Theoretical Computer Science 410 (2009) 1022–1029

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Approximation algorithm for maximum edge coloring

Wangsen Feng [a,*], Li'ang Zhang [b], Hanpin Wang [b]

[a] *Key Laboratory of Network and Software Security Assurance, Computing Center, Peking University, Ministry of Education, Beijing 100871, China*

[b] *Key Laboratory of High Confidence Software Technologies, School of Electronic Engineering and Computer Science, Peking University, Ministry of Education, Beijing 100871, China*

ARTICLE INFO

Keywords:
Maximum edge coloring
Approximation algorithm
Wireless mesh networks

ABSTRACT

We propose a polynomial time approximation algorithm for a novel maximum edge coloring problem which arises from wireless mesh networks [Ashish Raniwala, Tzi-cker Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: INFOCOM 2005, pp. 2223–2234; Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, Mobile Comput. Commun. Rev. 8 (2) (2004) 50–65]. The problem is to color all the edges in a graph with maximum number of colors under the following q-Constraint: for every vertex in the graph, all the edges incident to it are colored with no more than q ($q \in \mathbb{Z},\ q \geq 2$) colors. We show that the algorithm is a 2-approximation for the case $q = 2$ and a $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for the case $q > 2$ respectively. The case $q = 2$ is of great importance in practice. For complete graphs and trees, polynomial time accurate algorithms are found for them when $q = 2$. The approximation algorithm gives a feasible solution to channel assignment in multi-channel wireless mesh networks.

1. Introduction

Graph coloring problems occupy an important place in graph theory. There are two types of coloring: vertex coloring and edge coloring. For vertex coloring, Brooks states that $\chi(G) \leq \Delta(G)$ for any graph G except complete graphs K_n and odd circles C_{2k+1}, where *chromatic number* $\chi(G)$ is the minimum number of colors needed in a vertex coloring of G [3]. Karp proves that to compute $\chi(G)$ is an NP-hard problem [4]. Garey and Johnson point out that there is even no polynomial time approximation algorithm with ratio 2 for the problem if P $\neq$ NP holds [5]. However, Turner designs an algorithm of time complexity $O(|V| + |E| \log k)$ with probability almost 1 to color any given k-colorable graph with k colors when k is not too large relative to $|V|$ [6]. For edge coloring, Vizing states that for any graph G, either $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$, where *chromatic index* $\chi'(G)$ is the minimum number of colors needed in an edge coloring of G [7]. Holyer proves that it is also an NP-hard problem to determine $\chi'(G)$ [8]. If the coloring solution is not necessary to be optimal, the proof of the Vizing Theorem yields an algorithm which can find an edge coloring solution using $\Delta(G) + 1$ colors. Uriel Feige et al. study the problem of maximum edge t-coloring in multigraphs [9]. It colors as many edges as possible using t colors, such that no two adjacent edges are colored with the same color. They show that the problem is NP-hard and design constant factor approximation algorithms for it.

All the above problems are traditional coloring problems, in the sense that they aim to find a *minimum* number of colors with some constraints. However, we propose a *maximum* edge coloring problem in this paper, which aims to color all the edges in a graph using as many colors as possible, with the q-Constraint: for every vertex in the graph, all the edges incident

* Corresponding author.
E-mail address: fengws@pku.edu.cn (W. Feng).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.10.035

to it are colored with no more than q ($q \in \mathbb{Z},\ q \geq 2$) colors. Two adjacent edges are not necessary to be colored with different colors, i.e. they can be colored with the same color.

1.1. Motivation

In 2005, Ashish Raniwala and Tzi-cker Chiueh [1] proposed a multi-channel wireless mesh network architecture (called *Hyacinth*) that equips each mesh network node with multiple 802.11 network interface cards (NICs). They point out that intelligent channel assignment is critical to *Hyacinth*'s performance. A series of experiments have been carried out and the results show that with 2 NICs on each node, it may improve the network throughput by a factor of 6 to 7 compared with the conventional single-channel ad hoc network architecture [1,2]. In such kind of channel assignment problem in multi-channel wireless mesh networks, a novel computational problem is involved. "How many channels can be used in the network at most from the mathematical point of view?" The value provides an upper bound on the number of channels the mesh network can hold. This bound is an important parameter for wireless mesh network designers.

Let V_N be the set of mesh routers in a mesh network, E_N be the set of pairs of mesh routers which can communicate directly, q be the number of network interface cards each node owns. A wireless mesh network can be modeled as a network graph $G_N = (V_N, E_N, q)$. Clearly, G_N is a connected undirected simple graph and the number of channels assigned to each node cannot exceed the number of its NICs: q. We formulate the problem as an edge coloring problem with the name "maximum edge coloring problem".

