中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-021

Generalized Edge Coloring for Channel Assignment in Wireless Networks

Chun-Chen Hsu, Pangfeng Liu, Da-Wei Wang, Jan-Jan Wu

December 2005 || Technical Report No. TR-IIS-05-021

http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/tr05.html

Generalized Edge Coloring for Channel Assignment in Wireless Networks

Chun-Chen Hsu
Institute of Information Science
Academia Sinica
Taipei, Taiwan

Pangfeng Liu
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan

Da-Wei Wang Jan-Jan Wu
Institute of Information Science
Academia Sinica
Taipei, Taiwan

Abstract

This paper introduces a new graph theory problem called *generalized edge coloring* (g.e.c). g.e.c is similar to traditional edge coloring, with the difference that a vertex can be adjacent to up to k edges that share the same color. g.e.c. can be used to formulate the channel assignment problem in multi-channel multi-interface wireless networks. We provide theoretical analysis for this problem. Our theoretical findings can be useful for system developers of wireless networks.

We show that when $k = 3$, there is no optimal solution. When $k = 2$ and the maximum degree of the graph is no more than 4, or is a power of 2, we derive optimal algorithms to find a g.e.c. Furthermore, if given one extra color, we can find a g.e.c. that uses minimum number of edge colors for each vertex.

1 Introduction

Many modern wireless LAN standards, such as IEEE 802.11b/802.11g [2] and IEEE 802.11a [1], provide multiple non-overlapped frequency channels that can be used simultaneously within a neighborhood. Ability to utilize multiple channels substantially increases the effective bandwidth available to wireless network nodes [3, 10, 12, 11]. One way to utilize multiple channels is to equip each node with multiple network interface cards (NICs) [7]. For direct communication, two nodes need to be within communication range of each other, and need to have a common channel assigned to their interfaces. Node pairs using different channels can communicate simultaneously without interference. Furthermore, since the number of interface cards per node is limited, each node typically uses one interface to communicate with multiple of its neighbors. The channel assignment problem is to bind each neighbor to a network interface and also bind each network interface to a radio channel with the goal to minimize interference [7, 12]

Specifically, we consider channel assignment that satisfies the following constraints: (1) The total number of radio channels assigned to a wireless node is bounded by the number of network interface cards it has, (2) the capacity of a radio channel within a communication range is bounded by a constant number k; the consequence is that an interface on a node can communicate with up to k neighboring nodes, and (3) two nodes that need to communicate with each other directly should share at least one common channel. Clearly, the channel assignment for each network interface affects the number of interface cards a node must have in order to communicate with all of its neighbors. It also affects the total number of channels that are actually used.

Graph coloring seems to be a natural formulation for this problem. However, standard vertex coloring [8, 4] (and more recently, vertex-multi-coloring) [5] cannot capture the third constraint that communicating vertices need to be assigned a common color. Standard edge coloring [8] fails to capture the second constraint that no more than k colors can be assigned to the adjacent edges of a vertex.

In this paper, we introduce a new graph theory problem called *generalized edge coloring* (g.e.c.). Generalized edge coloring is similar to traditional edge coloring, with the difference that a vertex can be adjacent to up to k edges that share the same color. We show that the channel assignment problem described above can be formulated as a generalized edge coloring problem as follows. By picking a color for an edge, we assign the channel number on the two interfaces on two neighboring nodes. By restricting the number of adjacent edges that have the same color, we limit the number of neighbors that can communicate with the same interface. In this paper, we provide theoretical analysis for the generalized edge coloring problem. Our theoretical findings are interesting and can be useful for system developers of multi-channel multi-interface wireless networks.

