Soft Edge Coloring

Yoo-Ah Kim[1], Seungjoon Lee[2], and Min-ho Shin[2]

[1] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA. E-mail: ykim@engr.uconn.edu.
[2] Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail: slee,mhshin@cs.umd.edu.

Abstract. We consider a variant of edge coloring problem, which arises in multi-channel wireless networks. We are given a graph $G = (V, E)$, and constraint C_v for each v, which is the number of colors that edges incident to v can use. In addition, we have a constraint on the number of colors available in the network (denoted as C_G). For a pair of edges incident to a vertex, they are said to be *conflicting* if the colors assigned to them are the same. Our goal is to color edges so that we can minimize the number of conflicts while satisfying constraints on the number of colors that can be used. We first consider a homogeneous network where $C_v = k$ for all node v. For an arbitrary k, the problem is NP-hard by a reduction from edge coloring. We show that a simple greedy algorithm gives a solution with at most $(1 - \frac{1}{k})|E|$ more conflicts than the optimal. Moreover, we show that the approximation result is best possible unless $P = NP$. In a heterogenous network, each node can have different C_v. We show that the problem is NP-hard even when $C_v = 1$ or 2. For the case when $C_v = 1$ or k, we present two algorithms. One algorithm is completely combinatorial and gives a solution with conflicts at most $(2 - \frac{1}{k})OPT + (1 - \frac{1}{k})|E|$. We also present an SDP-based algorithm, which gives a solution with at most $1.122OPT + 0.122|E|$ conflicts for $k = 2$ and $(2 - \frac{1}{k} - \frac{1}{5(k-1)})OPT + (1 - \frac{k}{5(k-1)^2})|E|$ conflicts for $k > 2$.

1 Introduction

We consider a problem arising in channel assignment in a multi-channel wireless network. In wireless networks, due to the broadcast property of the medium, nearby nodes *interfere* with each other and cannot simultaneously transmit over the same wireless channel. One way to overcome this limitation is to assign independent channels (that can be used without interference) available in the system. Consider the example shown in Figure 1. When all links use the same channel, only one pair of nodes can communicate with each other at a time due to conflicts. However, if there are three channels available and each node has two wireless interface card (so it can use two channels), then we can assign a different channel to each link and there is no conflicts among edges in this channel assignment. Channel assignment to utilize multiple channels have recently been studied by many researchers in networking community [1–4].

We informally define the SOFT EDGE COLORING problem as follows. We are given a graph $G = (V, E)$, and constraints on the number of wireless cards C_v for all v, which limit the number of colors that edges incident to v can use. In addition, we have a constraint on the number of channels available in the network (denoted as C_G). For a

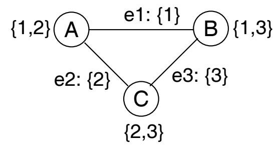

Fig. 1. In this network, each node has two wireless interface cards (thus can use two different channels) and three channels are available in total. We can assign a distinct channel to each link as shown above. With the above channel assignment, there is no conflict among edges.

pair of edges incident to a vertex, they are said to be *conflicting* if the colors assigned to them are the same. Our goal is to color edges (assign channels) so that we can minimize the number of conflicts while satisfying constraints on the number of colors that can be used.

SOFT EDGE COLORING is a variant of the EDGE COLORING problem. In our problem, coloring need not be proper (two adjacent edges are allowed to use the same color) but the goal is to minimize the number of such conflicts. In addition, each node has its local color constraint, which limits the number of colors that can be used by the edges incident to the node. For example, if a node has two wireless cards ($C_v = 2$), the node can choose two colors and edges incident to the node should use only those two colors.

Our results. We briefly summarize our results. We first consider a homogeneous network where $C_v = k$ and $C_G \geq C_v$ for all node v. For an arbitrary k, the problem is NP-hard by a reduction from EDGE COLORING. We show that a simple greedy algorithm gives a solution with at most $(1 - \frac{1}{k})|E|$ more conflicts than the optimal. Moreover, we show that this approximation result is best possible unless $P = NP$.

In a heterogeneous network, each node can have different C_v. We show that the problem is NP-hard even when $C_v = 1$ or 2, and $C_G = 2$. For the case when $C_v = 1$ or k, we present two algorithms. The first algorithm is completely combinatorial and gives the solution with conflicts at most $(2 - \frac{1}{k})OPT + (1 - \frac{1}{k})|E|$. We also present an SDP-based algorithm. When $C_v = 1$ or 2, the algorithm gives a solution with at most $1.122OPT + 0.122|E|$ conflicts. When $C_v = 1$ or k ($k > 2$), it gives $(2 - \frac{1}{k} - \frac{1}{5(k-1})OPT + (1 - \frac{k}{5(k-1)^2})|E|$ conflicts, which is slightly better than the combinatorial algorithm.

Related Work. Kann *et al.* [5] studied the MIN k-PARTITION problem where we color vertices with k different colors so that the total number of conflicts (monochromatic edges) is minimized. Our problem for a homogeneous network when $C_G = C_v = k$ for all v can be considered as MIN k-PARTITION problem when the given graph is a line graph[3]. They showed that for $k > 2$ and for every $\epsilon > 0$, there exists a constant α such that the MIN k-PARTITION cannot be approximated within $\alpha|V|^{2-\epsilon}$ unless $P = NP$.

