Journal of the Operations Research
Society of Japan
Vol. 39, No. 2, June 1996

APPLICATION OF THE GRAPH COLORING ALGORITHM TO THE FREQUENCY ASSIGNMENT PROBLEM

Taehoon Park Chae Y. Lee
Korea Advanced Institute of Science and Technology

(Received July 4, 1994; Revised April 12, 1995)

Abstract The frequency assignment problem is introduced and solved with efficient heuristics. The problem is to assign channels to transmitters using the smallest span of frequency band while satisfying the requested communication quality. A solution procedure which is based on Kernighan-Lin's two way uniform partitioning procedure is developed for the k-coloring problem. The k-coloring algorithm is modified to solve the frequency assignment problem. The performance of the proposed algorithm is tested with randomly generated graphs with different number of nodes, density types and graph types. The computational result shows that the proposed algorithm gives far better solution than a well-known heuristic procedure.

1 Introduction

Recently, the frequency band has become an important resource for communication service. There has been large increase in demand for using the frequency bands caused by the fast growth in mobile communication, satellite communication and mass communication service areas. To maximize its utilization, the limited band of available frequency is divided into a number of channels. A channel can be reused many times for different transmitters if the transmitters are far enough from one another so that the co-channel interference between them is low enough. If there are two close transmitters using the same channel simultaneously, they will suffer from severe co-channel interference and the quality of communication service will be unsatisfactory. Thus it is important to assign adequate channels to the transmitters. Since the available frequency band is limited, we are interested in using as small band of frequency as possible while satisfying all the frequency demand and the co-channel constraints.

The frequency assignment problem is described as follows [5]: Assume that there is a collection of possible channels for frequencies to be assigned to a set of transmitters. The level of interference between each pair of transmitters is known. Two transmitters which are close to each other must have two different channels to maintain a requested grade of communication quality. The minimum difference between two assigned frequencies, to be assigned to two close transmitters, depends on the interference level between them. The objective of the frequency assignment problem is to find an assignment using the smallest span of the frequency band while satisfying the communication quality constraints.

It is shown that the frequency assignment problem is equivalent to an extended version of graph coloring problem [5] and many algorithms have been developed for the problem. The graph coloring problem [3] is to color all the nodes of a graph with as small number of colors as possible such that any two adjacent nodes do not have the same color. The smallest number of colors needed to color a graph is called the chromatic number of the graph. Since it is a well-known combinatorial problem which is of NP-complete class [6], many heuristic algorithms are developed to solve the problem in a reasonable amount of time. The frequency assignment problem is very similar to the graph coloring problem but it has more condition on the available colors for the adjacent nodes.

A variation of the graph coloring problem is the k-coloring problem [4]. In the problem, if the chromatic number of a graph is less than or equal to k, then the graph is k-colorable and no penalties occur. However, if the graph is not k-colorable, then penalties occur for the edges which violate the coloring constraint. Thus, the objective of the k-coloring problem is to minimize the sum of weights on the edges between adjacent nodes with the same color. The solution procedure to handle the k-coloring problem is employed to solve the extended graph coloring problem.

The frequency assignment problem is defined in section 2. The k-coloring problem is defined and an algorithm for k-coloring problem is developed in section 3. In section 4, it is presented how the algorithm for k-coloring problem is employed for the graph coloring problem and the frequency assignment problem. An extended version of k-coloring problem is formulated to solve the frequency assignment problem and the algorithm for k-coloring problem is also modified. Computational results for some random graphs are given in section 5 and the conclusions are in section 6.

