

A Distributed Approach for Frequency Allocation using Graph Coloring in Mobile Networks

Vinod Maan

Department of Computer Science
Faculty of Engineering & Technology
Mody Institute of Technology & Science
Lakshmangarh, India

G.N. Purohit

Department of Computer & Mathematical Sciences
Banasthali University
Jaipur, India

ABSTRACT

The proposed technique explores a general framework for studying distributed frequency allocation in mobile cellular network. The problem can be abstracted as a multicoloring problem on a cellular graph. This paper concerns a algorithmic problem for solving Maximal independent set of a graph in a distributed fashion and then reducing independent set to graph coloring. The objective of frequency allocation is to assign required number of frequency to each cell such that efficient frequency spectrum utilization is provided and interference effects are minimized [1].

General Terms

Frequency allocation, Multicoloring, graph coloring

Keywords

Frequency, Maximal independent set, Graph Coloring, Neighbors, Set, Vertex, and Edge

1. INTRODUCTION

The demand of cellular mobile service is increasing at a very high rate each year and in a lot of metropolitan areas the demand has already far exceeded the capacity. Different techniques can be used to increase the capacity. One of the best solution for this is the frequency reuse.

In particular, given the cellular graph of mobile cellular networks, imagine frequencies as colors then allocating frequencies to users of the network is equivalent to multi coloring of the nodes of the cellular graph [1, 2]. The coloring use minimum number of colors (i.e., the communication between users must be obtained with the use of minimum number of frequencies) and nodes of the cellular graph connected with an edge must be assigned different colors (i.e., users located close to each other according to the reuse distance of the network, must be assigned different frequencies).

DISTRIBUTED ALGORITHM

The study of distributed algorithm is driven by questions that concern the basic communication between the processors forming the distributed network. There are many models of distributed computation. It is generally assumed that processors are loosely coupled; that is, there are information channels between certain pairs of processors, but no other information is shared. In particular, there is no notion of a global state, as there is in a centralized algorithm. However, occasionally it does make sense to assume that all processors can work synchronously, through access to a global clock. Such networks are synchronous. If assumption of the existence of a global clock is ignored, the network is termed

as computation asynchronous. Distributed algorithms are mainly used for telecommunication services, real time process control, distributed information processing etc.

A Distributed algorithm is implemented for vertices in a fixed, finite and undirected network (V, E) and which consists of at least two vertices [3]. The mobile (cellular) network is viewed as a cellular graph. Two adjacent vertices can communicate a message. Since edges are undirected, each vertex can both send and receive messages through any of its adjacent edges. Except for delivering messages properly edge can damage, lose, duplicate and reorder message in transit.

When considering distributed algorithms, it is very difficult (if not impossible) to narrow the state of the art down to a small set of algorithms as the distributed algorithms may be designed for very different goals[4, 5]. These goals may concern, for example, the performance for particular topologies, the minimization of execution time (or communication rounds), the generation of the best colorings possible, or the performance for dynamically changing topologies [6, 7, and 8].

An algorithm is developed for constructing Independent set for a cellular graph. Further to this is the assignment of different colors to each independent set with the constraint that no vertex of cellular graph is having a same color. The procedure is explained with example in problem implementation part. There are three steps for the problem of assigning frequencies to each cell in mobile cellular network.

(1) Graph Creation: graph for cellular network is created.

(2) Computation of Independent Set: with the help of

Algorithm independent set is constructed.

(3) Reducing Independent Set to Graph Coloring: assignment

of different colors to each independent set.

2. RELATED WORK

Luby(1985) presented a randomized algorithm which require a linear number of processors. The main idea is to obtain for each vertex a local total order or a local election which breaks the local symmetry and then each vertex can decide locally whether to joins the MIS or not.

Linial(1992) showed a matching lower bound for time needed for 3 coloring of a n-cycle and proved that there is no constant time distributed algorithm.

Another remarkable work from the field of distributed algorithms for graph coloring is the Distributed Largest First(DLF) algorithm introduced in Kubale and Kuszner (2002). DLF is an improvement over an algorithm for randomly ordering the nodes and using this order to allow nodes to choose colors under the constraint that colors which have been already assigned to neighbors cannot be used. DLF differs from the original algorithm version in the order of the nodes. In DLF, the higher the degree of a node the earlier the node will be able to choose a color. The random order is only used to break ties between nodes which have the same degree. DLF achieved better results than the original algorithm on random graphs.