[3] The line graph of G has a vertex corresponding to each edge of G, and there is an edge between two vertices in the line graph if the corresponding edges are incident on a common vertex in G.

Fitzpatrick and Meertens [6] have considered a variant of graph coloring problem (called the SOFT GRAPH COLORING problem) where the objective is to develop a distributed algorithm for coloring vertices so that the number of conflicts is minimized. The algorithm repeatedly recolors vertices to quickly reduce the conflicts to an acceptable level. They have studied experimental performance for regular graphs but no theoretical analysis has been provided. Damaschke [7] presented a distributed soft coloring algorithm for special cases such as paths and grids, and provided the analysis on the number of conflicts as a function of time t. In particular, the conflict density on the path is given as $O(1/t)$ when two colors are used, where the conflict density is the number of conflicts divided by $|E|$.

In the traditional edge coloring problem, the goal is to find the minimum number of colors required to have a proper edge coloring. The problem is NP-hard even for cubic graphs [8]. For a simple graph, a solution using at most $\Delta + 1$ colors can be found by Vizing's theorem [9] where Δ is the maximum degree of a node. For multigraphs, there is an approximation algorithm which uses at most $1.1\chi' + 0.8$ colors where χ' is the optimal number of colors required [10] (the additive term was improved to 0.7 by Caprara *et al.* [11]).

We have also developed several soft edge coloring heuristics for more general cases (heterogeneous networks in different interference models) and are performing experimental studies[12].

1.1 Problem definition

We are given a graph $G = (V, E)$ where $v \in V$ is a node in a wireless network and an edge $e = (u, v) \in E$ represents a communication link between u and v. Each node v can use C_v different channels and the total number of channels that can be used in the network is C_G. More formally, let $E(v)$ be the edges incident to v and $c(e)$ be the color assigned to e. Then $|\bigcup_{e \in E(v)} c(e)| \leq C_v$ and $|\bigcup_{e \in E} c(e)| \leq C_G$.

A pair of edges e_1 and e_2 in $E(v)$ are said to be conflicting if the two edges use the same color. Let us define the *conflict number* (CF_e) of an edge $e \in E$ to be the number of other edges that conflict with e. In other words, for an edge $e = (u, v)$, CF_e is the number of edges (other than e itself) in $E(u) \bigcup E(v)$ that use the same channel as e. Our goal is to minimize the total number of conflicts. That is,

$$CF_G = \frac{1}{2} \sum_{e \in E} CF_e. \tag{1}$$

In the remainder of this paper, we mean *channels* by *colors* and use edge coloring and channel assignment, interchangeably. We also use conflicts and interferences interchangeably.

2 A Greedy Algorithm for Homogeneous Networks

In this section, we consider the channel assignment problem in a homogeneous network where for all node v, the number of channels that can be used is the same ($C_v = k$).

For an arbitrary k, the problem is NP-hard as we can reduce the edge coloring problem to our problem by setting $k = C_G = \Delta$ where Δ is the maximum degree of nodes

We present a greedy algorithm for this problem. The algorithm can be performed in a distributed manner and each node needs only local information. The algorithm works as follows (see Algorithm 1). We use colors from $\{1, \ldots, k\}$ (We only use k colors even when $C_G > k$ and the approximation ratio of our algorithm is the same regardless of the value of C_G.) For any uncolored edge $e = (u, v)$, we choose a color for edge e that introduces the smallest number of conflicts. More formally, when we assign a color to $e = (u, v)$, we count the number of edges in $E(u) \bigcup E(v)$ that are already colored with c (denoted as $n(c, e)$), and choose color c with the smallest $n(c, e)$.

Algorithm 1 **Greedy Algorithm**

```
for each edge e = (u, v) do
  for each color i do
    compute the number of edges in E(u) and E(v) using color i.
  end for
  let c be the color with min n(i, e) for all colors i.
  assign color c to edge e.
end for
```

Theorem 1. *When we use the greedy algorithm, the total number of conflicts in homogeneous networks is at most*

$$CF_G = \frac{1}{2} \sum_{e \in E} CF_e \leq OPT + (1 - \frac{1}{k})|E|. \tag{2}$$

To prove the theorem, we first derive a lowerbound.

Lemma 2. *The total number of conflicts when $C_v = k$ for all node v is at least*

$$\frac{1}{2} \sum_{v} \frac{d_v^2}{k} - |E|.$$

Proof. For a node v, let $E(v)$ be the edges incident to v. $E(v)$ will be partitioned into k sets according to colors assigned to edges. Let $E_i(v)$ be a set of edges with color i. $E(v) = \bigcup_i E_i(v)$. Let $c(e)$ be the color assigned to e. Then

$$\begin{aligned} \frac{1}{2} \sum_{e} CF_e &= \frac{1}{2} \sum_{e=(u,v)} (|E_{c(e)}(v)| + |E_{c(e)}(u)| - 2) \\ &= \frac{1}{2} \sum_{v} \sum_{e \in E(v)} (|E_{c(e)}(v)| - 1) \end{aligned}$$

$$= \frac{1}{2} \sum_v \sum_i |E_i(v)|(|E_i(v)| - 1)$$

$$= \frac{1}{2} \sum_v \sum_i |E_i(v)|^2 - |E|$$

Note that for each node v, $\sum_i |E_i(v)|^2$ is minimized when the size of $E_i(v)$ is the same for all colors i. Therefore, we have

$$\frac{1}{2} \sum_e CF_e \geq \frac{1}{2} \sum_v (\frac{d_v}{k})^2 \cdot k - |E| = \frac{1}{2} \sum_v \frac{d_v^2}{k} - |E|.$$

□

The following lemma gives an upperbound on the number of conflicts in our solution.