Maximum edge coloring problem: Given a connected undirected simple graph $G = (V, E)$ and a positive integer $q \geq 2$, how to color all the edges of E with maximum number of colors under the q-Constraint: for every vertex in V, all the edges incident to it are colored with no more than q colors?

We would like to say a few more sentences about the two restrictions in the above definition. First, the input graph is restricted to be connected in the definition. Unconnected graphs can also be solved if a solution to any connected graph can be found. Secondly, $q \geq 2$ is required in the definition. In fact, if $q = 1$, then every edge must be colored with the same color because of the connectivity of the graph. This case is trivial.

In this paper, notations are used in a standard way. For example, $ALG(G)$ is used to denote the number of colors used in the solution given by our algorithm on the input graph G; $OPT(G)$ to denote the number of colors used in an optimal coloring solution of G. $\Delta(G)$ is the maximum degree of the vertices in G, also called the degree of G. Please refer to [10] for more details on approximation algorithms.

The rest of the paper is organized as follows. In Section 2 two important properties of the problem are introduced. In Sections 3 and 4, we discuss the approximation algorithm and the approximation ratios for the case $q = 2$ and the case $q > 2$ respectively. Because the mesh routers in a wireless mesh network often have two network interface cards, the case $q = 2$ is very important. In Section 5, maximum edge coloring in complete graphs and trees for the case $q = 2$ is discussed. In Section 6, we conclude the paper with possible future research direction.

2. Preliminaries

Given a connected graph $G = (V, E)$, let $OPT(G) = m$, i.e. optimal coloring solutions of G use m colors: $1, 2, \ldots, m$. Based on the color of each edge, E can be divided into m subsets: $E_1, E_2, \ldots, E_m$, where E_i is the set of edges colored with color i ($1 \leq i \leq m$). A *character subgraph* of G is a subgraph induced by $e_1, e_2, \ldots, e_m$, where e_i is chosen from E_i ($1 \leq i \leq m$).

Lemma 1. *Let H be a character subgraph of a connected graph $G = (V, E)$, then*

(1) $\Delta(H) \leq q$;
(2) *H consists of paths and cycles, if $q = 2$;*
(3) $OPT(G) \leq |V|$, *if $q = 2$.*

Proof. (1) Because the colors of the edges in H are different from each other and the optimal solution satisfies the q-Constraint, the degree of each vertex v in H satisfies $1 \leq d_H(v) \leq q$. It yields $\Delta(H) \leq q$.
(2) If $q = 2$, it is clear that H is a set of paths and cycles, i.e. each connected component of H is either a path or a cycle.
(3) According to (2), $OPT(G) = m = |E(H)| \leq |V(H)| \leq |V|$. □

Lemma 2. *Given a vertex cover V^* of a graph $G = (V, E)$ with $|V^*| = k$, let H be the subgraph induced by V^* in G. Then:*

(1) $OPT(G) \leq kq$;
(2) *If H has a matching of size m, then $OPT(G) \leq kq - m$;*
(3) *If $q = 2$ and H is connected, then $OPT(G) \leq k + 1$;*
(4) *If $q = 2$ and H has l connected components ($1 \leq l \leq k$), then $OPT(G) \leq k + l$.*

Proof. (1) Since V^* is a vertex cover, every edge in G is incident to a vertex of V^* at least. On the other hand, the edges incident to V^* can be colored with $|V^*|q$ colors at most based on the q-Constraint. Thus, $OPT(G) \leq |V^*|q = kq$.
(2) Let M_H be a matching in H of size m. E can be divided into two non-intersecting parts: M_H and $E - M_H$. It is clear that $E - M_H$ can be colored with $(k - 2m)q + 2m(q - 1) = kq - 2m$ new colors at most, no matter how M_H is colored. On the other hand, M_H can be colored with m colors at most. Thus, $OPT(G) \leq (kq - 2m) + m = kq - m$.

(1) **If** $(q = 2)$ **Then**
 Compute a maximum matching M of G;
 Else
 Find a maximum subgraph H_{q-1} with $\Delta(H_{q-1}) \leq q-1$;
(2) Assign a new color to each edge in $M(H_{q-1})$;
(3) Delete the edges of $M(H_{q-1})$ from the original graph G and for each connected component of the residual graph G' which is not an isolated vertex, assign to it a new color;
(4) Output each edge with the color assigned to it.

Fig. 1. Algorithm 1.

(3) Suppose H is colored with x colors. By Lemma 1, $1 \leq x \leq k$. If H is connected, then there are at least $x-1$ vertices in V^* incident to two edges colored with different colors in H. Thus, $E - E(H)$ can be colored with $[k-(x-1)](q-1)+(x-1)(q-2) = k-x+1$ new colors at most. As a consequence, $OPT(G) \leq (k-x+1)+x = k+1$.