Madhav V. Marathe[2004] have performed an empirical study of a distributed an empirical study of a distributed edge coloring. They studied several classes of graphs viz. trees, bipartite graphs, cliques, hypercubes and random graphs. One of the aim of their study was to see how using a few extra colors affects the performance and reported that if 5%-15% extra colors are used then the performance is significantly better.

One of the most general works was presented by Finocchi (2005). The authors introduced three versions of a distributed algorithm and studied its behavior under various conditions. The authors considered both the problem of obtaining $O(\deg(v)+1)$ colorings in as few communication rounds as possible, as well as the problem of generating the best possible colorings without any limit on the number of communication rounds. Here, $\Delta(G)$ refers to the maximum degree of a graph. The authors provide extensive experimental results for both cases. Most of their experimentation is based on random graphs, which are not publicly available. However, they also presented results on a well-known set of publicly available instances from the DIMACS challenge (Center for Discrete Mathematics and Theoretical Computer Science 2006).

3. PROBLEM FORMULATION

For the algorithm in this section, several assumptions are made as illustrate here. For example,

- **Requirement of** that channels between processors be reliable, so no messages are lost.
- This also usually require that every machine has a unique ID (such as an IP address) and that the network does not change during the execution of the algorithm.
- Each machine is assumed to know its local neighborhood; that is, which machine it is concerned to directly by which channels.

Thus far, only the physical aspect of network design has been treated. However, the communication facility permits a task (program or other code sequence) to be spread over the processors of network. Of course there will be, in general, be a need for the constituent parts (called modules or processes) to exchange information resulting in a communication overhead being incurred.

The cost may well be more than offset via:

- Better turnaround resulting from modules being executed concurrently.
- Better processor utilization from load balancing
Modules being run on processors best suited to requirements.

3.1 Vertex Coloring

Graph coloring is composed of three types of vertex coloring, edge coloring and region coloring. The concept of vertex coloring of a graph can be used to model many scheduling problems, optimal assignment of channels to radio stations and optimal assignment of spectrum frequencies to mobile operations. However, the topic originated with the following map coloring conjecture. At most four colors are required to color any map of a country so that adjacent states receive different color.

A vertex coloring of a Graph G is a coloring of the vertices of G such that no two adjacent vertices of G receive the same color. The objective of the problem is to find a coloring that uses as few colors as possible. The minimum number of colors needed to color the vertices of a graph is known as Chromatic number. the register allocation problem in compiler design and the map coloring problem are instances of the vertex coloring problem. The problem of deciding if the chromatic number of a given graph is at most a given integer k is NP – complete. The problem is NP- complete even for fixed $k \geq 3$. For $k = 2$, the problem is to decide if the given graph is bipartite, and this can be solved in linear time using depth –first or breadth – first search. A greedy coloring of a graph yields a coloring that uses at most $\Delta+1$, where Δ is the maximal degree of the graph.

The edge coloring problem is similar to the vertex coloring problem. In this problem, the goal is to color the edges of a given graph using the fewest colors such that no two edges incident to a common vertex are assigned the same color. The problem finds application in assigning class room to course and in scheduling problems. The minimum number of colors needed to color a graph is known as its chromatic index. Since any two incident edges must receive distinct colors, the chromatic index of a graph with maximal degree Δ is at least Δ .

The region coloring is used to color the map. With each map, a graph is associated. This graph has a vertex for each region of the map and an edge connecting two vertices if and only if the regions they represent share a common boundary. A coloring of the map so that no two regions with a common boundary have the same color corresponds to a coloring of the vertices of the associated graph so that no two adjacent vertices are assigned the same color. A famous Four coloring theorem is used for coloring of regions.

Four Color Theorems: The regions of any simple planar map can be colored with only four colors, in such a way that any two adjacent regions have different colors.

This traditional application of graph colorings has been one of the major factors in the development of graph theory. Few applications are – Examination scheduling, Task Scheduling: Meeting Scheduling, Maintenance Scheduling:

3.2 Maximal Independent set

Symmetry breaking technique plays a major role in distributed network algorithms. Various computational task require the co-operation of many processors in the network, but prohibit certain processors pairs (or large group) from operating simultaneously. For example, in certain cases it may be prohibited for neighboring processors to act concurrently. This may require us to find a way to efficiently schedule processor groups under such prohibited rules.

