study to apply our algorithm to provide improved resource management to the network.

## II. GRAPH COLOURING USE – CASE FOR TELECOM NETWORK

There are many applications for Graph colouring such as Time Tabling and Scheduling, Register Allocation, Printed Circuit Board Testing I – Colouring, Printed Circuit Board Testing II – Clique, Pattern Matching, Analysis of Biological and Archaeological Data [4]. However, in the area of telecommunication field graph theory used in the following issues:

### *A. Frequency Assignment*

Frequency Assignment is a problem of assigning frequencies to mobile radios and other users of the electromagnetic spectrum. In the simplest case, two customers who are sufficiently close to one another must be assigned different frequencies, while those that are distant away can share frequencies. The problem of minimizing the number of frequencies used can be categories as a graph colouring problem [4].

### *B. Physical Cell ID Resource Allocation in LTE*

In the telecommunication network such as LTE, the physical cell ID (PCI) is needed to distinguish the signal of one Base Station (BS) from the signal of another BS. Moreover to be more challenging in LTE networks there is a small cell known as femtocell that can be deployed in thousands with scarce resources in which only 504 PCI are on hand, thus reuse the available 504 is inevitable. Accordingly, neighbouring BSs or femtocells should not use the same PCI. In addition to avoid confusion in Hand-Over (HO) a cell should not have neighbours and neighbours of neighbours with the identical PCI. Thus the PCI configuration problem can be turned into a graph-colouring problem for the graph of two-hop neighbours [5].

## III. Previous Works

Many works have been done [6-10] in the field of telecommunications to tackle the two aforementioned problems, i.e. Frequency Assignment and PCI Allocation in LTE. However there is still a room for improvements. Here we review some of these works that discussed the PCI assignment and conflict [11, 12].Starting with Jae Seung Song et al. Where the authors focused in their proposed work [6] on one of management problems, which is the self-configuration of PCIs for femtocell where they proposed a model-checking to help formal correctness verification to verify the correctness of the Self-configuration assignments process. This approach has the ability of searching for error in the configuration starting from partial configuration state snapshot. Yet their work has been done on small sample that did not reflect realistic case where many drawbacks could appear under larger deployment. Moreover their algorithm has failed under the concurrent femtocells deployment. Another wok proposed by T. Bandh and G. Carle [7] where they proposed a solution for the problem of PCI interference connected to Co-tier LTE networks, this solution developed for the PCI auto-configuration by using colouring-based mathematical method (graph colouring) where each PCI assigned to a colour that different from its neighbour i.e. different ID based on the theory of graphic colouring. A work by M. Amirijoo et. al. [8] suggested to update the Neighbour-Cell-List (NCL) in the network of LTE single-tier by using UE's measurements to discover PCI interference where the idea is if the PCI interference appears the UE will send this information to the Core Network (CN), after that the Operation Support System (OSS) will request the ID of the femtocells to change their PCIs. However, the above two solutions cannot be considered completed due to that the proposed solutions solved only one part of the problem which is Co-tire network and not whole heterogeneous network that include macrocell and femtocell (cross- tier network).

## IV. Conventional Graph Colouring Algorithm Challenges

In the conventional graph colouring algorithms it considered that no two adjacent vertices have the identical colour, while in the case of PCI no adjacent neighbours(N) and neighbours of neighbours(N of N) have the same PCI (i.e. the same colour). To shed some light on the challenges that come along with PCI assignment let us consider the network in Fig. 1.The network has 10 nodes (vertices). By using conventional graph colouring algorithm it is relatively easy to satisfy the constraint of non-adjacent nodes have the alike colour. But the problem gains more complexity when the constraint becomes: no (N) or (N of N) share the same colour. In this case the size of its Neighbour List (NL) matrix $M_{m \times n}$ that must be considered is increase from 10×2 to 42×2, and if we assume the same increasing factor applied to another network with 1000 nodes, that means the size of considered matrix jump to 42,000×2, in this case two challenges could emerge, first: how to extract large-scale networks to have their N and (N of N) to consider them in the solution without help from the system itself? However in LTE network the system provides this information by exchanging messages between nodes BSs (Marcocell or femtocells) to update N and (N of Ns') information and list them by using X2 interface links [13]. But for other applications if any disconnection occurs in one of these links for any reason (down links, etc.), this might lead to conflict between the components of this network. The second challenge is how to deal with the resulted large matrix or array of N plus (N of N) and track their status and location precisely. In fact, this can be a quite tricky task.