(4) Denote by $c_1, c_2, \ldots, c_l$ ($\sum_{i=1}^{l} c_i = k$) the number of vertices in the l connected components of H respectively. According to (3), $OPT(G) \leq \sum_{i=1}^{l}(c_i+1) = k+l$. □

3. Approximation algorithm

Clearly, if $\Delta(G) \leq q$, the number of colors used by an optimal solution is equal to $|E|$. Based on this fact, a greedy strategy is adopted to design the following approximation algorithm. The main idea is that, first, we find a maximum subgraph H in G with $\Delta(H) \leq q-1$, then assign a new color to each edge of the subgraph H. When dealing with the remaining edges, we must be more careful, because this procedure may lead to conflict, which means the q-Constraint could be broken. To avoid the conflict, a simple trick is employed as follows: delete the edges in H from the original graph G and just let every non-isolated vertex connected component of the residual graph G' share one new color.

The approximation algorithm is given in Fig. 1. A maximum matching can be found in $O(|V|^{\frac{1}{2}}|E|)$ time [11]. To show that it is a polynomial time algorithm, the classical maximum b-matching problem is introduced.

Maximum b-matching problem: Given an undirected graph $G = (V, E)$ and a function b: $V \rightarrow \mathbb{Z}^+$ specifying an upper bound for each vertex, the maximum b-matching problem asks for a maximum cardinality set $M_b \subseteq E$ such that $\forall\, v \in V,\ d_{M_b}(v) \leq b(v)$, where $d_{M_b}(v)$ is the degree of v in the subgraph induced by M_b.

The results on matchings are strongly self-refining, as was pointed out by Tutte [12,13], Edmonds and Johnson [14,15]. By applying splitting techniques to ordinary matchings, maximum b-matchings can be found in polynomial time too. Gabow [16] designed an algorithm of complexity $O(|V||E|\log|V|)$ for the maximum b-matching problem in 1983. As a consequence, the time complexity of Algorithm 1 is $O(|V||E|\log|V|)$.

4. Analysis

In this section, we will analysis the performance ratio of Algorithm 1 and show that it is a 2-approximation for the case $q = 2$ and $(1+\frac{4q-2}{3q^2-5q+2})$-approximation for the case $q > 2$.

4.1. Case $q = 2$

It can be seen that, if the graph is chosen to be a bipartite graph, Algorithm 1 is a factor 2 approximation algorithm. Since in bipartite graphs, there exists the equation $max_{matching\ M}|M| = min_{vertex\ cover\ U}|U|$, combined with Lemma 2, we have

$$\frac{OPT(G)}{ALG(G)} \leq \frac{2|U_{min}|}{|M_{max}|} \leq 2. \tag{1}$$

In fact, the ratio also stands in general graphs.

Theorem 1. *For any connected graph, Algorithm* 1 *achieves an approximation factor of* 2 *when* $q = 2$.

Proof. Given a connected graph $G = (V, E)$ with $OPT(G) = m$, let H be a character subgraph of G. By Lemma 1, H is a set of paths and cycles. The theorem is proved by two steps:

(1) Construct a matching in G with size $\geq \lfloor \frac{m}{2} \rfloor$ based on H.

(2) According to the result in (1), we can easily draw the conclusion:

$$\frac{OPT(G)}{ALG(G)} \leq 2. \tag{2}$$

Step (1): A path of odd(even) length is called an odd(even) path. Similarly, a cycle of odd(even) length is called an odd(even) cycle. Let p_1, p_2, c_1 and c_2 be the number of odd paths, even paths, odd cycles and even cycles in H. Denote all the odd paths in H by OP_i ($1 \le i \le p_1$), all the even paths by EP_j ($1 \le j \le p_2$), all the odd cycles by OC_s ($1 \le s \le c_1$) and all the even cycles by EC_t ($1 \le t \le c_2$). Use $l(OP_i)$ ($1 \le i \le p_1$), $l(EP_j)$ ($1 \le j \le p_2$), $l(OC_s)$ ($1 \le s \le c_1$) and $l(EC_t)$ ($1 \le t \le c_2$) to denote the lengths of OP_i, EP_j, OC_s and EC_t respectively. Clearly, for the paths or cycles of even length $2k$, the size of their maximum matchings is k. For the paths of odd length $2k+1$, the size is $k+1$, and for the cycles of odd length $2k+1$, the size is k. Let m be the number of edges in H, then:

$$m = \sum_{i=1}^{p_1} l(OP_i) + \sum_{j=1}^{p_2} l(EP_j) + \sum_{s=1}^{c_1} l(OC_s) + \sum_{t=1}^{c_2} l(EC_t). \tag{3}$$

And the size of a maximum matching M_H in H is:

$$|M_H| = \sum_{i=1}^{p_1} \frac{1}{2}[l(OP_i)+1] + \sum_{j=1}^{p_2} \frac{1}{2} l(EP_j) + \sum_{s=1}^{c_1} \frac{1}{2}[l(OC_s)-1] + \sum_{t=1}^{c_2} \frac{1}{2} l(EC_t). \tag{4}$$

Case 1: Clearly, if $c_1 = 0$, then $|M_H| \ge \lfloor \frac{m}{2} \rfloor$. M_H is the matching to be constructed.