2 The Frequency Assignment Problem

There has been large growth in demand for using the frequency as a way of communication. In order to handle the demand for communication service at the required grade of service, a frequency channel should be assigned to each transmitter while satisfying the constraints on the channels. Due to the limited frequency spectrum, the available number of channels is finite. The frequency assignment problem is very important to manage the demand for the channels. Although there are many versions of the frequency assignment problem, one of them to be adopted in this paper is as follows:

Assume we have a collection $F = 1, 2, ..., k$ of the frequency channels to be assigned to a set of transmitters $X = 1, 2, ..., n$. Only one channel is assigned to a transmitter. The frequency channels are evenly spaced and two adjacent numbers in F are identified with two adjacent frequency channels. Let $f(i)$ represent the frequency channel assigned to the ith transmitter. An n by n compatibility matrix $Q = (q_{ij})$ with nonnegative integer components is given. Then the objective of the frequency assignment problem is to find an assignment $f : X \rightarrow F$ such that $z = \max_{i \in X} f(i)$ is minimized while satisfying $|f(i) - f(j)| \geq q_{ij}$ for all node pair (i, j). We call this constraint on the channel the minimum separation condition. If all the q_{ij}'s are 0 or 1, this problem reduces to the simple graph coloring problem. Hence the frequency assignment problem is equivalent to an extended version of graph coloring problem. For the frequency assignment problem, the value of q_{ij} depends on the distance between the two transmitters i and j and their transmitter power. For example, in [5], two-leveled constraints are given as follows:

(a) Two transmitters i, j which are close to each other must receive frequencies $f(i), f(j)$ which differ by at least 50 kHz.
(b) Two transmitters i, j which are very close to each other must receive frequencies $f(i), f(j)$ which differ by at least 100 kHz.

If the bandwidth of a channel is 50 kHz, q_{ij} is 1 in (a) and 2 in (b). The degree of "closeness" is defined as the interference level. Let P_i and P_j be the transmitter power from transmitters i and j, respectively, and d_{ij} be the distance between them. Then the interference level is defined as $I_{ij} = \max(P_i, P_j)/d_{ij}^2$. $q_{ij} = 1$ if $\theta_1 \leq I_{ij} < \theta_2$ and $q_{ij} = 2$ if $I_{ij} \geq \theta_2$ for some threshold values θ_1 and θ_2.

3 The k-coloring Problem and Algorithm

The k-coloring problem (kCP) is defined as coloring all the nodes in a graph with k colors such that the sum of weights on the edges between adjacent nodes with the same color is minimized. Let $f(i) = l$ or $i \in P_l$, if node i is colored with color l. A set P_l is a cluster of nodes which have the same color. The set of clusters $P = \{P_l\}_{l=1,2,\dots,k}$ is called a k-way partition of a graph $G(N, E)$, if $P_1 \cup \cdots \cup P_k = N$ and $P_l \cap P_m = \phi$ whenever $l \neq m$. Let w_{ij} be the weight of edge (i,j) and $|N| = n$. Then the k-coloring problem is formulated as:

$$(k\text{CP}) \text{ Minimize } \textstyle\sum_{f(i)=f(j),i<j} w_{ij}$$
$$\text{subject to } 1 \leq f(i) \leq k \text{ for all } i = 1,\dots,\, n$$

The k-coloring problem can be considered as an unrestricted version of the k-way graph partitioning problem [2], which is to divide the nodes of a graph into k subsets such that the total cost of the edge cut is minimized. A difference between kCP and the k-way graph partitioning problem is that the objective function of the kCP is to minimize the sum of internal weights (each weight of edge between two nodes in the same cluster) while that of graph partitioning is to minimize the sum of external weights (each weight of the edge between two nodes in different clusters).

The graph partitioning problem usually has a restriction on the size of each cluster. The restriction is called admissibility. In most cases, weights on the edges are given as nonnegative real numbers. Without considering admissibility, the optimal partition for the k-way graph partitioning problem with cost of zero is easily obtained by setting $P_1 = N$ and $P_2 = P_3 = \cdots = P_k = \phi$. However, the admissibility is not necessary for the kCP.