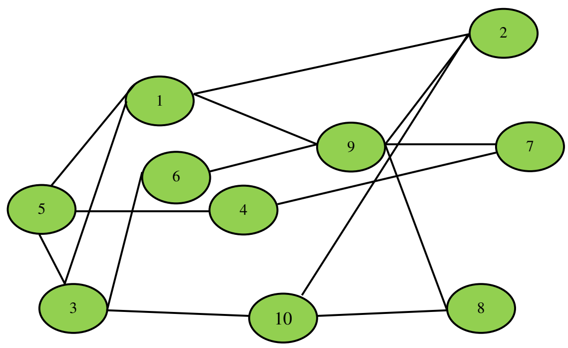


Fig. 1. Example of simple network

### A. Proposed Algorithm Framework

The main framework of our proposed algorithm is designed to avoid the weak points of conventional graph colouring algorithms when it comes to deal with telecom network .i.e. dealing with complicated and large graph efficiently. Our algorithm is inspired by a type of matrix called *Incidence Matrix.* The incidence matrix of a graph gives the (0, 1) matrix, which has a row for each vertex and column for each edge and (V, E) = 1 if vertex V is incident with an edge [14]. Furthermore, incidence matrix can be defined as a matrix with a column for each vertex and a row for each edge, this matrix has the ability of representing a graph vertices relationship. Fig. 2 shows an example of incidence matrix representing a network graph [15].

$$A_{J,K} = \begin{cases} 1 & \text{if vertix J is an end of edge } E \\ 0 & \text{Otherwise} \end{cases}$$

$$A_{J,K} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

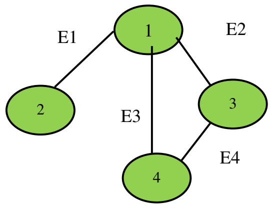


Fig. 2. The network graph and its incidence matrix

By looking at the incidence matrix of Fig. 2, one can have all the needed information of the graph, for example row number 1 in the matrix "A" represents the node or (BS) number 1 that is connected to node 2 via edge 1(E1), node 3 via (E2) and node 4 via (E3) and so on for the rest of the nodes. Taking benefit from the Incidence Matrix structure, the new algorithm converts the Graph Colouring problem to Constraint Problem Solving (CPS). Thus for the problem of PCI the constraint is: for three consecutive vertices no vertex is allowed to share the same colour with the other two adjacent neighbour vertices. The reason of choosing the number "three" is that the "3GPP" organization, suggested that to ensure free collision and confusion in the LTE system the PCI ID must not be the same for N and (N of N) for any PCI that could represent BS (femtocell or macrocell) or both [16].In addition there are another requirements to have free conflicts during the process of PCI assignment that already specified in [16],yet these requirements beyond the focus of this work.

*B. Modifying the Incidence Matrix*

As the standard incidence matrix allows only two vertices to be represented in each column, using this matrix with large graph leads to produce a very big size matrix, thus a smaller matrix is needed to represent the graph. By using the same concept of incidence matrix the modified matrix can use unlimited vertices in the same column.