Case 2: When $c_1 = 1$, there is only one odd cycle, OC_1, in H. We can construct a matching M' with $|M'| \ge \lfloor \frac{m}{2} \rfloor$ as follows:

If $G = H = OC_1$, then the original graph is just an odd cycle. We can let M' be a maximum matching of OC_1. Clearly, $|M'| \ge \lfloor \frac{m}{2} \rfloor$.

If OC_1 is a real subgraph of G, then there is at least one vertex $v \in V$ and $v \notin V(OC_1)$, since there is no other edge among the vertices of OC_1 in G. Clearly, there is no edge in G among those nonadjacent 2-degree vertices in H. For each 1-degree node of a path in H, it cannot be adjacent to two nonadjacent 2-degree vertices in H. Otherwise, it will contradict the fact that the optimal coloring solution is feasible. Because G is connected and $G \ne OC_1$, we can always find a vertex v_1 in OC_1 and v_1 is a neighbor of an outside vertex v_2, which is not in the cycle. Based on the above facts, v_2 must belong to one of the following three sets: $V_1 = \{$the vertices not in $H\}$; $V_2 = \{$the 1-degree vertices in even paths in $H\}$; $V_3 = \{$the 1-degree vertices in odd paths in $H\}$. Now, it is time to construct M'.

(1) If $v_2 \in V_1$, construct a maximum matching M_C of OC_1 leaving v_1 as an unsaturated vertex. Let $M'_C = M_C \cup \{e = (v_1, v_2)\}$. Clearly, $|M'_C| = \frac{1}{2}[l(OC_1)-1]+1 > \frac{1}{2} l(OC_1)$.

(2) If $v_2 \in V_2$, construct a maximum matching M_C of OC_1 leaving v_1 as an unsaturated vertex and find a maximum matching M_P of the even path EP_1 leaving v_2 as an unsaturated vertex. Let $M'_C = M_C \cup M_P \cup \{e = (v_1, v_2)\}$. Clearly, $|M'_C| = \frac{1}{2}[l(OC_1)-1] + \frac{1}{2} l(EP_1) + 1 > \frac{1}{2}[l(OC_1) + l(EP_1)]$.

(3) If $v_2 \in V_3$, construct maximum matchings M_C and M_P of OC_1 and the odd path OP_1 respectively. Let $M'_C = M_C \cup M_P$. Clearly, $|M'_C| = \frac{1}{2}[l(OC_1)-1] + \frac{1}{2}[l(OP_1)+1] = \frac{1}{2}[l(OC_1)+l(OP_1)]$.

For the remaining connected components in H, find one maximum matching M_R in them. Let $M' = M_R \cup M'_C$. Obviously, $|M'| \ge \lfloor \frac{m}{2} \rfloor$, M' is the matching to be constructed.

Case 3: When $c_1 > 1$, we construct a new graph G/H by shrinking every connected component H_i of H into a new vertex v_i ($1 \le i \le p_1+p_2+c_1+c_2$). Clearly, G/H has vertex set $(V/V(H)) \cup \{v_1, v_2, \ldots, v_{p_1+p_2+c_1+c_2}\}$. For each edge e in G, an edge of G/H is obtained from e by replacing any end point in H_i by the new vertex v_i. (Here we ignore loops and multiple edges that may arise.) Obviously, G/H is also connected. If there is an edge in G/H between an original vertex v that is not in H but in G and a new vertex coming from H_i, then there must be an edge in G between v and a vertex in H_i. If there is an edge in G/H between a new vertex from H_i and another new vertex from H_j, then there is an edge in G between a vertex in H_i and a vertex in H_j.

There are no edges among the new vertices from cycle components in G/H and each such vertex is only adjacent to vertices which are either new vertices from path components or original vertices. For convenience, new vertices from path components and original vertices are called *compatible vertices*. For each compatible vertex, it can be adjacent to two new vertices from cycle components at most. For one new vertex from a path component, if it is a neighbor of two new vertices from cycle components in G/H, then it must be that each of its two 1-degree nodes connects to a vertex in one of the two cycle components in G respectively.