Suppose we have an exact algorithm for kCP. Then the cost of kCP by the algorithm is 0 if and only if the graph is k-colorable. Inspired by this idea, we present an algorithm for the kCP and employ it for the graph coloring problem. Here we propose an improving algorithm for the k-coloring problem by modifying the Kernighan-Lin's two way uniform partitioning procedure [2]. The algorithm developed in [2] for the two way partitioning problem considers two clusters at each step and decides nodes to be exchanged. Since there is no admissibility condition in the kCP, the algorithm presented in this section considers not only the selection of nodes to be exchanged but also the selection of a single node and a cluster for the node to move in.

Let $W(i,l)$ be the cost occurred by the weights of all the edges between node i and all the nodes of P_l. Since it is the sum of all the edges joining node i and nodes in P_l, $W(i,l) = \sum_{j \in P_l} w_{ij}$. Let $D(i,l) = W(i,l) - W(i,f(i))$ such that $D(i,l)$ represents the cost increment when node i is moved to P_l. Then $-D(i,l)$ represents the cost reduction when node i is moved to P_l as shown in the following proposition.

Proposition 1 If node i is moved to P_l, the cost reduction (old cost $-$ new cost) becomes $-D(i,l)$.

Proof Let z be the cost occurred by the weights of all the edges except those between i and all the nodes of $P_{f(i)}$. Then the old cost $= z + W(i,f(i))$ and the new cost $= z + W(i,l)$. Thus, the cost reduction becomes $W(i,f(i)) - W(i,l) = -D(i,l)$.

Proposition 2 If nodes a and b are exchanged, the cost reduction becomes $-D(a, f(b)) - D(b, f(a)) + 2w_{ab}$

Proof Let z be the sum of weights of all edges except the following two classes of edges:

(i) edges that are connected to node a from nodes colored with $f(b)$

(ii) edges that are connected to node b from nodes colored with $f(a)$

Then

old cost $= z + W(a, f(b)) + W(b, f(a)) - w_{ab}$

new cost $= z + W(a, f(a)) + W(b, f(b)) + w_{ab}$

cost reduction $= D(a, f(b)) + D(b, f(a)) - 2w_{ab}$.

The proposed algorithm for k-coloring problem is as follows:

The initial color of each node is set to 1. In other words, the initial partition is composed of a cluster which consists of all the nodes of G and $k-1$ empty clusters. After initial coloring of all the nodes, the partition is improved by examining the cost reduction obtainable from each node. At each iteration the algorithm selects the best (node, color) pair with the maximal cost reduction. Thus by Proposition 2 the algorithm selects a pair (a, l^*) such that $D(a, l^*) = \min_{i \in C, 1 \leq l \leq k} D(i, l)$, where C is the candidate list of nodes.

When it is impossible to reduce the cost by moving a node to another cluster, the algorithm exchanges two nodes a and b so that the cost reduction by their exchange is maximized. It is shown by experiments that this operation helps finding better rearrangement of nodes. The improving procedure (Move and Exchange) continues until the cost reduction is nonpositive for M consecutive iterations.

During the search process to prevent a node from being selected again, a tabu list [1] is employed. The list contains a set of nodes which have been already moved or exchanged in the last L iterations. It thus provides a list of forbidden nodes to the candidate list C.

The value of cost increment $D(i, l)$ must be updated whenever a node is moved to a cluster or two nodes are exchanged according to the following propositions.

Proposition 3 After node a is moved to P_{l^*}, the D-values are updated as follows:

(3.1) $D'(i, f(a)) = D(i, f(a)) - w_{ia}$ if $f(i) \neq f(a)$ and $f(i) \neq l^*$

(3.2) $D'(i, f(a)) = D(i, f(a)) - 2w_{ia}$ if $f(i) = l^*$

(3.3) $D'(i, l^*) = D(i, l^*) + w_{ia}$ if $f(i) \neq f(a)$ and $f(i) \neq l^*$

(3.4) $D'(i, l^*) = D(i, l^*) + 2w_{ia}$ if $f(i) = f(a)$

(3.5) $D'(i, l) = D(i, l) + w_{ia}$ if $f(i) = f(a)$ and $l \neq f(a), l \neq l^*$

(3.6) $D'(i, l) = D(i, l) - w_{ia}$ if $f(i) = l^*$ and $l \neq f(a), l \neq l^*$

(3.7) $D'(i, l) = D(i, l)$ otherwise

Proof For a node i which is neither with $f(a)$ nor l^*, $D(i, f(a))$ values are decreased by w_{ia} because a is now with l^* and $W(i, f(a))$ value is decreased by the same amount. Thus (3.1) holds. Since the amount is the same as the increase of $W(i, l^*)$, (3.3) also holds.