*C. The Proposed Algorithm Steps*

We call this algorithm as Modified Incidence Matrix (MIM)-based Graph Colouring. The algorithm performs the following steps:

- **Step 1**: In this step, the algorithm reads the NL. Next it can choose to extract the (N of N) from the graph or to stick with the conventional CPS of checking the two adjacent neighbours only. To extract the (N of N) from the main NL, consider a matrix M with m=4, n=2. Where vertex 1 is connected to its neighbour vertex 2, while the (N of N) of 1 is: 3, 6 and 10, thus the vertex 1's N and (N of N) can be written as: {1-2, 1-3, 1-6, 1-10}.

$$M = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 6 \\ 1 & 10 \end{bmatrix}$$

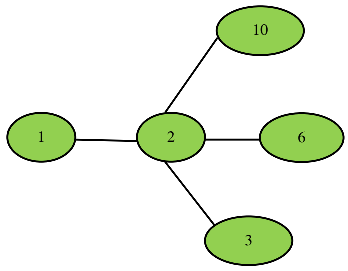


Fig. 3. Network example with its N and (N of N) plus matrix for vertex #1

The ability of extracting the list for (N of N) is one of the advantages of this algorithm, for example in the case of no information for the (N of N) in the network (for any reason), the network still able to do its duty without any hazard of conflicts. That can be done by using N information only, the second advantage of this ability is aimed for researchers. For instance, if a test is needed to be perform in the lab, and it is necessary to have the list of N and (N of N), especially for large-scale network it is easy to use this part of the algorithm to find such data instead of using random numbers that do not reflect the real topology of the network under study. Fig. 5 shows the pseudo code for (N of N) extracting.

- **Step 2**: Here the algorithm reads the maximum value existed in the NL to create a matrix of $m \times m$, where $m$ is the maximum value in the NL. For example the maximum value in the graph of Fig. 2 is 4, thus $m$=4, after that the algorithm follows the manner of the incidence matrix as Fig. 2 with one difference, our algorithm uses the modify matrix approach by assigning more than two vertices in the same column if needed.
- **Step 3:** After representing each vertex and its adjacent Ns in the connection matrix the algorithm then checks the matrix column by column to satisfy the CPS condition (i.e. No two N or (N of N) shares the same value). However the checking process is performed for the NL (N or and N of N) only and not for all the connection matrix that has zeros as well. For instance, from Fig. 3, vertex #1 has the following N, N= {1-2, 1-3, 1 6, 1- 10}; the algorithm checks position (1,1) with (1,2) in the matrix, then if the two position has the same value (we chose the #1 as initial value for all the vertices), then the value in position (1,2) must to be changed by adding one to the target vertex . Checking neighbours with such method adds extra advantage to our proposed algorithm because instead of reading and checking all the values in the matrix the algorithm checks the targeted neighbours directly without wasting the time and the CPU's resources. Moreover it is an accurate way to follow each vertex's neighbour precisely. Next after no more CPS condition can be satisfied the algorithm unifies the values of each row in the matrix by using the minimum value existed in each row, because the value of vertex cannot be fluctuated where vertices have only one value that can represented in the graph if this value conflicted with other rows, this value will increase /decreased by one until no conflict occur among the rows, by doing this we can ensure assigning minimum number of colours. After that the algorithm starts checking again until no more two adjacent N or (N of N) share the same value (or colour).Table I depicts the incidence matrix in Fig. 2 where V# represents the vertex number.

TABLE I.VERTEX MATRIX FOR THE INCIDENCE MATRIX OF FIG.2.

| | | | | |
|---|---|---|---|---|
| **V#1** | 1 | 1 | 1 | 0 |
| **V#2** | 2 | 0 | 0 | 0 |
| **V#3** | 0 | 2 | 0 | 2 |
| **V#4** | 0 | 3 | 3 | 3 |

### *D. Pseudo code of the Proposed Algorithm*

Pseudo code of this algorithm is given in Fig. 6.

```
Initialization:

k=1:m,% number of rows
w1=w2=w3=1%counterto help moving through
the rows in matrix.
Column always =2.
Begin

if NL(row,column+1)==NL(row+k,column+1)
X1=NL(row,column);
X2=NL(row+k,column);

X_1(w1,column)=X1;%to store the new
matrix.
X_1(w1,column+1)=X2;

NL1=[NL,X_1]%new (N of N).

end

if NL(row,column+1)==NL(row+k,column)
X3=X1(row,column);
X4=X1(row+k,column+1);

X_2(w2,column)=X3;
X_2(w2,column+1)=X4;

NL2=[NL,X_2]
end

if X1(row,column)==X1(row+k,column)
X5=NL(row,column+1);
X6=NL(row+k,column+1);
X_3(w3,column)=X5;
X_3(w3,column+1)=X6;
NL3=[NL,X_3]
end
NL=[NL;NL1;NL2;NL3];%the new NL matrix.
```