Lemma 3. *The total number of conflicts introduced by Algorithm 1 is at most*

$$\frac{1}{2} \sum_v \frac{d_v^2}{k} - \frac{|E|}{k}.$$

Proof. To upperbound the number of conflicts by the greedy algorithm, consider an edge $e = (u, v)$. Let $n(e)$ be the number of conflicts that e introduces when it gets colored. The total number of conflicts in the final coloring is $\sum_e n(e)$. Since we choose a color for e such that it introduces the smallest conflicts in node u and v, $n(e)$ is at most $\lfloor \frac{d_v(e)+d_u(e)}{k} \rfloor$ where $d_v(e)$ and $d_u(e)$ are the number of edges that get colored *before* e in $E(v)$ and $E(u)$, respectively. Therefore, the total number of conflicts by the greedy algorithm is:

$$CF_G = \sum_e n(e) \leq \sum_{e=(u,v)} \lfloor \frac{d_v(e) + d_u(e)}{k} \rfloor$$

$$\leq \sum_v \sum_{e \in E(v)} \frac{d_v(e)}{k} = \frac{1}{k} \sum_v \sum_{i=0}^{d_v - 1} i$$

$$= \frac{1}{k} \sum_v \frac{d_v(d_v - 1)}{2} = \frac{1}{k} \sum_v \frac{d_v^2}{2} - \frac{1}{k} \sum_v \frac{d_v}{2}$$

$$= \frac{1}{2} \sum_v \frac{d_v^2}{k} - \frac{|E|}{k}.$$

□

Theorem 1 follows from Lemma 2 and 3. The following corollary will be useful to prove approximation factors for heterogenous networks.

Corollary 4. *Given an optimal solution, let $OPT(S)$ ($S \subseteq V$) be the number of conflicts at vertices in S and $|E(S)|$ be $\sum_{v \in S} \frac{d_v}{2}$.*

$$OPT(S) \geq \frac{1}{2} \sum_{v \in S} \frac{d_v^2}{k} - |E(S)|.$$

Proof. The lemma can be proven in a similar way to Lemma 2. □

In fact, we can show that the approximation ratio given by the greedy algorithm is best possible unless $P = NP$.

Theorem 5. *It is NP-hard to approximate the channel assignment problem in homogeneous networks within an additive term of $o(|E|^{1-\epsilon})$, given a constant ϵ.*

Proof. Suppose that we have a simple graph $G = (V, E)$. It is known that finding the edge chromatic number $\chi'(G)$ of G is NP-hard (the edge chromatic number is the minimum number of colors for edge-coloring G) [8]. By the Vizing's theorem [9], the chromatic index of a simple graph G is Δ or $\Delta + 1$ where Δ is the maximum degree of any vertex $v \in V$.

Given a constant ϵ, let $G' = (V', E')$ be the graph which has $|E|^{\frac{1}{\epsilon}-1}$ copies of G. Note that $|E'| = |E|^{\frac{1}{\epsilon}}$. We set $C_G = C_v = \Delta$ If $\chi'(G) = \Delta$ then the optimal solution of the channel assignment problem is 0. Otherwise if $\chi'(G) = \Delta + 1$, then each of component of G' has at least one conflict and therefore, the optimal solution has at least $|E|^{\frac{1}{\epsilon}-1}$ conflicts, which is the same as $|E'|^{1-\epsilon}$. Thus if we have an approximation algorithm with additive term of $o(|E'|^{1-\epsilon})$ for a graph $G' = (V, E')$, we can decide the chromatic index of G, which is NP-hard. Contradiction. □

3 Algorithms for Heterogenous Networks

We first prove that the problem is NP-hard even when $C_v = 1$ or 2, and $C_G = 2$.

Theorem 6. *The channel assignment problem to minimize the number of conflicts is NP-hard even when $C_v = 1$ or 2, and $C_G = 2$.*

Proof. See Appendix A. □

In the following sections, we present two algorithms for networks with $C_v = 1$ or 2 and analyze the approximation factors of the algorithms. The case where $C_v = 1$ or 2 is still interesting since (i) as shown above, even in this case, the problem becomes NP-hard. (ii) it reflects a realistic setting, in which most of mobile stations are equipped with one interface card and some of them have two wireless cards. The results can be generalized to the case where $C_v = 1$ or k.

3.1 Extended Greedy Algorithm

We present an extended greedy algorithm when $C_v = 1$ or k, and $C_G \geq C_v$. The approximation factor is $2 - \frac{1}{k}$. Even though the algorithm based on SDP (semi-definite programming) gives a better approximation factor (see Section 3.2), the greedy approach gives a simple combinatorial algorithm. The algorithm generalizes the idea of the greedy algorithm for homogeneous networks. In this case, an edge cannot choose its color locally since the color choice of an edge can affect colors for other edges to obey color constraints.