For a node i with $f(a)$, $W(i, f(a))$ is decreased by w_{ia} and $W(i, l^*)$ is increased by w_{ia}, so (3.4) and (3.5) holds. For a node i with l^*, similar argument shows that (3.2) and (3.6) holds. Otherwise, there is no change in $W(i, l)$ and $W(i, f(i))$. Thus there is no change of value $D(i, l)$.

Proposition 4 After nodes a and b are exchanged, the values are updated as follows :

(3.8) $D'(i, f(a)) = D(i, f(a)) + w_{ib} - w_{ia}$ if $f(i) \neq f(a)$ and $f(i) \neq f(b)$

(3.9) $D'(i, f(a)) = D(i, f(a)) + 2w_{ib} - 2w_{ia}$ if $f(i) = f(b)$

(3.10) $D'(i, f(b)) = D(i, f(b)) + w_{ia} - w_{ib}$ if $f(i) \neq f(a)$ and $f(i) \neq f(b)$

(3.11) $D'(i, f(b)) = D(i, f(b)) + 2w_{ia} - 2w_{ib}$ if $f(i) = f(a)$
(3.12) $D'(i, l) = D(i, l) + w_{ia} - w_{ib}$ if $f(i) = f(a)$ and $l \neq f(a), l \neq f(b)$
(3.13) $D'(i, l) = D(i, l) + w_{ib} - w_{ia}$ if $f(i) = f(b)$ and $l \neq f(a), l \neq f(b)$
(3.14) $D'(i, l) = D(i, l)$ otherwise

Proof For a node i which is neither with $f(a)$ nor with $f(b)$, $D(i, f(a))$ values are increased by $w_{ib} - w_{ia}$, because a is replaced by b and $W(i, f(a))$ is increased by the same amount. Thus (3.8) holds. Similarly, (3.10) also holds. If $f(i) = f(a)$ then $W(i, f(a))$ and $W(i, f(b))$ are increased by $w_{ib} - w_{ia}$ and $w_{ia} - w_{ib}$, respectively. Thus (3.11) and (3.12) holds. If $f(i) = f(b)$, similar argument shows that (3.9) and (3.13) holds. Otherwise, there is no change in $W(i, l)$ and $W(i, f(i))$.

Based on the discussion of this section we propose an algorithm for the k-coloring problem.

Algorithm for kCP

Step 0: Set $f(i) = 1$ for each i;
Step 1: $C = N$;
    Calculate $W(i, l), D(i, l)$ values for each i, l;
Step 2: Move;
    If cost reduction ≤ 0 Exchange;
    Update tabulist
Step 3: If cost reduction ≤ 0
        count = count + 1;
    Else
        count = 0;
    End If
Step 4: If count = M
        Stop;
    Else
        Go to Step 2;
    End If

Algorithm Move

Step 1: Find a in C and l^* between 1 to k such that
    $D(a, l^*) = \min D(i, l)$;
Step 2: Mode = 1;
Step 3: Update $D(i, l)$ values for each i in C and each l by Proposition 3;
    Let cost reduction = $-D(a, l^*)$;
    If cost reduction > 0 then $f(a) = l^*$;

Algorithm Exchange

Step 1: Find a and b in C such that
    $g(a, b) = \min g(i, j)$
    where $g(i, j) = D(i, f(j)) + D(j, f(i)) - 2w_{ij}$;
Step 2: Mode = 2;
Step 3: Update $D(i, l)$ values for each i in C and each l by Proposition 4;
    Let cost reduction = $-g(a, b)$;
    Exchange the values of $f(a)$ and $f(b)$;