Denote by $U = \{u_1, u_2, \ldots, u_{c_1}\}$ the set of new vertices from odd cycles. The following procedure is used to extract a set of compatible vertices from G/H which can dominate U. The graph output by the procedure is called matching graph B. (See Fig. 2)

(1) $B = \emptyset$;
(2) **while** $(U \ne \emptyset)$
{
1) Take an element u from U, scan its neighbors in G/H;
2) **if** (u is adjacent to a compatible vertex v by edge e and v is not adjacent to any other new vertex from an odd cycle) **then**
{ Add u, v and e into B, $U = U - \{u\}$; }
else if (u is adjacent to a compatible vertex v by edge e and v is also a neighbor of another new vertex from an odd cycle

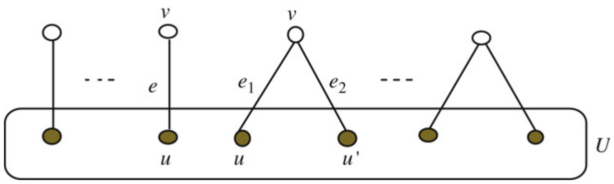

Fig. 2. Matching graph B: the filled vertices are new vertices from odd cycles, the empty ones correspond to compatible vertices.

which has been added into B)
then
{ Add u, v and e into B, $U = U - \{u\}$; }
else (in this case, u must be only adjacent to those compatible vertices which connect to two elements which are still in U at this time)
{ Suppose u is adjacent to a compatible vertex v by edge e_1 and v is also adjacent to another new vertex u' in U by edge e_2.
Add u, u', v and e_1, e_2 into B, $U = U - \{u, u'\}$. }
}
(3) output B.

For u_i in a path of length 1 in B, the case is similar to the case $c_1 = 1$. We pay more attention to the case of u_i in a path of length 2 in B. In this case, two new vertices from odd cycles are adjacent to the same compatible vertex. Denote by OC_1 and OC_2 the two odd cycles. Now, let us discuss how to construct M'.

(1) If the compatible vertex is an original vertex, say v, then we can always find v_1 in OC_1, v_2 in OC_2, which are neighbors of v in G. Construct a maximum matching M_C of OC_1 and OC_2 leaving v_1 as an unsaturated vertex. Let $M'_C = M_C \cup \{e = (v, v_1)\}$. Clearly, $|M'_C| = \frac{1}{2}[(l(OC_1) - 1) + (l(OC_2) - 1)] + 1 = \frac{1}{2}[l(OC_1) + l(OC_2)]$.

(2) If the compatible vertex is a new vertex from an even path EP_1, then we can always find v_1 in OC_1 and v_2 in OC_2 such that they are adjacent to the two 1-degree nodes, say v_3 and v_4, in EP_1 in G respectively. Construct a maximum matching M_C of OC_1 and OC_2 leaving v_1, v_2 as unsaturated vertices and find the maximum matching M_P in EP_1 leaving v_3 as a saturated vertex. Let $M'_C = M_C \cup M_P \cup \{e_1 = (v_1, v_3)\}$. Clearly, $|M'_C| = \frac{1}{2}[(l(OC_1)-1)+(l(OC_2)-1)]+\frac{1}{2}l(EP_1)+1 = \frac{1}{2}[l(OC_1)+l(OC_2)+l(EP_1)]$.

(3) If the compatible vertex is a new vertex from an odd path OP_1, then we can always find v_1 in OC_1 and v_2 in OC_2 such that they are adjacent to the two 1-degree nodes, say v_3 and v_4, in OP_1 in G respectively. Construct a maximum matching M_C of OC_1 and OC_2 leaving v_1, v_2 as unsaturated vertices and find the maximal matching M_P in OP_1 leaving v_3, v_4 as unsaturated vertices. Let $M'_C = M_C \cup M_P \cup \{e_1 = (v_1, v_3), e_2 = (v_2, v_4)\}$. Clearly, $|M'_C| = \frac{1}{2}[(l(OC_1) - 1) + (l(OC_2) - 1)] + \frac{1}{2}[l(OP_1) - 1] + 2 > \frac{1}{2}[l(OC_1) + l(OC_2) + l(OP_1)]$.

Thus we can always construct a matching M' of G with size $\geq \lfloor \frac{m}{2} \rfloor$ as follows:

(1) Construct a character subgraph H in G;
(2) **if** ($c_1 = 0$) **then** { let $M' = M_H$; }
else if (G is an odd cycle)
then { let M' be a maximum matching of G; }
else {
1) shrink G into G/H;
2) extract the matching graph B from G/H;
3) for each connected component in B, construct M'_C as above;
4) for the remaining connected components in H, which is not in B, construct a maximum matching M_R in them;
5) let $M' = (\bigcup M'_C) \cup M_R$;
}

Step (2): Since M is a maximum matching of G, thus

$$\frac{OPT(G)}{ALG(G)} \leq \frac{m}{|M| + 1} \leq \frac{m}{|M'| + 1} \leq \frac{m}{\lfloor \frac{m}{2} \rfloor + 1} \leq \frac{m}{\frac{m}{2}} = 2. \quad (5)$$

Here, we assume that the residual graph $G' = G - M$ has at least one edge. If G' has no edge, then $M = G$. Thus, $ALG(G) = OPT(G) = |E|$, Theorem 1 follows immediately. □

The graph shown in Fig. 3 gives a tight example for the case $q = 2$.