Before describing the algorithm, we define some notations. Let $V_i \subseteq V$ be the set of nodes v with $C_v = i$ (i.e., we have V_1 an V_k). V_1 consists of connected clusters $V_1^1, V_1^2, \ldots V_1^t$, such that nodes $u, v \in V_1$ belong to the same cluster if and only if there is a path composed of nodes in V_1 only. (See Figure 4 in Appendix for example.) Let E_1^i be a set of edges both of which endpoints are in V_1^i. We also define B_1^i to be a set of edges whose one endpoint is in V_1^i and the other is in V_k. We can think of B_1^i as a set of edges in the boundary of cluster V_1^i. Note that all edges in $E_1^i \bigcup B_1^i$ should have the same color. E_k is a set of edges both of which endpoints are in V_k. E_1 is defined to be $\bigcup_i E_1^i$.

In the greedy algorithm for homogeneous networks, each edge greedily chooses a color so that the number of interferences it creates (locally) is minimized. Similarly, when $C_v = 1$ or k, edges in the same cluster V_1^i choose a color so that the number of conflicts it creates is minimized. Formally, we choose a color c with minimum value of $\sum_{e=(u,v)\in B_1^i, v\in V_k} n_c(v)$ where $n_c(v)$ is the number of edges $e' \in E(v)$ with color c. Note that, however, it is not easy to choose the color in distributed manner since all edges in the cluster should agree to have the same color. Once edges in $E_1^i \bigcup B_1^i$ for all i choose their colors, the remaining edges (edges belonging to E_k) greedily choose their colors. Algorithm 2 describes the extended greedy algorithm.

Algorithm 2 Extended Greedy Algorithm

```
for each cluster V_1^i do
  (choose a color for edges in E_1^i ⋃ B_1^i as follows)
  if B_1^i is empty then
    choose any color for E_1^i.
  else
    for each color c ∈ {1, ..., k} do
      count the number of conflicts to be created when we choose color c for E_1^i ⋃ B_1^i.
      Formally, count ∑_{e=(u,v)∈B_1^i, v∈V_k} n_c(v) where n_c(v) is the number of edges e' ∈
      E(v) with color c.
    end for
    choose a color c that minimizes ∑_{e=(u,v)∈B_1^i, v∈V_k} n_c(v).
  end if
end for
for each edge that belongs to E_k do
  choose a color using the greedy algorithm in Section 2.
end for
```

Any edges (u, v) incident to a vertex in V_1 should use the same color and therefore are conflicting with each other no matter what algorithm we use. Given an optimal solution, consider $OPT(V_1)$ and $OPT(V_k)$ where $OPT(S)$ is the number of conflicts at vertices in $S \subseteq V$. Similarly, we have $CF(V_1)$ and $CF(V_k)$ where $CF(S)$ is the number of conflicts at vertices in $S \subseteq V$ in our solution. Then we have $OPT(V_1) = CF(V_1)$. Therefore, we only need to compare $OPT(V_k)$ and $CF(V_k)$.

Theorem 7. *The number of conflicts created by the extended greedy algorithm at* V_k *is at most* $(2-1/k)OPT(V_k)+(1-1/k)|E(V_k)|$ *where* $|E(V_k)|$ *is* $\sum_{v\in V_k}\frac{d_v}{2}$.

Proof. For each $e \in E\setminus E_1$, $n(e)$ be the number of conflicts at vertices in V_k which are introduced when we assign a channel to e. Then the total number of conflicts at V_k is $\sum n(e)$.

We first consider the number of conflicts created when we assign colors to edges in B_1^i (recall that B_1^i is a set of edges of which endpoints are in V_1^i and V_k). For an edge $e=(u,v)$ where $u \in V_1^i$ and $v \in V_k$, let $d_v(e)$ be the number of edges in $E(v)$ to which a color is assigned before e. Then when we choose a color for $E_1^i \bigcup B_1^i$, the number of conflicts at vertices in V_k with edges not in $E_1^i \bigcup B_1^i$, is at most

$$\frac{\sum_{v\in V_k}\sum_{e\in E(v)\bigcap B_1^i} d_v(e)}{k}$$

as we choose a color with minimum conflicts. If v has $e_i(v)$ edges in B_1^i, $\frac{1}{2}e_i(v)(e_i(v)-1)$ additional conflicts (between edges in B_1^i) are created.

For edges in E_k we use the greedy algorithm presented in Section 2. Therefore, the number of conflicts created when we assign colors in E_k is at most

$$\frac{\sum_{v\in V_k}\sum_{e\in E(v)\bigcap E_k} d_v(e)}{k}.$$

Summing up all the conflicts,

$$\sum_{e\in E\setminus E_1} n(e) \le \frac{\sum_{v\in V_k}\sum_{e\in E(v)} d_v(e)}{k} + \frac{\sum_i \sum_{v\in V_k}(e_i^2(v)-e_i(v))}{2}$$

For each node v, we have

$$\sum_{e\in E(v)} d_v(e) \le \frac{d_v(d_v-1)}{2} - \frac{\sum_i (e_i^2(v)-e_i(v))}{2}$$

as colors for edges in B_1^i will be determined at the same time.