Algorithm Update tabulist

Step 1: For all i in tabulist
   $tabutime(i) = tabutime(i) + 1;$
   If $tabutime(i) > L$ $tabutime(i) = 0;$

Step 2: If Mode = 1
   $tabutime(a) = 1;$
  Else If Mode = 2
   $tabutime(a) = 1;$
   $tabutime(b) = 1;$
  End If

Step 3: $C = \{i | tabutime(i) = 0\}$

4 Application to the Graph Coloring and Frequency Assignment Problems

The algorithm for the k-coloring problem presented in Section 3 is first applied to solve the graph coloring problem. Note that the objective of the graph coloring problem (GCP) is to find the smallest number k such that the graph is k-colorable. In the algorithm for kCP, if each weight on the existing edge is set to one, the cost of kCP is exactly the number of violated constraints on the graph coloring problem. If the cost of kCP is zero, then the graph is k-colorable. By reducing k one by one, we obtain the smallest positive integer k such that the cost of kCP is zero. If the algorithm is exact, it is obvious that the number k is the least possible number such that the graph is k-colorable. However, since the algorithm for kCP is not exact, the procedure provides a good upper bound.

Therefore, the algorithm for the graph coloring problem is stated as follows: The initial number k is set to a number which is large enough to make the cost of the kCP become zero. At each iteration the number of colors used in the graph is computed by the algorithm for kCP. The procedure then continues to solve the graph coloring problem with one less number of colors than used in the previous iteration. The algorithm is repeated in this way as far as the cost of k-coloring problem is zero. Otherwise, the algorithm fails to find a k-coloring and it stops. An upper bound on the minimum number of colors required for the graph is obtained.

To solve the frequency assignment problem an additional condition needs to be considered. In the graph coloring problem, any color is available on a node if the color is not assigned to its adjacent nodes. However, in the frequency assignment problem, the channel to be assigned to a node must keep the minimum separation condition at adjacent nodes.

To solve the frequency assignment problem, we define a corresponding kCP as follows:

$$\text{(4.1)} \qquad \begin{array}{l} \text{Minimize } \sum_{i<j} c_{ij} \\ \text{subject to } 1 \leq f(i) \leq k \text{ for } i = 1, ..., n \\ \text{where } c_{ij} = \max(q_{ij} - |f(i) - f(j)|, 0) \end{array}$$

It is clear that the cost of (4.1) is zero, if the assignment violates none of the minimum separation constraints. However, in order to apply the algorithm in Section 3, W- and D-values must be redefined as follows:

$W(i, l) = \sum_{f(j)=l} c_{ij}$
$D(i, l) = W(i, l) - W(i, f(i))$

By these new definitions, D-values are updated according to the following proposition. We omit the proof.

Proposition 5 After node a is moved to P_{l^*}, the D-values are updated as follows:
(4.2) $D'(i,l) = D(i,l) + \max(q_{ia} - |f(i) - f(a)|, 0) - \max(q_{ia} - |l - f(a)|, 0) + \max(q_{ia} - |l - l^*|, 0) - \max(q_{ia} - |f(i) - l^*|, 0)$

When the nodes a and b are exchanged, D-values can be updated by applying Proposition 5 twice. First, move a to $P_{f(b)}$ and update D-values, then move b to $P_{f(a)}$ and update D-values again.

5 Computational Results

The proposed algorithm for frequency assignment problems are tested with 8 randomly generated graphs. The test graphs are generated in three aspects. The number of nodes (200 and 400), the density type (sparse and dense), and the graph type (Euclidean or random). Euclidean graphs are generated in this way: Let P_i and P_j be the transmitter power from transmitters i and j, respectively, and d_{ij} be the distance between them. Then the interference level is defined as $I_{ij} = \max(P_i, P_j)/d_{ij}^2$. The nodes are uniformly distributed in 100 by 100 grid. The distance between each pair of nodes is given as the Euclidean distance. The power of each transmitter is uniformly distributed in the integer interval 1, 5. The correspondence between the interference level and the value of q_{ij} is shown in Table 1. Random graphs are generated with the same portion of q_{ij}'s as in Table 1, but the edges are generated randomly in the grid.