Example 1. In the graph shown in Fig. 3, the set of vertical edges is a maximum matching of G. On the other hand, G can be colored with $2m$ colors at most. Thus, $ALG(G) = m + 1$, $OPT(G) = 2m$.

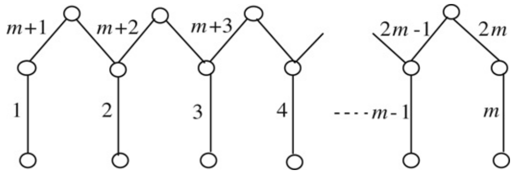

Fig. 3. Tight example for the case $q = 2$.

4.2. Case $q > 2$

Theorem 2. *For any connected graph G, Algorithm* 1 *achieves an approximation factor of* $1 + \frac{4q-2}{3q^2-5q+2}$ *when* $q > 2$.

Proof. According to Lemma 1, the number of edges in the maximum subgraph H_q with $\Delta(H_q) \le q$ in G is an upper bound of $OPT(G)$, i.e. $|E(H_q)| \ge OPT(G)$. On the other hand, the number of edges in the maximum subgraph H_{q-1} with $\Delta(H_{q-1}) \le q-1$ is a lower bound of $ALG(G)$, i.e. $ALG(G) \ge |E(H_{q-1})|$. Suppose H_q has s vertices of degree q: $v_1, v_2, \ldots, v_s$. Let M_D be a maximum matching in the subgraph D induced by $\{v_1, v_2, \ldots, v_s\}$ in H_q and $|M_D| = t$. Then we can pick out all the t edges in M_D to turn the $2t$ saturated vertices of degree q into $2t$ vertices of degree $q-1$. For each of the remaining $s-2t$ unsaturated vertices of degree q in H_q, pick out one edge incident to it. Thus we pick out $s-t$ edges from H_q totally and get a subgraph H with $\Delta(H) \le q-1$. Obviously, $|E(H_q)| - (s-t) = |E(H)| \le |E(H_{q-1})|$ and $|E(H_q)| \ge \frac{1}{2}sq$. Thus, the approximation ratio satisfies:

$$\frac{OPT(G)}{ALG(G)} \le \frac{|E(H_q)|}{|E(H_q)| - (s-t)} = 1 + \frac{s-t}{|E(H_q)| - (s-t)} = 1 + \frac{1}{\frac{|E(H_q)|}{s-t} - 1}. \tag{6}$$

Now, we need only consider the lower bound of $\frac{|E(H_q)|}{s-t}$. It is easy to see that the $s-2t$ M_D-unsaturated vertices form an independent set in H_q. These unsaturated vertices are either neighbors of the left $2t$ saturated vertices of degree q or neighbors of those vertices of degree $\le q-1$ not in D but in H_q. We can imagine that if t is too small, then some of the $s-2t$ unsaturated vertices must be adjacent to the vertices of degree $\le q-1$, which is not in D but in H_q.

First, consider how many degrees the $2t$ saturated vertices can provide to connect to $s-2t$ unsaturated vertices at most. Since M_D is a maximum matching of the subgraph D, the M_D-saturated vertices cannot connect to the unsaturated vertices arbitrarily in order to avoid M_D-augmenting paths. For the two end nodes of an edge e in M_D, they can connect to the unsaturated vertices in two ways only:

(1) Both of them are adjacent to the same unsaturated vertex, and they cannot connect to any other unsaturated vertex any more. It follows that they can provide two degrees for unsaturated vertices.
(2) Only one of them is adjacent to some unsaturated vertices, and the other end node cannot connect to any other unsaturated vertex any more. It yields that they can provide $q-1$ degrees at most.

Noting that $q > 2$, we can conclude that for the two end nodes of e, they can provide $q-1$ degrees at most. Thus the $2t$ saturated vertices can provide $t(q-1)$ degrees to connect to the unsaturated vertices at most. On the other hand, the $s-2t$ unsaturated vertices can provide $(s-2t)q$ degrees.

If $t(q-1) < (s-2t)q$, then there must be some of the $s-2t$ unsaturated vertices adjacent to other vertices of degree $\le q-1$ in H_q. By solving the inequality $t(q-1) \le (s-2t)q$, we get $t \le \frac{sq}{3q-1}$. It is natural for us to discuss the lower bound of $\frac{|E(H_q)|}{s-t}$ according to the value of $t \in [0, \lfloor \frac{s}{2} \rfloor]$:

Case 1: $t \ge \frac{sq}{3q-1}$. Clearly, $|E(H_q)| \ge \frac{1}{2}sq$. It is easy to see that the lower bound of $\frac{|E(H_q)|}{s-t}$ can be obtained at $t = \frac{sq}{3q-1}$:

$$\frac{|E(H_q)|}{s-t} \ge \frac{\frac{1}{2}sq}{s-t} \ge \frac{\frac{1}{2}sq}{s - \frac{sq}{3q-1}} = \frac{3q^2-q}{4q-2}. \tag{7}$$