Fig.5.Pseudo code to extract (N of N) list

## V. NUMERICAL RESULTS

The main objective for our algorithm is not only using minimum number of colours (PCIs) where there are many others good and robust algorithms, but also to fill the gap of the lack of controlling the available resource. The proposed algorithm has the following two goals:

```
Initialization:

Choose one of the two options:
   1- Run N.
   2- Run for N and(N of N).

if choice =1 ,then L=N
             else L= N & N of N

       - Choose a value for y %the
       available number of colors.
       - Initialize Counter =1
       - MV_max = maximum value in the N
       list
- V=0;initial value of vertex
Begin

While counter >0
Counter =0
Create a matrix M=MV_max × MV_max
Assign "1" to each connected pair of
        vertices,"0"Otherwise
for all M
While two vertices share the same value;
  Add"1" to the second vertex value
       V,(V+1), 1≤ V≥ y;
Counter =counter+1;
Unify Vertices values in each row;
       end
end
end
Print Out resulted matrix M
end
```

Fig.6. The pseudo code of the proposed algorithm.

- To minimize the number of Iteration needed by the network to converge and reach its study state.
- To minimize the number of PCIs that needed to be reassigned. This because reassigning the already on air PCI for operating femtocell might lead to network oscillation and consequently service disturbance and service experience degradation.

To test our proposed algorithm, we introduced the network to variety of link density levels to check the robustness of our algorithm by comparing its performance with Back Tracking algorithm [17]. The reason of using an algorithm such as Back Tracking is that both algorithms work under the concept of centralization. The random networks generated by using Mathematica software [18]. To calculate the density of the generated random network we used the equation [19]:

$$D = \frac{2|E|}{|V|\ (|V| - 1)}$$

Where D and E are the graph density and the number of edges respectively, while "2" represents the two-way connection of the edge. V represents the vertices number. Table II shows the used vertex and the applied density for each network.

TABLE II. USED VERTEX AND APPLIED DENSITY ON EACH NETWORK

| Vertex (V) | Applied Density |
|---|---|
| 200 | $D_{200}$=[0.001,0.002,0.003………0.01] |
| 400 | $D_{400}$=[0.001,0.002,0.003………0.01] |
| 600 | $D_{600}$=[0.001,0.002,0.003………0.01] |

After substituting D and V into the equation we can obtain the values of edges (E) for the direct neighbours, yet the relation of neighbours of neighbours (NN) for each femtocell also needed to insure network confusion free, However our algorithm has the ability of analyzing the (NN) network relations from the introduced network without assuming random NN as usually done in most of the previous works. Table III shows the results of the comparison between our algorithm and the Back Tracking algorithm.