The task that needs to be solved towards constructing such a schedule can be formalized as a vertex coloring or MIS (Maximal Independent Set) problem. Efficient MIS construction techniques can also help in decomposing the network into clusters of low diameter, which is often very useful in designing and implementing distributed divide and conquer algorithms.

The selection of MIS (maximal independent set) in a given graph is a very well-studied graph-algorithmic problem. A MIS is a basic building block in distributed computing; some other problems pretty much follow directly from the MIS problem [10]. Given a cellular graph $G = (V, E)$ an independent set is a subset of vertices $U \subseteq V$, such that no two vertices in U are adjacent. An independent set is maximal if no vertices can be added without violating independence. An independent set of maximal cardinality is called maximum independent set.

There is a relation between independent sets and vertex coloring, since each color class is an independent set, however, not necessarily a MIS still, starting with a coloring. One can easily derive a MIS algorithm: First choose all vertices of first color. Then, for additional color, add in parallel (without conflict) as many colors as possible.

The complete flow graph of the algorithm is shown in figure 1 given below.

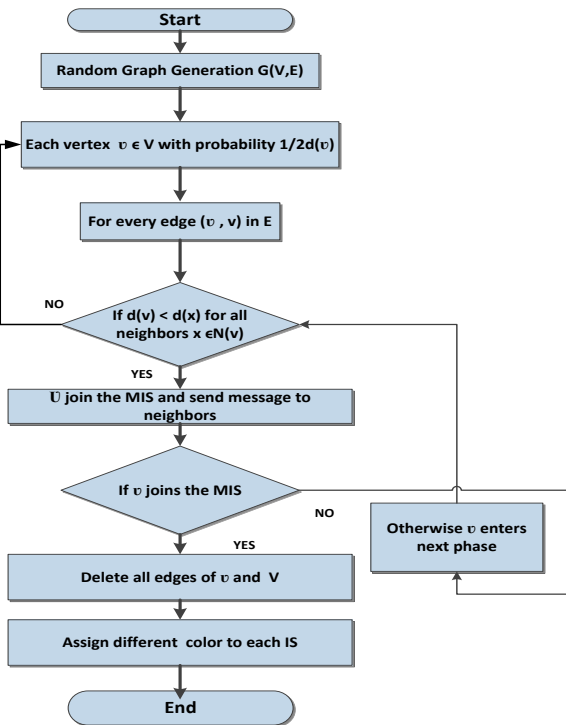


Fig 1: Flow graph for the Algorithm

4. PROBLEM IMPLEMENTATION

The problem is divided into three sub problems.

1. Graph Creation
2. Computation of Independent Set
3. Reducing Independent Set to Graph Coloring

4.1 Graph Creation

The Graph $G = (V, E)$ where V represents non-empty set of vertices and E represents non empty set of edges which shows relationship among vertices.

Test Case: Random Graph G with 8 Vertices

The input of the algorithm is a graph G , where $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $E = \{(1,2),(1,4),(1,6),(1,7),(2,1),(2,4),(2,7),(2,8),(3,4),(3,7),(3,8),(4,1),(4,2),(4,3),(5,1),(5,8),(6,1),(7,3),(7,2),(7,1),(8,5),(8,3),(8,2)\}$. The graphical view of the test case is shown in figure 2 as shown below.

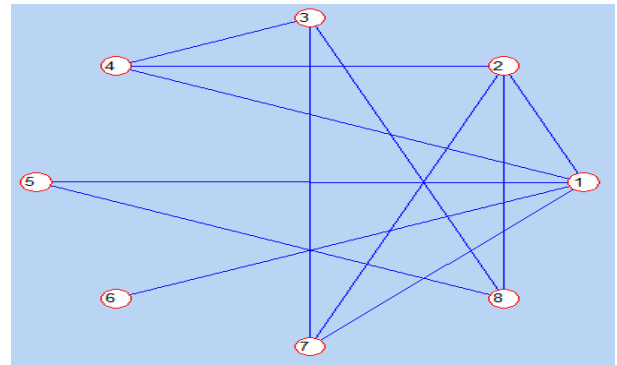


Fig 2: Random Graph for the Test Case

4.2 Computation of Independent Set

First the following information regarding the Graph considered is calculated.

1. Degree : $Deg(V)$
2. Neighbors of V : $Neigh(V)$
3. Probability : $Prob(1/2Deg(V))$

Table 1 shows the various results for the test graph in terms of the neighbors and Probability of each vertex of graph.