Case 2: $t \le \frac{sq}{3q-1}$. In this case, t is so small that some of the $s-2t$ unsaturated vertices are adjacent to other vertices of degree $\le q-1$ in H_q. Thus we can give a larger lower bound of $|E(H_q)|$. It is not hard to get: $|E(H_q)| \ge \frac{1}{2}sq + \frac{1}{2}[(s-2t)q - t(q-1)]$. Thus,

$$\frac{|E(H_q)|}{s-t} \ge \frac{\frac{1}{2}sq + \frac{1}{2}[(s-2t)q - t(q-1)]}{s-t} = f(t) = \frac{sq - \frac{3}{2}tq + \frac{1}{2}t}{s-t}. \tag{8}$$

In order to find the minimum value of $f(t)$ in the interval $[0, \frac{sq}{3q-1}]$, we calculate the differential coefficient: $f'(t)$.

$$f'(t) = \frac{s - sq}{2(s-t)^2} < 0. \tag{9}$$

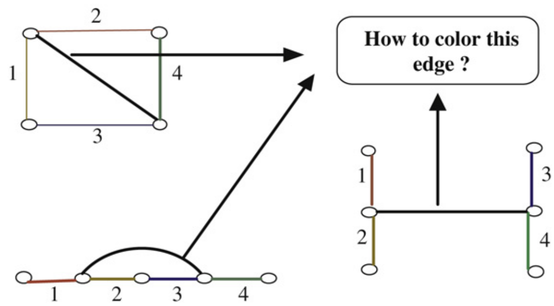

Fig. 4. There are no k-cycles and k-paths ($k \geq 4$) in H, nor multiple paths of length ≥ 2.

Thus $f(t)$ is a decreasing function. As a consequence, when $t = \frac{sq}{3q-1}$, $f(t)$ reaches its minimum value. We have

$$\frac{|E(H_q)|}{s-t} \geq f(t) \geq f\left(\frac{sq}{3q-1}\right) = \frac{3q^2-q}{4q-2}. \quad (10)$$

As a result, the lower bound of $\frac{|E(H_q)|}{s-t}$ is always $\frac{3q^2-q}{4q-2}$. The approximation ratio:

$$\frac{OPT(G)}{ALG(G)} \leq 1 + \frac{1}{\frac{|E(H_q)|}{s-t} - 1} \leq 1 + \frac{1}{\frac{3q^2-q}{4q-2} - 1} = 1 + \frac{4q-2}{3q^2-5q+2}. \quad \square \quad (11)$$

5. Maximum edge coloring in complete graphs and trees

For complete graphs and trees, we can get an accurate solution for the case $q = 2$. Obviously, $OPT(K_3) = 3$. For K_n ($n \geq 4$), Theorem 3 stands.

Theorem 3. *For a complete graph K_n ($n \geq 4$), $OPT(K_n) = \lfloor \frac{n}{2} \rfloor + 1$.*

Proof. Case 1: $n = 4$ or 5. It is easy to verify that the number of colors used in an optimal coloring solution is 3, that is $\lfloor \frac{n}{2} \rfloor + 1$.

Case 2: $n > 5$. $OPT(K_n) \geq ALG(K_n) = \lfloor \frac{n}{2} \rfloor + 1$. To prove the theorem, we need to show that $OPT(K_n) \leq \lfloor \frac{n}{2} \rfloor + 1$. Let H be a character subgraph of K_n with $E(H) = m = OPT(K_n)$. We will show that H is either the union of a 2-path (path of length 2) and $m-2$ discrete edges or the union of a 3-path (path of length 3) and $m-3$ discrete edges.

First, we prove that there are no cycles in H. For cycles of length $\geq$4, it can be seen that, we cannot assign a color to the diagonal. For cycles of length 3, it can be inferred that all the other edges in the complete graph must be colored with one of the three colors. However, $OPT(K_n) \geq ALG(K_n) = \lfloor \frac{n}{2} \rfloor + 1 > 3$ ($n > 5$), a contradiction. As a consequence, there are no cycles in H. Secondly, it can be proved that there is no k-path ($k \geq 4$) in H and there is at most one 2-path or one 3-path in H analogously. (See Fig. 4.)

