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.). A 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. The concept of generalized edge coloring 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 are graphs that do not have generalized edge coloring that could achieve the minimum number of colors for every vertex. On the contrary, when $k = 2$ we show that if we are given one extra color, we can find a generalized edge coloring that uses the minimum number of colors for each vertex. In addition, we show that for certain classes of graphs we are able to find a generalized edge coloring that uses the minimum number of colors for every vertex *without* the extra color. These special classes of graphs include bipartite graph, graphs with a power of 2 maximum degree, or graphs with maximum degree no more than 4.

I. Introduction

Many modern wireless LAN standards, such as IEEE 802.11b/802.11g [1] and IEEE 802.11a [2], 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], [4], [5], [6]. 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], [6].

Specifically, we consider channel assignment that satisfies the following constraints. First, the total number of radio channels that can be assigned to an interface is bounded by the underlining architecture. For example, IEEE 802.11b/802.11g can use up to 11 channels in total. Second, the capacity of a radio channel within a communication range is bounded by a constant number k, so that an interface on a node can communicate with up to k neighboring nodes, and 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. For example, we consider the network in Figure 1. The k is set to 2 so at most two edges adjacent to the same vertex can be colored with the same color. The total number of colors, i.e., the total number of radio channels, used in this coloring is 3. The number of colors adjacent to node C is 2, so it requires two interface cards.

Graph coloring seems to be a natural formulation for this problem. However, standard vertex coloring [8], [9] (and more recently, vertex-multi-coloring) [10] 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

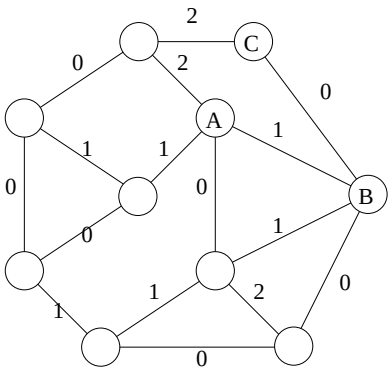

Fig. 1. An example of wireless network when $k = 2$. The number next to an edge is the assigned channel number.

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). The goals are to minimizing the total number of radio channels used in the network, and the number of network interface cards that we must install for each node. By minimizing the total number of channels used in the assignment, we are more likely to realize a network topology with the existing technology, e.g. the 11 channels in IEEE 802.11b/802.11g. By minimizing the number of network interface cards for each node, we minimize the total hardware costs to build a wireless mesh network.

It is easy to derive lower bounds for the total number of channels used in the wireless network, and the number of network interface cards on each node. Every generalized edge coloring will use at least $\lceil \frac{D}{k} \rceil$ radio channels, where D is the maximum degree of the wireless mesh network. Similarly, for each node with d neighbors, every generalized edge coloring will use at least $\lceil \frac{d}{k} \rceil$ network interfaces. We will use *discrepancy* to describe the quality of a generalized edge coloring. The *global discrepancy* of a g.e.c describes the difference between the lower bound $\lceil \frac{D}{k} \rceil$ and the actual number of radio channels used. The *local discrepancy* of a g.e.c describes the maximum among all nodes, the difference between the lower bound $\lceil \frac{d}{k} \rceil$ and the actual number of network interface cards used. A g.e.c. is optimal if it can achieve both zero discrepancy *globally and locally*.

Let us consider the network in Figure 1 when k is 2. The maximum degree D is 4 so the lower bound on the total number of colors is 2. Th coloring in Figure 1 uses three colors so the global discrepancy is 1. The local discrepancy of node B is 0 since it uses only two colors. However, the local discrepancy of node A is 1 since it has only 4 neighbors but uses 3 colors. Similarly node C has local discrepancy 1 since it has 2 neighbors but uses 2 colors. As a result the local discrepancy is therefore 1. This coloring is not optimal.

This papers shows that when k is 3, i.e., when a network interface can communicate with up to 3 neighbors, it is impossible to find an optimal generalized edge coloring for some graphs. However, when k is 2 and we are given one extra radio channel, we can derive a generalized edge coloring that achieves optimal number of interfaces for every node. In other words, with the price of *one* global discrepancy, we can achieve *zero* local discrepancy for every node. This result 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. In practice this is a very useful result since the new radio channels can be introduced by the advance of technology, but the number of network interface cards directly affects construction costs. Finally, when k is 2 we can also find optimal g.e.c. for several cases of special graphs: (1) the graph is bipartite, (2) when D is a power of 2, (3) or when D is no more than 4. The bipartite graph result is important since the topology matches the level-by-level relaying of wireless network.

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

II. 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 trivial lower bounds on the number of colors required for generalized edge coloring. 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 to be the 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 *local discrepancy* for a mapping function f is the maximum local discrepancy among all nodes, i.e. $\max_v(n(v) - \lceil \frac{d_v}{k} \rceil)$.