Therefore, we have

$$\begin{aligned}
\sum_{e\in E\setminus E_1} n(e) &\le \frac{1}{k}\sum_{v\in V_k}\Big(\frac{d_v(d_v-1)}{2} - \frac{\sum_i e_i^2(v)-e_i(v)}{2}\Big) + \frac{\sum_i\sum_{v\in V_k}(e_i^2(v)-e_i(v))}{2}\\
&= \frac{1}{2}\sum_{v\in V_k}\frac{d_v(d_v-1)}{k} + (1-\frac{1}{k})\frac{\sum_i\sum_{v\in V_k}(e_i^2(v)-e_i(v))}{2}\\
&= \frac{1}{2}\sum_{v\in V_k}\frac{d_v^2}{k} - \frac{1}{2}\sum_{v\in V_k} d_v + (1-\frac{1}{k})\frac{\sum_i\sum_{v\in V_k}(e_i^2(v)-e_i(v))}{2} + \frac{1}{2}(1-\frac{1}{k})\sum_{v\in V_k} d_v\\
&\le (2-\frac{1}{k})OPT(V_k) + (1-\frac{1}{k})|E(V_k)|.
\end{aligned}$$

where $OPT(V_k)$ is the optimal number of conflicts at vertices in V_k and $|E(V_k)|$ be $\sum_{v \in V_k} \frac{d_v}{2}$.

The last inequality comes from the fact that both $\frac{1}{2}\sum_{v \in V_k} \frac{d_v^2}{k} - \frac{1}{2}\sum_{v \in V_k} d_v$ (by Corollary 4) and $\frac{\sum_i \sum_{v \in V_k} (e_i^2(v) - e_i(v))}{2}$ are lower bounds of the optimal solution. □

Corollary 8. *The number of conflicts created by the extended greedy algorithm when $C_v = 1$ or k is at most $(2 - 1/k)OPT + (1 - 1/k)|E|$.*

Proof.

$$
\begin{aligned}
CF(V) &= CF(V_1) + CF(V_k) \\
&\leq OPT(V_1) + (2 - 1/k)OPT(V_k) + (1 - 1/k)|E(V_k)| \\
&\leq (2 - 1/k)OPT + (1 - 1/k)|E|.
\end{aligned}
$$

□

When $k = 2$, the extended greedy algorithm gives a 1.5-approximation (assuming that $d(v)$ is much greater than k). Note that the approximation ratio remains the same for any $C_G \geq k$. In the following section, we obtain a better approximation factor using SDP relaxation when $C_v = 1$ or k and $C_G = k$.

3.2 SDP-based algorithm

In this subsection, we assume that k different channels are available in the network and all nodes have 1 or k wireless cards. We formulate the problem using semidefinite programming. Consider the following vector programming (VP), which we can convert to an SDP and obtain an optimal solution in polynomial time.

We have an m-dimensional unit vector Y_e for each edge e ($m \leq n$).

VP:

$$\min \sum_v \sum_{e_1, e_2 \in E(v)} \frac{1}{k}((k-1)Y_{e_1} \cdot Y_{e_2} + 1) \tag{3}$$

$$|Y_e| = 1 \tag{4}$$

$$Y_{e_1} \cdot Y_{e_2} = 1 \quad \text{if } C_v = 1,\ e_1, e_2 \in E(v) \tag{5}$$

$$Y_{e_1} \cdot Y_{e_2} \geq \frac{-1}{k-1} \ \text{ for } \ e_1, e_2 \in E(v) \tag{6}$$

We can relate a solution of VP to a channel assignment as follows. Consider k unit length vectors in m-dimensional space such that for any pair of vectors v_i and v_j, the dot product of the vectors is $-\frac{1}{k-1}$. (It has been shown that $-\frac{1}{k-1}$ is the minimum possible value of the maximum of the dot products of k vectors [13, 14].) Given an optimal channel assignment of the problem, we can map each channel to a vector v_i. Y_e takes the vector that corresponds to the channel of edge e. If C_v is one, all edges incident to v should have the same color. The objective function is exactly the same as the number of conflicts in the given channel assignment since if $Y_{e_1} = Y_{e_2}$ (e_1 and e_2 have the same color), it contributes one to the objective function, and 0 otherwise. Thus the optimal solution of the VP gives a lower bound of the optimal solution in the channel assignment problem.

The above VP can be converted to a semidefinite programming (SDP) and solved in polynomial time (within any desired precision) [15–19], and given a solution for the SDP, we can find a solution to the corresponding VP, using incomplete Cholesky decomposition [20].

It remains to round the solution to the VP into a feasible channel assignment. We use the rounding technique used for MAXCUT by Goeman and Williamson [21] when $k = 2$ and show that the expected number of interferences in the solution is at most $1.122OPT + 0.122|E|$.When $k > 2$, we obtain the approximation guarantee of $2 - \frac{1}{k} - \frac{1}{5(k-1)}$ with additive term of $(1 - \frac{k}{5(k-1)^2})|E|$, using the rounding algorithm for MAX k-CUT [13].

When $k = 2$: We select a random unit vector r, and assign channel one to all edges with $Y_e \cdot r \geq 0$ and channel two to all other edges.