The proposed algorithm is compared to de Werra's greedy algorithm which is an extension of "Degree of Saturation Algorithm (Dsatur)" [5]. Note that the greedy algorithm selects a node such that the number of forbidden colors of the node is the highest. If ties happen in the selection, then the algorithm chooses the node with the highest degree. As shown in Table 2, the proposed algorithm for the frequency assignment problem gives far better solution than the greedy algorithm. The required span by the proposed algorithm is less than that by the greedy algorithm in every case.

6 Conclusions

The frequency assignment problem is considered which distributes channels to transmitters such that the span of frequency band is minimized. Since each channel assigned to a transmitter has to satisfy the minimum separation condition, the problem becomes an extended version of graph coloring problem. An efficient heuristic algorithm is developed for the k-coloring problem. The algorithm which is based on the Kernighan-Lin's two way uniform partitioning procedure controls the k clusters such that the cost reduction is maximized when a selected node is moved into a selected cluster or when two nodes are exchanged. Tabu list prevents cycles in the selection of nodes to move or to exchange. The algorithm for the k-coloring problem is modified to handle the minimum separation

Table 1. Conditions for Minimum Separation in Euclidean Graphs

q_{ij}	Interference Level for Sparse Graphs	Portion in Generated Graphs	Interference Level for Dense Graphs	Portion in Generated Graphs
0	$I_{ij} < 0.003$	0.717	$I_{ij} < 0.001$	0.373
1	$0.003 \le I_{ij} < 0.012$	0.200	$0.001 \le I_{ij} < 0.040$	0.407
2	$0.012 \le I_{ij} < 0.048$	0.059	$0.004 \le I_{ij} < 0.016$	0.157
3	$0.048 \le I_{ij} < 0.144$	0.015	$0.016 \le I_{ij} < 0.048$	0.040
4	$0.144 \le I_{ij}$	0.009	$0.048 \le I_{ij}$	0.023

Table 2. Computational Results on the Frequency Assignment Problems

Graph Type	Number of Nodes	Density Type	Required Span by Greedy Algorithm	Required Span by Proposed Algorithm
Euclidean	200	Sparse	46	43
Euclidean	200	Dense	90	87
Euclidean	400	Sparse	80	76
Euclidean	400	Dense	181	163
Random	200	Sparse	31	28
Random	200	Dense	51	49
Random	400	Sparse	65	63
Random	400	Dense	118	114

condition between a pair of nodes in the frequency assignment problem.

To solve the frequency assignment problem the interference level is defined for each pair of transmitters using the strength of power and distance. By employing the proposed algorithm the frequency assignment problem is solved with less span of frequency than by the greedy algorithm.

References

[1] Glover, F., (1990) "Tabu search: a tutorial", Interfaces 20, pp.74 – 94.

[2] Kernighan, B. and Lin, S., (1970),"An efficient heuristic procedure for partitioning graphs," The Bell System Technical Journal 49, pp. 291 – 307.

[3] Leighton, F. T., (1979), "A graph coloring algorithm for large scheduling problems," Journal of Research of the National Bureau of Standards 84, pp. 489 – 506.

[4] Taehoon P. and Lee, C. Y., (1994) "On the k-coloring problem", Journal of Korean OR/MS Society, 19, pp. 219 – 233.

[5] de Werra, D. and Gay, Y., (1994), "Chromatic scheduling and frequency assignment" Discrete Applied Mathematics 49, pp. 165 – 174.

[6] Williams, M. R., (1970) Combinatorial structures and their applications, Gorden and Breach, New York.

Taehoon Park and Chae Y. Lee
Korea Advanced Institute of Science and Technology
373-1 Kusung Dong, Taejon, Korea