Table 1: Neighbors for each vertices of Test Case

Vertices	Degree	Neighbors	Probability
1.	5	{2,4,5,6,7}	0.100
2.	4	{1,4,7,8}	0.1250
3.	3	{4,7,8}	0.1667
4.	3	{1,2,3}	0.1667
5.	2	{1,8}	0.2500
6.	1	{1}	0.5000
7.	3	{1,2,3}	0.1667
8.	3	{2,3,5}	0.1667

In each iteration, every vertex V executes following steps.

1. Select Vertices with lower probability
2. Neighbors of v : $Neigh(v)$
3. If degree of $v < \text{degree of } Neigh(v)$, v join Maximal Independent Set(MIS)

4. If $v \in MIS$ delete all edges of v else repeat step 3

Table 2 shows the calculation of various independent sets in different iterations.

Table 2: Independent Set for the Test Case

Independent Set	1	2	3	4	5	6	7	8
Iteration : 1	√		√					
Iteration : 2		√			√	√		
Iteration : 3				√			√	√

4.3 Reducing Independent Set to Graph Coloring

For the graph under consideration, three sets of Independent Set have been calculated. Now, in next step each Independent Set is assigned a different color. As in each Independent set the neighbors of vertex v don't exist such that no vertices in a Independent set are adjacent. So assign a single color to an Independent Set. Figure 3 shows the final implementation result of the proposed algorithm in terms of the colors used for coloring of test case.

Here it has been shown that by finding the independent set for a set of vertices of a random graph the optimum number of colors can be found by the proposed approach which can be used for assigning minimum frequency assignment in mobile networks.

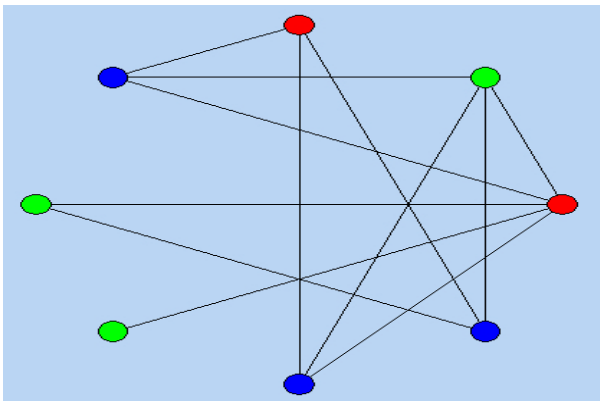


Fig 3: Graph Coloring for the Test Case

5. RESULTS

The experimental analysis of algorithm is performed and the test graphs are generated randomly by choosing the number of vertices. The algorithm is tested for the graph of vertices 5, 10, 15,20,30,50.

Table 3 shows the results of various number of nodes in terms of number of colors used and time consumed by algorithm for computation.

Table 3: Algorithm Performance for different Test Cases

Number of Vertices	Number of colors	Time Consumed
5	3	0.0981
10	3	0.1103
15	4	0.1145
20	3	0.1161
30	4	0.2340
50	5	0.3557

Table 4: Comparison of algorithm performance with optimized algorithm

Number of Vertices	Number of colors used by our Algorithm	Number of colors used by optimized channel assignment (Ref 11)
5	3	3
10	3	3
15	4	4
20	3	4
30	4	5
50	5	5

Table 4 shows the different cases for which the number of vertices are considered and the minimum number of colors are obtained by the proposed algorithm. It further compares the results of the proposed algorithm with one of the standard optimized algorithm as referred in the reference section.