TABLE III. COMPARISON BETWEEN OUR PROPOSED ALGORITHM AND BACKTRAKING ALGORITHM

| V | D | E-N | E-NN | It-Pr | It-BT | RA-Pr | RA-BT |
|---|---|---|---|---|---|---|---|
| 200 | 0.001 | 20 | 1 | 1 | 2 | 1 | 2 |
| 200 | 0.002 | 40 | 7 | 1 | 3 | 1 | 4 |
| 200 | 0.003 | 60 | 31 | 1 | 3 | 1 | 1 |
| 200 | 0.004 | 80 | 68 | 2 | 3 | 1 | 2 |
| 200 | 0.005 | 100 | 94 | 1 | 3 | 1 | 2 |
| 200 | 0.006 | 120 | 132 | 1 | 3 | 1 | 2 |
| 200 | 0.007 | 140 | 191 | 2 | 3 | 1 | 17 |
| 200 | 0.008 | 160 | 229 | 1 | 4 | 1 | 10 |
| 200 | 0.009 | 180 | 336 | 2 | 4 | 1 | 19 |
| 200 | 0.01 | 200 | 426 | 2 | 3 | 1 | 13 |
| 400 | 0.001 | 80 | 60 | 1 | 2 | 1 | 2 |
| 400 | 0.002 | 160 | 160 | 2 | 3 | 1 | 4 |
| 400 | 0.003 | 240 | 254 | 2 | 3 | 1 | 3 |
| 400 | 0.004 | 320 | 452 | 1 | 4 | 1 | 7 |
| 400 | 0.005 | 400 | 828 | 1 | 5 | 1 | 9 |
| 400 | 0.006 | 480 | 1232 | 1 | 4 | 1 | 17 |
| 400 | 0.007 | 560 | 1712 | 2 | 5 | 1 | 19 |
| 400 | 0.008 | 640 | 2249 | 1 | 6 | 1 | 100 |
| 400 | 0.009 | 720 | 2893 | 2 | 6 | 1 | 4 |
| 400 | 0.01 | 800 | 3256 | 2 | 6 | 1 | 107 |
| 600 | 0.001 | 180 | 109 | 2 | 3 | 1 | 5 |
| 600 | 0.002 | 360 | 181 | 1 | 4 | 1 | 3 |
| 600 | 0.003 | 540 | 967 | 1 | 5 | 1 | 1 |
| 600 | 0.004 | 720 | 1874 | 2 | 3 | 1 | 112 |
| 600 | 0.005 | 900 | 2939 | 2 | 5 | 1 | 1 |
| 600 | 0.006 | 1080 | 4326 | 2 | 7 | 1 | 113 |
| 600 | 0.007 | 1260 | 5702 | 1 | 8 | 1 | 4 |
| 600 | 0.008 | 1440 | 7482 | 1 | 8 | 1 | 129 |
| 600 | 0.009 | 1620 | 8013 | 2 | 9 | 1 | 78 |
| 600 | 0.01 | 1800 | 9221 | 2 | 9 | 1 | 6 |

Where:

V: Vertices.
D: Density.
E-N: Number of Edges for direct vertices with direct neighbour's relation.
E-NN: Number of Edges that connect vertex with its neighbours of neighbours.
It-Pr: the number of iteration for the proposed algorithm.
It-BT: the number of iteration for the Backtracking algorithm.
RA-Pr: Reassigned PCIs for the proposed algorithm.
RA-BT: Reassigned PCIs for the Backtracking algorithm.

To have clearer view on the results, Table IV shows the average of the outputs in Table III.

TABLE IV.THE AVARAGE OF THE COMPARISON RESULTS BETWEEN THE PROPOSED ALGORITHM AND BACKTRACKING ALGORITHM

| V | D | E-N | E-NN | It-P | It-BT | RA-P | RA-BT |
|---|---|---|---|---|---|---|---|
| 200 | 0.0055 | 110 | 151.5 | 1.4 | 3.1 | 1 | 7.2 |
| 400 | 0.0055 | 440 | 1309.6 | 1.5 | 4.4 | 1 | 27.2 |
| 600 | 0.0055 | 990 | 4081.4 | 1.6 | 6.1 | 1 | 45.2 |

From Tables III and IV it is clear that our proposed algorithm outperform Backtracking algorithm where that can be attributed to the stricture of our algorithm where the assigned PCI kept in a matrix that accessible by the algorithm , thus each time a new PCI assigned then the algorithm knew where is the exact position of this PCI regarding to the other PCIs positions .Next the algorithm assigns a PCI that unique in its area for this very position only without changing the already assigned PCIs. In contrast the other algorithms such as Backtracking assign a PCI randomly and when a conflict recognized then the algorithm will change the PCI that caused the conflict regardless if it is already on-air or not .