There are two criteria to evaluate the quality of a generalized edge coloring. The first is the total number of colors used (which is equivalent to the total number of channels used in the wireless network), and the second is the number of edge colors adjacent to a vertex (which is equivalent to the number of network interface cards on each node). An optimal g.e.c. should use only $\lceil \frac{D}{k} \rceil$ channels where D is the maximum degree of the wireless mesh network, and for each node with d neighbors, it should use only $\lceil \frac{d}{k} \rceil$ network interfaces. We show that when k is 3, it is impossible to find optimal g.e.c. for some graphs. However, when k is 2, we have an optimal algorithm for several cases: (1) when D is no more than 4,(2) when D is a power of 2, or (3) the graph is bipartite. In addition, if we are given one extra channel, we can derive a g.e.c. that achieves optimal number of interfaces for each node. This is very similar to the case of traditional edge coloring where k is 1, and finding an edge coloring with D colors is NP-complete, but it is always possible to color any graph with $D+1$ colors.

The rest of the paper is organized as follows. Section 2 formally defines the generalized edge coloring and the quality measurement criteria. Section 3 describes our results on the generalized edge coloring problem. Finally Section 4 concludes with some interesting open problems in this research topic.

2 Problem

This section defines our terminology about *generalized edge coloring*. Given a graph $G = (V, E)$ we color every edge with mapping function f from E to a color set C. In particular, we require that every node in V is adjacent to at most k edges of the same color. As a result the traditional edge coloring is a special case when k is 1.

We can derive some trivial lower bounds on the number of colors required. For example, let D be the maximum degree of G, then we need at least $\lceil \frac{D}{k} \rceil$ colors to color G. Also if the number of neighbors of a node v is d_v, the number of colors required to color the edges adjacent to v is $\lceil \frac{d_v}{k} \rceil$. We define the *global discrepancy* of a coloring function f as difference between the total number of colors f actually uses and the lower bound $\lceil \frac{D}{k} \rceil$, i.e. $|C| - \lceil \frac{D}{k} \rceil$. Similarly we define the *local discrepancy* of a node v to be the difference between the actual number of colors adjacent to a node v and the lower bound $\lceil \frac{d_v}{k} \rceil$, i.e. $n(v) - \lceil \frac{d_v}{k} \rceil$, where $n(v)$ is the number of colors adjacent to v under f. The *maximum local discrepancy* is the maximum among all the local discrepancy, i.e. $\max_v(n(v) - \lceil \frac{d_v}{k} \rceil)$.

We now use the global and the maximum local discrepancy to evaluate the quality of a coloring function. A coloring function is a (k, g, l) generalized edge coloring if every node in V is adjacent to at most k edges of the same color, the global discrepancy is bounded by g, and the maximum local discrepancy is bounded by l. For example, we know that the problem of determining whether a graph has a $(1, 0, 0)$ g.e.c. is NP-complete, and the Vizing's theorem says that it is always possible to find a $(1, 1, 0)$ g.e.c. for any graph. A generalized edge coloring is *optimal* if and only if it is a $(k, 0, 0)$ coloring.

3 Results

We first give an impossibility result on the case of $k \geq 3$, and show that there are cases that we cannot find $(k, 0, 0)$ g.e.c. for them. The construction is as follows. First we construct a ring of $2k$ nodes, and each node is connected to its two neighbors with two edges. This leaves $k - 2$ edges for each nodes along the ring. Now we place $k - 2$ nodes in the middle of the ring, and connect each one of them to every node along the ring. Now each node in the middle has degree $2k$. Suppose we can find a $(k, 0, 0)$ g.e.c. for this graph, the edges along the ring must be colored with the same color, since each node along the ring is of degree k, and from the 0 local discrepancy requirement, it can have at most one color. This forces all the edges going to the nodes in the middle to be colored with the same color, which violates the requirement that a node can be adjacent to at most k edges of the same color. Figure 1 illustrates the constructed graph when k is 3.

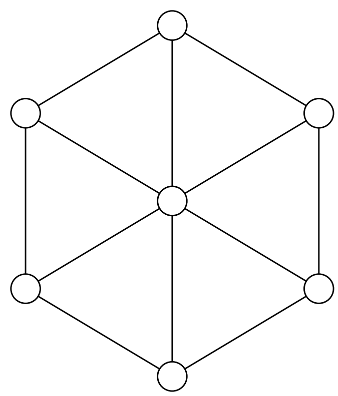

Figure 1: A graph without $(3, 0, 0)$ g.e.c.