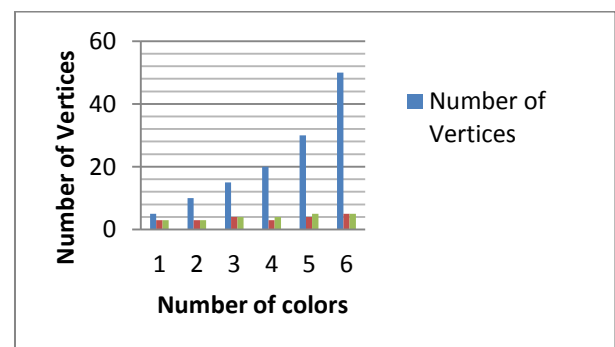


Fig 4: Performance Comparison chart with optimized channel assignment

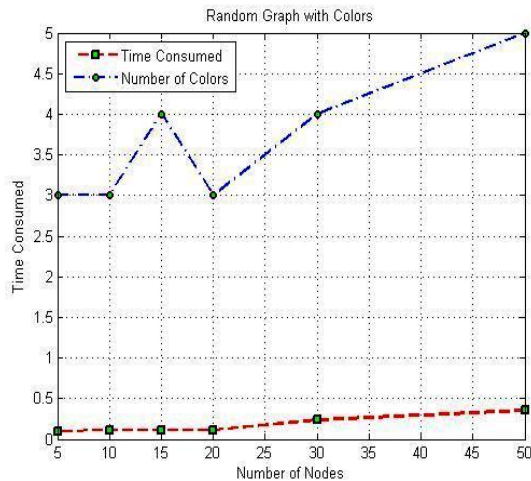


Fig 5: Performance for 6 different Test Cases

Figure 4 is drawn from the table 4 to indicate graphically the comparison with one of optimized channel assignment references.

Figure 5 shows the relation that the number of colors will not sharply increase as the number of nodes increase considerably which can be considered as a measure of optimum performance of algorithm.

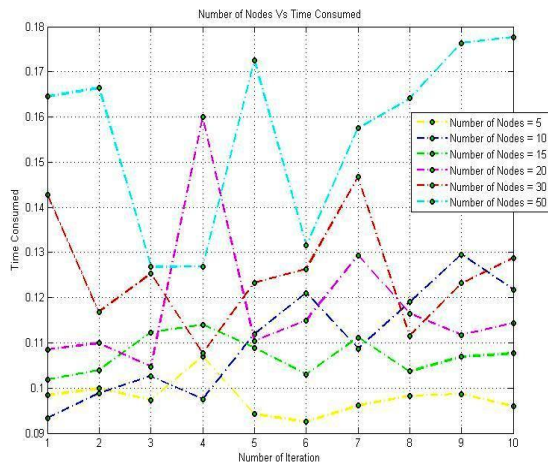


Fig 6: Timing Variation for 6 different Test Cases

The figure 6 clearly shows that the proposed algorithm is effective enough in computing the optimum number of colors used without having a sharp increase in time consumption.

The proposed algorithm is implemented in MATLAB version 7.11 on core2Duo processor 1.8 GHz CPU with 2 GB RAM.

6. CONCLUSION

The proposed algorithm is effective enough in handling larger graphs and the number of colors are always optimum even for greater number of vertices. Future scope includes consideration of improving the allocation performance in a situation where each mobile station is independent and the mobile station can only communicate to its neighbors. It is possible to improve the allocation of frequencies to mobile stations in mobile network so that mobile station can use frequencies more intelligently by applying graph coloring heuristics.

7. REFERENCES

- [1] Hale, W.K., "Frequency Assignment: Theory and Applications", *Proc. IEEE*, 68, 1980, pp. 1497-1514.
- [2] L. Narayanan and S. Shende, "Static Frequency Assignments in Cellular Networks", *Algorithmica*, 29, 2001, pp. 396-409.
- [3] Lynch, N. : Distributed algorithms , Morgan Kaufman,1996
- [4] Marek Kubale and Lukasz Kuszner, "A better practical algorithm for distributed graph coloring", *Proc. International Conference on Parallel Computing in Electrical Engineering*, 2002
- [5] Linial N., Locality in distributed graph algorithms, *SIAM J. on computing*, 20, 1992, pp.193-201.
- [6] Colin McDiarmid and Bruce Reed, " Channel Assignment and Weighted Coloring, *Networks*,36,2000, pp. 114-117
- [7] Irene Finocchi ,Alessandro Panconesi , Riccardo Silvestri ,"Experimental analysis of simple, distributed vertex coloring algorithms", *SODA '02 Proc. Thirteenth annual ACM-SIAM symposium on Discrete algorithms* ,2002
- [8] Madhav V. Marathe, Alessandro Pansonesi, Larry D.Risinger Jr, "An Experimental study of a Simple ,Distributed Edge Coloring Algorithm", *ACM Journal of Experimental Algorithmics* 9, 2004
- [9] V.Maan and D.sangwan " A Genetic Implementation of graph Coloring for cellular networks" *International Journal of electronics and engineering* 2(1) , 2010, pp. 183-187
- [10] Luby M., "A simple parallel algorithm for the maximal independent set problem", *Proc. Of ACM Symposium on theory of computing*, (1985)1-10.
- [11] Sunita Choudhary, G. N. Purohit, " Optimized Channel Assignment in Distributed Environment", *Proceedings of the World Congress on Engineering Vol I* ,2010