Lemma 9. *[21] For $-1 \leq t \leq 1$, $\frac{\arccos t}{\pi} \geq \frac{\alpha}{2}(1-t)$, where $\alpha > .87856$.*

Theorem 10. *The expected number of total conflicts at V_k by our algorithm is at most $1.122OPT(V_k) + 0.122|E(V_k)|$.*

Proof. Let X_{ij} be 1 if e_i and e_j have the same color for $e_i, e_j \in E(v)$. For any vertex v with $C_v = 1$, all edges in $E(v)$ should have the same color in any solution. Therefore, we only consider edges $e_i, e_j \in E(v)$ for vertices with $C_v = 2$. Let $c(e)$ be the color assigned to edge e.

$$\begin{aligned}
E[X_{ij}] &= Pr(c(e_i) = c(e_j)) \\
&= 1 - Pr(c(e_i) \neq c(e_j)) \\
&= 1 - 2Pr(Y_{e_i} \cdot r \geq 0,\ Y_{e_j} \cdot r < 0) \\
&= 1 - \frac{\arccos(Y_{e_i} \cdot Y_{e_j})}{\pi} \\
&\leq 1 - \frac{\alpha}{2}(1 - Y_{e_i} \cdot Y_{e_j}) \\
&= (1-\alpha) + \frac{\alpha}{2}(Y_{e_i} \cdot Y_{e_j} + 1)
\end{aligned}$$

The total number of such conflicts is $X = \sum X_{ij}$. Let $OPT(S)$ ($S \subseteq V$) be the optimal number of conflicts at vertices in S and $|E(S)|$ be $\sum_{v \in S} \frac{d_v}{2}$.

$$\begin{aligned}
\sum_{v \in V_k} \sum_{e_i, e_j \in E(v)} E[X_{ij}] &= \sum_{v \in V_k} \sum_{e_i, e_j \in E(v)} ((1-\alpha) + \frac{\alpha}{2}(Y_{e_i} \cdot Y_{e_j} + 1)) \\
&\leq \sum_{v \in V_k} \sum_{e_i, e_j \in E(v)} (1-\alpha) + \alpha OPT(V_k)
\end{aligned}$$

$$\leq (1-\alpha)\sum_{v\in V_k}\frac{d_v^2-d_v}{2}+\alpha OPT(V_k)$$
$$=(1-\alpha)\sum_{v\in V_k}\frac{d_v^2}{2}-(1-\alpha)|E(V_k)|+\alpha OPT(V_k)$$
$$=2(1-\alpha)(\sum_{v\in V_k}\frac{d_v^2}{4}-|E(V_k)|)+(1-\alpha)|E(V_k)|+\alpha OPT(V_k)$$
$$\leq 2(1-\alpha)OPT(V_k)+\alpha OPT(V_k)+(1-\alpha)|E(V_k)|$$
$$\leq (2-\alpha)OPT(V_k)+(1-\alpha)|E(V_k)|.$$

By Lemma 9, we have the theorem. □

Corollary 11. *The expected number of total conflicts by our algorithm is at most* $1.122OPT+0.122|E|$.

When $k>2$: We use the rounding algorithm for MAX k-CUT when $k>2$ [13]. Given an optimal solution for VP, we obtain a coloring as follows. We first select k random vectors, denoted as $R=\{r_1,r_2,\cdots,r_k\}$. Each random vector $r_i=(r_{i,1},r_{i,2},\dots r_{i,n})$ is selected by choosing each component $r_{i,j}$ independently at random from a standard normal distribution $N(0,1)$. For each edge e, assign e to vector r_i if r_i is the closest vector to Y_e (i.e., the vector with the maximum value of $Y_e\cdot r_i$). Ties are broken arbitrarily. For $k>2$ we can obtain the approximation guarantee of $2-\frac{1}{k}-\frac{1}{5(k-1)}$. Let $\beta_{ij}=Y_{e_i}\cdot Y_{e_j}$.

Lemma 12. *[13]* $E[X_{ij}]\leq\frac{1}{k}+\frac{\beta_{ij}}{5(k-1)}$ *when* $\beta_{ij}<0$

Lemma 13. *For* $0\leq\beta_{ij}\leq 1$, $E[X_{ij}]\leq\frac{1}{k}((k-1)\beta_{ij}+1)$.

Proof. As $E[X_{ij}]=\frac{1}{k}((k-1)\beta_{ij}+1)$ when $\beta_{ij}=0$ and 1, and $E[X_{ij}]$ is a convex function in $[0,1]$ [13], we have the lemma. □

Note that for any pair of edges e_i,e_j incident to a vertex $v\in V_1$, $\beta_{ij}=1$ by Constraints (5). We now analyze $E[X_{ij}]$ for $e_i,e_j\in E(v)$ for some $v\in V_k$. Let P include all pairs of edges in $E(v)$ for any $v\in V_k$. For a pair $(i,j)\in P$, (i,j) is included in pP (positive pairs)$\subseteq P$ if $\beta_{ij}\geq 0$ and (i,j) is included in nP (negative pairs) $\subseteq P$ if $\beta_{ij}<0$. We define $Val_1(S)$ to be $\frac{1}{k}|S|$ for any set $S\subseteq P$. In addition, let $Val_2(S)$ be $\frac{1}{k}\sum_{(i,j)\in S}((k-1)\beta_{ij}+1)$.

Lemma 14. *The number of conflicts at V_k by SDP-based algorithm is at most* $Val_2(P)+(\frac{5(k-1)^2}{k}-1)\Delta$ *where* $\Delta=\frac{-1}{5(k-1)}\sum_{(i,j)\in nP}\beta_{ij}$.