From now on we focus on the case where k is 2. We will show that when the maximum degree is bounded by 4, we can always find $(2, 0, 0)$ g.e.c. using Euler cycle.

3.1 Euler Cycle

It is well known that a graph has a Euler cycle if and only if every node is of even degree. We will construct a $(2, 0, 0)$ g.e.c. based on the Euler cycle. The first step of our algorithm is to pair up all the nodes with degree 1 or 3, so that every node is now of degree 2 or 4. Since the number of odd-degreed nodes in a graph is always an even number, the step will not leave any odd-degreed nodes. We use G' to denote the graph after transformation.

The second step is to remove some degree 2 node to simplify the later coloring process. Consider the nodes with degree 2 – these nodes are all on paths that connect degree 4 nodes. If the path connect two different degree 4 nodes, as in Figure 2 (a), we remove all of them and place a single edge. If the path goes back to the same degree 4 node and forms a self loop, as in Figure 2 (b), we remove all but two nodes from the path. We denote the transformed graph as G^*.

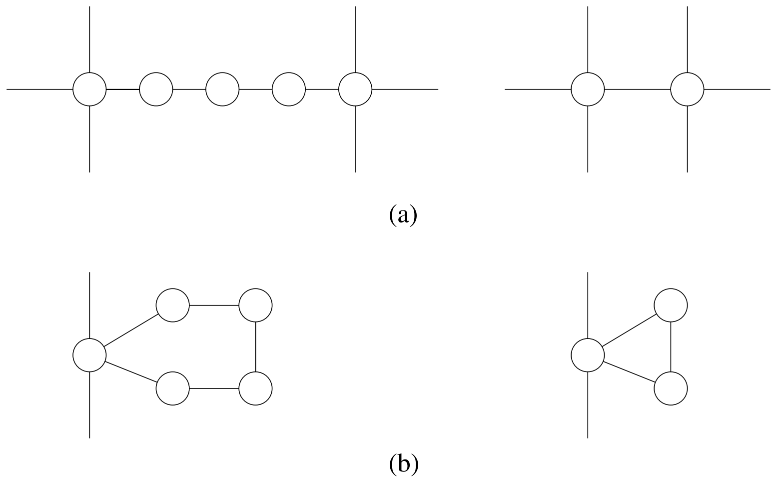

Figure 2: Two cases to remove some degree 2 nodes.

Now we construct a Euler cycle for the transformed graph. Since every node is of degree 2 or 4, the construction is possible. We then index each edge with a sequence number according to the order it appears in the cycle. For all edges that have even indices we color them with 0, and the other edges are colored with 1.

Lemma 1 *The Euler cycle constructed from G^* has even length, and every node has the same number of adjacent edges that are colored with 0 and 1.*

Proof. The length of the Euler cycle is equal to the number of edges in G^*, which is equal to the sum of all degrees of nodes in G^* divided by 2. Since there are only degree 4 nodes and pairs of degree 2 nodes in G^*, the Euler cycle has even length. In addition, the color are given in alternative manner, each degree 4 node has two edges of 0 and two edges of 1, and each degree 2 node has one 0 edge and one 1 edge. ■

Now we need to derive the actual coloring function for G'. If a set of nodes is replaced by a single edge since the path they form connects two different degree 4 nodes, the entire path is colored with the same color from the G^* coloring. This is feasible since k is 2. On the other hand, if a path form a self loop and is replaced by a path of length 3 (with two degree 2 nodes), the first and the third edge is colored the same color due to the alternating coloring. As a result we can color all the nodes in that path with the same color. Note that this special treatment is necessary, otherwise the alternating coloring process will be complicated.

Finally we need to remove the added edges from G'. We only added edges to those nodes in G that have degree 1 or 3. These nodes in G' now has the same number of edges colored by 0 or 1, so no matter which edge we remove, the local discrepancy will not increase. Formally we have the following theorem.

Theorem 2 *There exists a $(2, 0, 0)$ generalized edge coloring for every graph with maximum degree bounded by 4.*

The pseudo code of the alternating coloring process is as follows.