Now, we are sure that H is either the union of a 2-path and $m-2$ discrete edges or the union of a 3-path and $m-3$ discrete edges. Thus, the size of a maximum matching in H: $|M_H| = m-1 = OPT(K_n)-1$. Obviously, the size of a maximum matching of the complete graph: $|M| = \lfloor \frac{n}{2} \rfloor$. $OPT(K_n) - 1 = |M_H| \leq |M| = \lfloor \frac{n}{2} \rfloor$, that is, $OPT(K_n) \leq \lfloor \frac{n}{2} \rfloor + 1$. As a result, $OPT(K_n) = \lfloor \frac{n}{2} \rfloor + 1 = ALG(K_n)$. $\square$

A vertex in a tree is called an internal vertex if and only if it is of degree at least two. If a tree is just an edge, then there is no internal vertex in it.

Theorem 4. *For any tree T, $OPT(T) = |V_{in}| + 1$, where V_{in} is the set of internal vertices in T.*

Proof. If T is just an edge, then $V_{in} = \emptyset$ and $OPT(T) = 1 = |V_{in}| + 1$. Otherwise, the set of internal vertices, V_{in}, is a vertex cover of T and the subgraph induced by V_{in} is also a tree, which is connected. By Lemma 2, $OPT(T) \leq |V_{in}| + 1$. In the following, a simple algorithm is introduced which can give a solution with $|V_{in}| + 1$ colors. (See Figure 5 for an example.)

(1) For the edges incident to the root, assign to them two colors;
(2) Process the left internal nodes from top to down, from left to right.
For the current node, assign one new color to all the edges from it to its children;
(3) When all the internal nodes are processed, all the edges are colored. Output the solution.

It is easy to see that the algorithm outputs a feasible solution and $ALG(T) = |V_{in}| + 1$. As a result, $OPT(T) \geq ALG(T) = |V_{in}| + 1$, $OPT(T) = |V_{in}| + 1$. $\square$

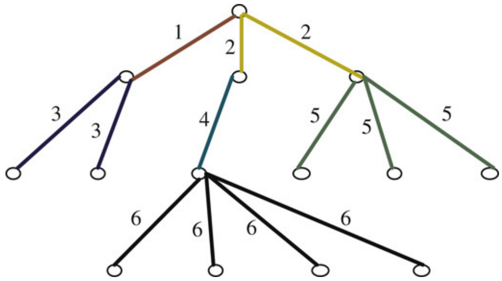

Fig. 5. An example for tree coloring.

6. Conclusion

This paper investigates a novel maximum edge coloring problem which arises from wireless mesh networks. We have designed a polynomial time approximation algorithm and shown that it is a 2-approximation for the case $q = 2$ and $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for the case $q > 2$ respectively. However, we do not know the complexity of the problem. The corresponding decision problem can be defined as: **Maximum-edge-coloring** $= \{(G, q, k) | (G, q)$ has a k-color solution$\}$. Obviously, it belongs to the NP class. We conjecture that it is NP-complete.

Acknowledgements

We would like to thank Xiaoming Li and Wei Yan's group for providing the problem. The authors were supported by the National High-tech Research and Development 863 Program of China under Grant No. 2006AA01Z160.

References

[1] Ashish Raniwala, Tzi-cker Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: INFOCOM 2005, pp. 2223–2234.
[2] Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, Mobile Comput. Commun. Rev. 8 (2) (2004) 50–65.
[3] R.L. Brooks, On colouring the nodes of a network, Proc. Cambridge Phil. Soc. 37 (1941) 194–197.
[4] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[5] M.R. Garey, D.S. Johnson, The complexity of near optimal graph coloring, J. ACM 23 (1976) 43–49.
[6] J.S. Turner, Almost all k-colorable graphs are easy to color, J. Algorithms 9 (1988) 63–82.
[7] V.G. Vizing, On an estimate of the chromatic class of a p-graph, Diskret. Anal. 3 (1964) 25–30 (in Russian).
[8] I.J. Holyer, The NP-completeness of edge-coloring, SIAM J. Comput. 10 (1981) 718–720.
[9] Uriel Feige, Eran Ofek, Udi Wieder, Approximating maximum edge coloring in multigraphs, in: APPROX 2002, pp. 108–121.
[10] Vijay V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin, 2001.
[11] S. Micali, Vijay V. Vazirani, An $O(|V|^{\frac{1}{2}}|E|)$ algorithm for finding maximum matching in general graphs, in: Proc. 21st IEEE FOCS, 1980, pp. 17–27.
[12] W.T. Tutte, The factors of graphs, Canad. J. Math. 4 (1952) 314–328.
[13] W.T. Tutte, A short proof of the factor theorem for finite graphs, Canad. J. Math. 6 (1954) 347–352.
[14] J. Edmonds, E.L. Johnson, Matching: A well-solved class of integer linear programs, in: Combinatorial Structures and Their Applications, Gordon and Breach, New York, 1970, pp. 89–92.
[15] J. Edmonds, E.L. Johnson, Matching, Euler tours and the Chinese postman, Math. Program. 5 (1973) 88–124.
[16] H.N. Gabow, An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems, in: Proc. 15th ACM STOC, 1983, pp. 448–456.