Proof. By Lemma 12 and Lemma 13, we have

$$\sum_{(i,j)\in P}E[X_{ij}]\leq Val_2(P)-(\frac{k-1}{k}\sum_{(i,j)\in nP}\beta_{ij}-\frac{1}{5(k-1)}\sum_{(i,j)\in nP}\beta_{ij})$$
$$\leq Val_2(P)+(\frac{5(k-1)^2}{k}-1)\Delta.$$

□

Lemma 15. *The number of conflicts at* V_k *by SDP-based algorithm is at most* $Val_1(P) - \Delta + Val_2(P)(1-\frac{1}{k})$ *where* $\Delta = \frac{-1}{5(k-1)}\sum_{(i,j)\in nP}\beta_{ij}$.

Proof. By Lemma 12, the number of conflicts between a pair of edges in nP is at most $Val_1(nP) - \Delta$. Due to Lemma 13 for pairs in pP, the number of conflicts is at most $Val_1(pP) + \frac{k-1}{k}\sum_{(i,j)\in pP}\beta_{ij}$. Thus the total conflict is given as

$$\begin{aligned}\sum_{(i,j)\in P} E[X_{ij}] &\le Val_1(nP) - \Delta + Val_1(pP) + \frac{k-1}{k}\sum_{(i,j)\in pP}\beta_{ij} \\ &= Val_1(P) - \Delta + \frac{k-1}{k}\sum_{(i,j)\in pP}\beta_{ij} \\ &\le Val_1(P) - \Delta + Val_2(P)(1-\frac{1}{k}).\end{aligned}$$

The last inequality comes from the fact that $Val_1(pP) + \frac{k-1}{k}\sum_{(i,j)\in pP}\beta_{ij} = Val_2(pP)$ and $Val_1(pP) = \frac{1}{k}|pP| \ge \frac{1}{k}\sum_{(i,j)\in pP}\beta_{ij}$.

□

Theorem 16. *The number of conflicts at* V_k *created by SDP-based algorithm is at most* $(2-\frac{1}{k}-\frac{1}{5(k-1)})OPT(V_k) + (1-\frac{k}{5(k-1)^2})|E(V_k)|$.

Proof. By Lemma 14 and 15, the number of conflicts at V_k created by SDP-based algorithm is upperbounded by $\min(Val_1(P) - \Delta + Val_2(P)(1-\frac{1}{k}), Val_2(P) + (\frac{5(k-1)^2}{k} - 1)\Delta)$. It is maximized when $\Delta = \frac{1}{5(k-1)^2}(kVal_1(P) - Val_2(P))$, which gives

$$(1-\frac{k}{5(k-1)^2})Val_1(P) + (1-\frac{1}{k}+\frac{1}{5(k-1)^2})Val_2(P).$$

Note that $Val_1(P) \le OPT(V_k) + |E(V_k)|$ and $Val_2(P) \le OPT(V_k)$. Therefore, the total conflict at V_k is at most

$$(2-\frac{1}{k}-\frac{1}{5(k-1)})OPT(V_k) + (1-\frac{k}{5(k-1)^2})|E(V_k)|.$$

□

Corollary 17. *The number of conflicts created by SDP-based algorithm is at most* $(2-\frac{1}{k}-\frac{1}{5(k-1)})OPT + (1-\frac{k}{5(k-1)^2})|E|$.

4 Discussion

Note that in all of our algorithms the total number of different colors used in the network is only $\max C_v$ rather than C_G. Although it is clear that if we use more channels then we can reduce the number of conflicts (see Figure 1), it is not easy to make sure that each edge has at least one channel which are available in both endpoints if we allow to use colors more than $\max C_v$. Moreover, it may be possible that the size of the set of common channels is small, which may create more conflicts. One possible solution is to further improve the solution by recoloring edges with additional colors after obtaining the solution by the algorithm. It will be an interesting future work to analyze how much we can improve the performance by such recoloring. Another way to utilize more channels available in the network, we allow some edges in the network to be dropped. In our recent results [4, 12], we have developed several distributed heuristics and performed experimental evaluation using this idea.

References

1. A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-ch annel wireless mesh network," in *Proceedings of Infocom*, March 2005.
2. P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interfa ce wireless networks," in *Proceedings of WCNC*, March 2005.
3. Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li, "Joint channel assignment and routing for throughput optimization in mu lti-radio wireless mesh networks," in *In Proceedings of ACM MobiCom*, 2005.
4. Min ho Shin, Seungjoon Lee, and Yoo-Ah Kim, "Distributed channel assignment for multi-radio wireless networks," in *The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2006.
5. V. Kann, S. Khanna, J. Lagergren, and A. Panconesi, "On the hardness of max k-cut and its dual," in *Proc. 5th Israel Symposium on Theory and Computing Systems (ISTCS)*, 1996, pp. 61–67.
6. Stephen Fitzpatrick and Lambert Meertens, "An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs," in *1st Symposium on Stochastic Algorithms, Foundations and Applications (SAGA)*, 2001.
7. Peter Damaschke, "Distributed soft path coloring," in *STACS '03: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, London, UK, 2003, pp. 523–534, Springer-Verlag.
8. Ian Holyer, "The np-completeness of edge-coloring," *SIAM J. Computing*, vol. 10, no. 4, pp. 718–720, 1981.
9. V. G. Vizing, "On an estimate of the chromatic class of a p-graph (russian)," *Diskret. Analiz.*, vol. 3, pp. 25–30, 1964.
10. T. Nishizeki and K. Kashiwagi, "On the 1.1 edge-coloring of multigraphs," *SIAM J. Disc. Math.*, vol. 3, no. 3, pp. 391–410, August 1990.
11. Alberto Caprara and Romeo Rizzi, "Improving a family of approximation algorithms to edge color multigraphs," *Information Processing Letters*, vol. 68, no. 1, pp. 11–15, 1998.
12. Hieu Dinh, Yoo-Ah Kim, Seungjoon Lee, and Min ho Shin, "Algorithms for channel assignment in multi-channel wireless networks," In preparation.
13. A. FRIEZE and M. JERRUM, "Improved approximation algorithms for MAX k-CUT and MAX BISECTION," in *Integer Programming and Combinatorial Optimization*, Egon Balas and Jens Clausen, Eds., vol. 920, pp. 1–13. Springer, 1995.
14. D. Karger, R. Motwani, and M. Sudan, "Approximate graph coloring by semidefinite programming," in *Proc. 35th IEEE Symposium on Foundations of Computer Science*, 1994, pp. 2–13.
15. F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995.
16. M. Grotschel, L. Lovasz, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, pp. 169–197, 1981.