3.2 One Extra Color

We now describe an algorithm that finds $(2, 1, 0)$ g.e.c. for every graph. It is well known that although to determine if a graph allows a $(1, 0, 0)$ g.e.c. is NP-complete, to find a $(1, 1, 0)$

procedure AlternatingColoring

1. Pair up odd-degree nodes and add edges.
2. Remove some degree 2-nodes according to Figure 2.
3. Find a Euler cycle.
4. Color the edges alternatively with 0 or 1.
5. Color the edges along the path in Figure 2. with the same color.
6. Remove edges added in step 1.

Figure 3: The pseudo code of finding a $(2, 0, 0)$ g.e.c. for graph with maximum degree 4.

g.e.c. in polynomial time is always possible from Vizing's theorem [13]. Therefore the first step of our algorithm is find a $(1, 1, 0)$ g.e.c.

Now we will reduce the number of color by half. Let D be the maximum degree of the graph G. From Vizing's theorem we know that we need at most $D+1$ colors to come up with a (1.1.0) g.e.c. By grouping two colors into a new color, we will have at most $\lceil \frac{D+1}{2} \rceil$ new colors. Since the original coloring is a $(1, 1, 0)$ g.e.c. The new coloring is a $(2, 1, L)$ where L is about $\frac{D}{4}$. The reason is that we might use one more color than the $\lceil \frac{D}{2} \rceil$ lower bound, and a node with $\frac{D}{2}$ edges may still have $\frac{D}{2}$ new colors adjacent to it after we combine colors, which is about $\frac{D}{4}$ higher than the $\lceil \frac{D}{4} \rceil$ lower bound, hence the maximum local discrepancy can go up to about $\frac{D}{4}$.

Now we want to reduce the maximum local discrepancy to 0. The idea is to find a node v and two colors c and d so that v is adjacent to exactly one edge (denoted by (v, w)) colored by c, and one edge (denoted by (v, u)) colored by d. If we can change the color of (v, w) from c to d without increasing the local discrepancy of w, we can reduce the local discrepancy of v. For ease of notation we use $N(v, c)$ to denote the number of edges adjacent to v that are colored c. If we can do this for every node v that has $N(v, c) = N(v, d) = 1$ for two colors c and d, we can reduce the maximum local discrepancy to 0 by repeatedly changing the c to d for every node v that has $N(v, c) = N(v, d) = 1$.

The key operation for changing color is to find a $c - d$ path. The idea about $c - d$ path is inspired by [9]. Without lose of generality we assume that we want to change color c to d. A $c - d$ path is defined as follows:

- A $c-d$ path starts from v, goes through the unique edge (v, w) that is colored c, travels along only edges colored with c or d, and ends at a node other than v.
- If we exchange the colors of the edges between c and d along the $c - d$ path , we will not increasing the local discrepancy of any node along the path.

Suppose we can alway find a $c - d$ path from v, we can reduce the maximum local discrepancy to 0.

The $c - d$ path construction is as follows: We always check for whether the current path under consideration is already a $c - d$ path. If so, we *stop* and declare that a path is found. If not we *extend* the current path and hope that we can stop at the next edge. Initially the path under consideration is from v to w, i.e. the unique edge colored c.

There are several case to consider while determining whether we should stop or extend. Without lose of generality we assume the we just extend to a node x through an edge colored c. Similar argument can be made for an edged colored d, since we are extending a path that could have both color c and d. If $N(x, c) = 1$ and $N(x, d) = 0$, we stop at x since x is adjacent to one c, and changing that only c to d will not increase the local discrepancy of w.

In the second case we have $N(x, c) = 2$ and $N(x, d) = 0$. We cannot stop at x in this case since that will increase the number of colors adjacent to x by one. As a result we extend the $c - d$ path through the other edge colored by c. Note that changing both c will not increase the local discrepancy of x, and we only extend the path by one more node.

In the third case we have $N(x, d) = 1$. In this case we can stop at w since both (x, c) and $N(x, d)$ are greater than 0. Changing the incoming c edge will not increase the number of colors adjacent to x, and since there is only one d edge before the change, adding another one will not violate the $k = 2$ constraint either.