We use the global and the local discrepancy to evaluate the quality of a coloring function. The global discrepancy describes the "unnecessary" number of radio channels we used to construct a network, and the local discrepancy describes the "unnecessary" number of network interface cards we used for a node. Of course we would like to have a generalized edge coloring that minimizes both global and local discrepancy. As a result we define the quality of a coloring function as follows. 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 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.

III. RESULTS

We first show that there are graphs that do not have optimal generalized edge coloring when $k \geq 3$; i.e., 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 2 illustrates the constructed graph when k is 3.

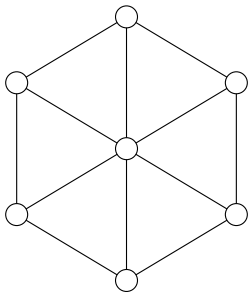

Fig. 2. A graph that does not have any optimal $(3, 0, 0)$ generalized edge coloring.

This result suggests that we need to relax the local discrepancy while dealing with the cases when k is 3 or larger. On the other hand, we show that when k is 2. we can always find optimal $(2, 0, 0)$ generalized edge coloring for certain classes of graphs.

A. Euler Cycle

We now find optimal $(2, 0, 0)$ generalized edge coloring for certain classes of graphs, and start with graphs that have maximum degree bounded by 4. 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 when the max degree of the graph is bounded by 4. 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 the transformation.

The second step is to remove some degree 2 nodes 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 3 (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 3 (b), we remove all but two nodes from the path. We denote the transformed graph as G^*.

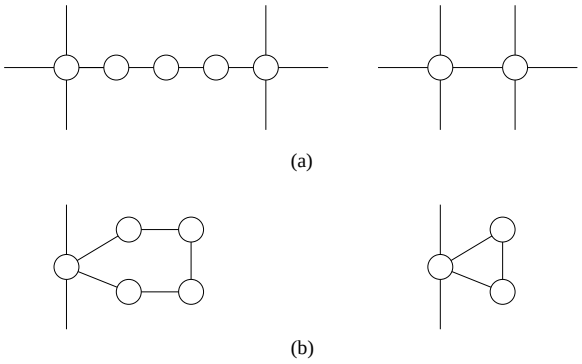

Fig. 3. Two cases to remove some degree 2 nodes.

Now we construct a Euler cycle for the transformed graph G^*. 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.

procedure AlternatingColoring

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

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

B. One Extra Color

We now describe an algorithm that finds $(2, 1, 0)$ generalized edge coloring for *every* graph. Notice that this result indicates that by having an extra color, i.e., an extra radio channel, we are able to achieve *zero* local discrepancy, i.e., zero unnecessary hardware cost for network interface cards. This tradeoff is practical since new radio channels can be easily introduced by the fast advance of technology, but the number of interface cards has a direct impact on the overall network infrastructure budget.

The result we will describe is very similar to the traditional edge coloring. It is well known that to determine if a graph has a $(1, 0, 0)$ g.e.c. is NP-complete, but it is always possible to find a $(1, 1, 0)$ g.e.c. in polynomial time by Vizing's theorem [11].

Our algorithm first finds a $(1, 1, 0)$ generalized edge coloring from Vizing's theorem, then it reduces the number of colors 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 $(2, 1, *)$ – the $*$ means "don't care". In other words, we reduce the global discrepancy to 1, and do not care about local discrepancy, which will

be taken care of later. To be more specific, the local discrepancy is bounded by $\frac{D}{4}$. 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 local discrepancy can go up to about $\frac{D}{4}$.

Now the important part is to reduce the 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 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 [12]. 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 increase the local discrepancy of any node along the path.

Suppose we can always 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.

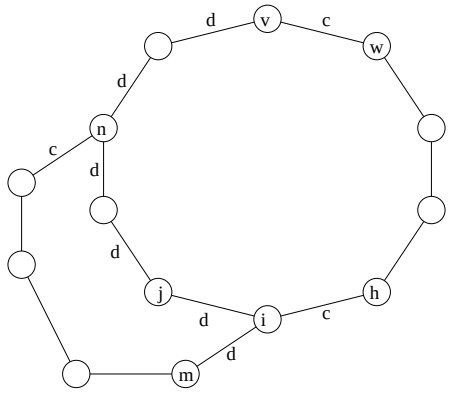

Fig. 5. There exists a $c - d$ path that starts from v but does not end at v.

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 5 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 5 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.

C. Power of 2

We now return to the quest of finding optimal $(2, 0, 0)$ generalized edge coloring for special classes of graphs. We first describe an algorithm that constructs a $(2, 0, 0)$ for every graph with maximum degree which is 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 4) 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 $(2, 0, *)$ g.e.c. C. 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 III-B, 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, a $(2, 0, *)$ g.e.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 which is a power of 2.

D. Bipartite graph

Now we study generalized edge coloring for bipartite graph. The reason we study bipartite graphs is as follows. In a wireless network usually there are certain nodes that are directly connected to the backbone. Depending on the distance to the backbone, the nodes can be arranged in level-by-level fashion so that those that are far away from the backbone can send information to the backbone by the relaying nodes between it and the backbone. As a result the nodes only need to communicate with those nodes in the adjacent levels, as indicated by Figure 6. The *entire* level-by-level graph is a bipartite graph.