17. M. Grotschel, L. Lovasz, and A. Schrijver, "Geometric algorithms and combinatorial optimization," *Springer-Verlag*, 1987.
18. V. Nesterov and A. Nemirovskii, "Self-concordant functions and polynomial time methods in convex programming," *Central Economical and Mathematical Institute, U.S.S.R. Academy of Science, Moscow*, 1990.
19. V. Nesterov and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming," *SIAM*, 1994.
20. G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.
21. M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, pp. 1115–1145, 1995.

A Proof of Theorem 6

A node v is defined to have a balanced assignment if for each color i used by any edge in $E(v)$, the number of edges assigned to color i is exactly $\frac{d_v}{C_v}$. A network has the minimum number of conflicts if every node has a balanced assignment. Given an instance C of the problem 3SAT, we construct a graph G, in which each node has a balanced assignment if and only if C is satisfiable.

We need three types of components — inverting components, variable setting components, and satisfaction testing components. Figures 2 and 3 in Appendix show the components we need. In each component, a black or gray node has $C_v = 1$, and a white node has $C_v = 2$. In inverting components, if the input pair of edges use the same channel, the output pair should use different channels (and vice versa) for a white node to have a balanced assignment. In a component, a pair of input or output edges are said to be $true$ if the same channel is assigned to the pair, and $false$ if different channels are assigned to them. (To assign *true* to a pair of edges we may choose either channel 1 or 2.) The inverting component can be used to obtain the invert of a variable.

Using the variable setting components, we can set pairs of edges to be either true or false. We need to have as many pairs as there are appearances of variable v_i or $\neg v_i$ in C. Note that the specific channel assigned to each edge can be chosen as we want when we assign true or false to a pair of edges. For example, we can either use channel one or two for true assignments.

For each clause c_j in C, we have one satisfaction testing component (see Figure 3). In a satisfaction testing component, a white node has a balanced assignment if and only if at least one of three pairs is true. That is, if all three pairs are false, then we have exactly three edges with channel one and three edges with channel two for the three pairs, which prevents the white node from having a balanced assignment. On the other hand, if at least one is true, we can find an assignment of either $(5, 1)$ or $(4, 2)$ for the three pairs ((i, j) means that i edges have channel one and j edges have channel two), and there are balanced assignments for both cases.

B Figures

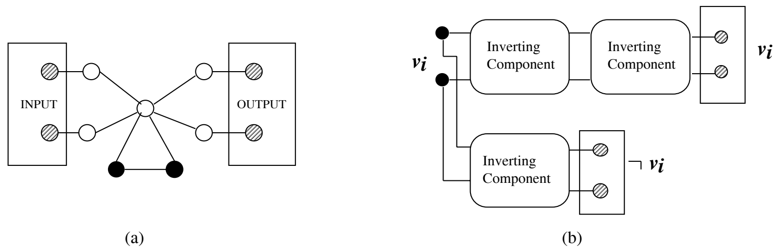

Fig. 2. Black and gray nodes have $C_v = 1$ and white nodes have $C_v = 2$. $C_G = 2$. (a) Inverting Components (b) Variable setting components.

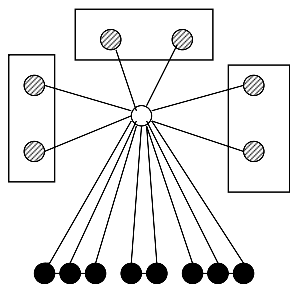

Fig. 3. Satisfaction Testing Component

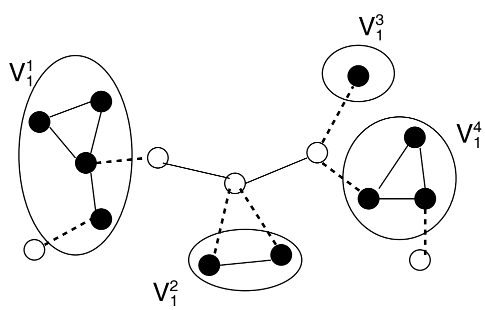

Fig. 4. The figure show an example of clusters V_1^i when $C_v = 1$ or k. Black nodes have only one wireless card and white nodes have k wireless cards. Dotted lines belong to B_1^i.