In the final case we have $N(x, d) = 2$. In this case we cannot stop at x, otherwise the number of d edges adjacent to x will be 3, violating the $k = 2$ constraint. As a result we pick an edge colored by d and extent the path.

Since each edge can only be used once in the process, eventually the process must stop and we find a $c - d$ path. The only complication is that the end node might be v, therefore we will not be able to reduce the local discrepancy of v. The following lemma says that we can always find a $c - d$ path that stops at a node other than v.

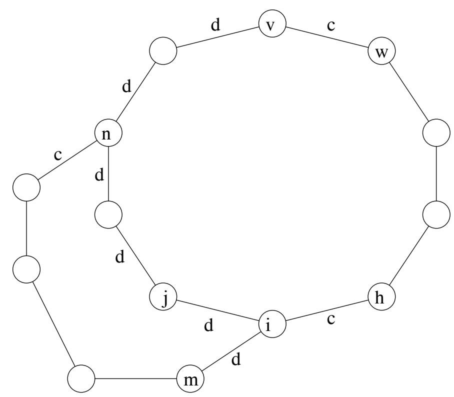

Figure 4: There exists a $c - d$ path that starts from v but does not end at v.

Lemma 3 *There exists a $c - d$ path that stops at a node other than v.*

Proof. Assume that we construct a $c - d$ path and eventually go back to v by a cycle C. Since the path starts with a c edge and ends with a d edge, let h denote the last node that extends a c edge, and this edge leads to node i. Since during the construction we extend through node i, therefore $N(i, d) = 2$ and there exists another edge (i, m) that is colored d. See Figure 4 for an illustration.

If we pick (i, m) to extend (instead of (i, j)) the $c - d$ path, it will be impossible to get back to v. The reason is that both $N(v, c)$ and $N(v, d)$ are 1, so the only way back to v is through the d edge. Please refer to Figure 4 for an illustration. If the $c - d$ path does reach v, we trace back from v to the node where it branches off the cycle C at node n. By definition we know that we will see only d edges before n, and we branch off C via a c edge, due to the $k = 2$ constraint. Recall that during the construction of C, when we enter n we will leave through another d edge only if there is no c edge adjacent to n (from the second case above). However, we do have $N(n, c) > 0$ and this contradicts to the formation of C. As a result we can be assured that there exists a $c - d$ path that will not go back to v. ■

Theorem 4 *There exists a $(2, 1, 0)$ generalized edge coloring for every graph.*

3.3 Power of 2

We now describe an algorithm that construct a $(2, 0, 0)$ for every graph with maximum degree as a power of 2, i.e., the graph $G = (V, E)$ has the maximum degree $D = 2^d$ for an positive integer d.

The basic idea of the construction is to divide the original graph G into two subgraphs, so that the maximum degrees of both subgraphs are equal. Recall that during the construction of $(2,0,0)$ g.e.d. for $D \leq 4$, we use a Euler cycle to color every edge so that that the number of 0-edges and 1-edges adjacent to a node differ by at most 1. Now we apply the alternating coloring process (Figure 3) to G, then divide the edges according to their colors. We have two induced subgraphs $G_0 = (V, E_0)$ and $G_1 = (V, E_1)$, where E_0 are those edges in G that are colored 0, and E_1 are those colored by 1. Both the maximum degree of G_0 and G_1 are 2^{d-1}. We can recursive apply this coloring process until the maximum degree is down to 4, and derive a $(2,0,0)$ for each subgraph. Now when we put all these g.e.c. together and view those colors in different g.e.c.'s as different colors, we have a $(2,0,l)$ g.e.c. (denoted by C) , where l is the maximum local discrepancy. Note that the key point of this construction is that we use only D colors to color the entire graph, so the global discrepancy is 0.

Next we need to reduce the local discrepancy of C for every node. Recall that during the construction for the $(2,1,0)$ in Section 3.2, we are able to convert a color c edge into a d edge, as long as they are adjacent to the same node v and there is no other edges colored by c or d adjacent to v. We now apply the same technique to the coloring C obtained in the previous step. As long as there exists a node v and two colors c and d so that $N(v,c) = N(v,d) = 1$, we convert the c edge into an d edge, without increasing the local discrepancy of other nodes. We repeat this step, just as we did in the construction of $(2,1,0)$, and eventually will convert C into a $(2,0,0)$ g.e.c.