## VI. CONCLUSION AND FUTURE WORK

The initial evaluations on our proposed algorithm MIM has shown encouraging results, especially in dealing with large-scale network. This algorithm also provides more reliability to the network by employing its ability of using minimum available data provided by the network; however, more tests will be conducted for different scenarios and cases for better reliability and to improve our work.

Acknowledgment

The authors would like to thanks UiTM and MOSTI, Malaysia for providing the research Science Fund under the Project No: 01-01-01-SF0551.

REFERENCES

[1] A. E. Eraghi, J. A. Torkestani, and M. R. Meybodi, “Cellular Learning Automata-based Graph Coloring Problem,” vol. 3, pp. 163–167, 2011.

[2] Kalid bin Mohammed (2012) Graph Theory Kent State Universitywebsite.[online].Available:ttp://www.personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/graphIntro.htm.

[3] Bruce Hoppe (2009) Graph Theory website.[online].Available :http://webwhompers.com/graph-theory.html.

[4] Michael Trick(1994) Tepper School of Business website.[online].Available:http://mat.gsia.cmu.edu/COLOR/color.html.

[5] F. Ahmed, O. Tirkkonen, M. Peltomäki, J.-M. Koljonen, C.-H. Yu, and M. Alava, “Distributed Graph Coloring for Self-Organization in LTE Networks,” *J. Electr. Comput. Eng.*, vol. 2010, pp. 1–10, 2010.

[6] F. Mhiri, K. Sethom, and R. Bouallegue : A survey on interference management techniques in femtocell self-organizing networks. In : Journal of Network and Computer Applications, vol. 36, no. 1, pp. 58–65, May 2012.

[7] T. Bandh and G. Carle :Graph Coloring Based Physical-Cell-ID Assignment for LTE Networks Categories and Subject Descriptors . pp. 116–120, 2009.

[8] M. Amirijoo, P. Frenger, F. Gunnarsson, H. Kallin, J. Moe, and K. Zetterberg . : Neighbor Cell Relation List and Physical Cell Identity Self-Organization in LTE. Wireless Access Networks,Ericsson Research, Ericsson AB, Sweden.

[9] A. Diab and A. Mitschele-Thiel: Comparative evaluation of distributed physical cell identity assignment schemes for LTE-

[10] advanced systems,” Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, vol. 12, p. 61, 2012.

[11] Y. Chang, Z. Tao, J. Zhang, and C. J. Kuo, “A Graph-based Approach to Multi-Cell OFDMA Downlink Resource Allocation,” 2008.

[12] L. M. Abdullah, M. Dani Baba, and S. G. A. Ali, “A novel scheme to resolve PCI conflicts and assignment problems in LTE-femtocell networks,” 2013 IEEE 3rd Int. Conf. Syst. Eng. Technol., pp. 109–112, Aug. 2013.

[13] L. M. Abdullah, M. D. Baba, S. G. A. Ali, “Self-configuration Concept to Solve Physical Cell ID Conflict for SON LTE-based Femtocell Environment,”ACEEE,vol. 10, no. 2,2014.

[14] LTE-femtocell networks,” 2013 IEEE 3rd Int. Conf. Syst. Eng. Technol., pp. 109–112, Aug. 2013.

[15] S. Sesia, I. Toufik, and M. Baker, LTE – The UMTS Long Term Evolution. John Wiley & Sons, Ltd, 2009.

[16] S. Skiena, Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Reading, MA: Addison-Wesley, pp. 135-136, 1990.

[17] 3GPP TS 32.500, "Telecommunication Management; Self-Organizing Networks (SON); Concepts and Requirements (ReI. 8), 2008.

[18] H.S. wILF.BreakTrack:An O(1) expected time algorithm for graph coloring processing Lett.18(1984) .p 119-122.

[19] Wolfram Mathematica website .[online].Available : http://www.wolfram.com/mathematica/

[20] Coleman, Thomas F.; Moré, Jorge J. (1983), "Estimation of sparse Jacobian matrices and graph coloring Problems", *SIAM Journal on Numerical Analysis* **20** (1): 187–209, doi:10.1137/0720013.