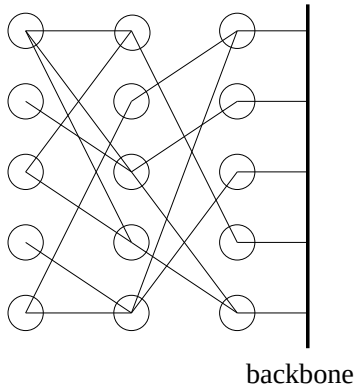

Fig. 6. A level-by-level connection graph in a wireless network.

Another reason to study bipartite graph is that it characterizes a *hierarchical data grid* model, due to its resemblance to hierarchical grid management, usually found in current grid systems [13], [14], [15], [16]. For example, in LCG (World-Wide Large Hadron Collider Computing Grid) [13] project 70 institutes from 27 countries form a grid system. The system is organized as a hierarchy, with CERN (the European Organization for Nuclear Research) as the root, or tier-0 site. There are 11 tier-1 sites directly under CERN that help

distribute data obtained from Large Hadron Collider (LHC) at CERN. Other tier-2 sites in LCG hierarchy receive data from its corresponding tier-1 site. The entire LCG grid can be represented as in Figure 7.

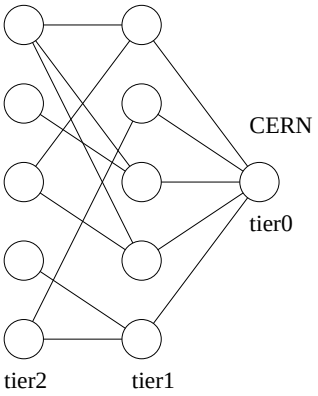

Fig. 7. A data grid connection graph in LGS grid system.

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 [17]. 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 $(2,0,*)$ g.e.c. 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.

IV. Conclusion

This paper introduces a new graph theory problem called generalized edge coloring. We show that when $k = 3$, there are graphs that do not have generalized edge coloring that could achieve the minimum number of colors for every vertex. On the contrary, when $k = 2$ we show that if we are given one extra color, we can find a generalized edge coloring that uses the minimum number of colors for each vertex. In addition, we show that for certain classes of graphs we are able to find a generalized edge coloring that uses the minimum number of colors for every vertex *without* the extra color. These special classes of graphs include bipartite graph, graphs with a power of 2 maximum degree, or graphs with maximum degree no more than 4.

There are several interesting open problems along this line of research. For example, although it is impossible to find $(k,0,0)$ generalized edge coloring for every graph when $k \geq 3$, is it possible to find a $(k,1,1)$ solution by relaxing the local discrepancy requirement? Also when $k = 2$ we can derive optimal generalized edge coloring for bipartite graphs and some special values of maximum degree D. Is it true that we can always find optimal generalized edge coloring for any graphs? 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 802.11b standard," http://standards.ieee.org/getieee802/download/802.11b-1999.pdf.
[2] "Ieee 802.11a standard," http://standards.ieee.org/getieee802/download/802.11a-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] A. Nasipuri and S. R. Das, "A multichannel csma mac protocol for mobile multihop networks," in *in Proceedings of IEEE WCNC*, 1999.
[5] 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.
[6] 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)*, vol. 8, pp. 50–65, 2004.
[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*. Academic Press, New York, 1972, ch. Graph Coloring Algorithms.
[9] J. A. Bonday and U. Murty, *Graph Theory with Applications*. American Elsevier, New York, 1976.
[10] T. R. Jensen and B. Toft, *Graph Coloring Problems*. Wiley Interscience, New York, 1995.
[11] V. G. Vizing., "On an estimate of the chromatic class of a p-graph (in Russian)," *Diskret. Analiz*, vol. 3, pp. 23–30, 1964.
[12] J. Misra and D. Gries, "A constructive proof of Vizing's theorem," *Information Processing Letters*, vol. 41, no. 3, pp. 131–133, Mar. 1992.
[13] W. L. C. Grid, "http://lcg.web.cern.ch/lcg/."
[14] W. B. David, "Evaluation of an economy-based file replication strategy for a data grid," in *International Workshop on Agent based Cluster and Grid Computing*, 2003, pp. 120–126.
[15] W. Hoschek, F. J. Janez, A. Samar, H. Stockinger, and K. Stockinger, "Data management in an international data grid project," in *In Proceedings of GRID Workshop*, 2000, pp. 77–90.
[16] K. Ranganathana and I. Foster, "Identifying dynamic replication strategies for a high performance data grid," in *In Proceedings of the International Grid Computing Workshop*, 2001, pp. 75–86.
[17] D. König, "Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre," *Mathematische Annalen*, vol. 77, pp. 453–465, 1916.