Theorem 5 *There exists a $(2,0,0)$ generalized edge coloring for every graph with maximum degree as a power of 2.*

3.4 Bipartite graph

It is well known that given a bipartite graph with maximum degree D, we can find an edge coloring with D colors in polynomial time [6]. In our terminology, it is easy to compute a $(1,0,0)$ g.e.c. for bipartite graphs. By combining this $(1,0,0)$ g.e.c. with the concept of $c-d$ path, we are able to find $(2,0,0)$ g.e.c. for every bipartite graph.

Given a bipartite graph, the algorithm first finds an edge coloring with D colors. We then group the colors into $\lceil \frac{D}{2} \rceil$ new colors. This results in a $(2,0,l)$ g.e.c. where l is the new local discrepancy. We then examine every node v. If there are two colors c and d, so that $N(v,c) = N(v,d) = 1$, we find a $c-d$ path for them. Eventually we have a $(2,0,0)$ g.e.c.

Theorem 6 *There exists a $(2,0,0)$ generalized edge coloring for every bipartite graph.*

4 Conclusion

This paper introduces a new graph theory problem called generalized edge coloring. We show that when the number of edges of same color is limited to 3, it is impossible to find optimal g.e.c. for some graphs. However, when the limitation is 2, we have an optimal algorithm when D is no more than 4, or is a power of 2, or the graph is bipartite. In addition, if we are given one extra color, we can derive an generalized edge coloring that achieves optimal number of colors for each node. This is similar to the case of traditional edge coloring where k is 1 – finding an edge coloring with D colors is NP-complete, but it is always possible to color any graph with $D+1$ colors.

There are several interesting open problems along this line of research. For example, although it is impossible to find $(k,0,0)$ g.e.c. for $k \geq 3$, is it possible to find a $(k,1,0)$ solution, just like the case when k is 2? Also we can derive optimal g.e.c. for $k = 2$ and some special values of maximum degree D, is it true that we can always find optimal g.e.c. for any maximum degree D? The authors will continue the investigation on these interesting problems.

This paper is the result of a summer visit program, hosted by Institute of Information Science, Academia Sinica. The authors thank the institute for the support.

References

[1] Ieee 802.11a standard. http://standards.ieee.org/getieee802/download/802.11a-1999.pdf.

[2] Ieee 802.11b standard. http://standards.ieee.org/getieee802/download/802.11b-1999.pdf.

[3] V. Bahl, A. Adya, J. Padhye, and A. Wolman. Reconsidering the wireless lan platform with multiple radios. In *Workshop on Future Directions in Network Architecture (FDNA)*, 2003.

[4] J. A. Bonday and U.S.R. Murty. *Graph Theory with Applications.* American Elsevier, New York, 1976.

[5] T. R. Jensen and B. Toft. *Graph Coloring Problems.* Wiley Interscience, New York, 1995.

[6] D. König. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Mathematische Annalen*, 77:453–465, 1916.

[7] P. Kyasanur and N. H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *in Proceedings of IEEE WCNC*, 2005.

[8] D. W. Matula, G. Marble, and J. F. Issacson. *Graph Theory and Computing*, chapter Graph Coloring Algorithms. Academic Press, New York, 1972.

[9] J. Misra and David Gries. A constructive proof of Vizing's theorem. *Information Processing Letters*, 41(3):131–133, March 1992.

[10] A. Nasipuri and Samir R. Das. A multichannel csma mac protocol for mobile multihop networks. In *in Proceedings of IEEE WCNC*, 1999.

[11] A. Raniwala and T. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *in Proceedings of IEEE INFOCOM*, 2005.

[12] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM Mobile Computing and Communications Review (MC2R)*, 8:50–65, 2004.

[13] V. G. Vizing. On an estimate of the chromatic class of a p-graph (in Russian). *Diskret. Analiz*, 3:23–30, 1964.