ELSEVIER

Theoretical Computer Science 262 (2001) 649–667

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Approximation algorithms for channel assignment with constraints ☆

Jeannette Janssen[a,*], Lata Narayanan[b]

[a] *Department of Mathematics and Statistics and Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada B3H 3J5*

[b] *Department of Computer Science, Concordia University, Montreal, Quebec, Canada H3G 1M8*

Received June 1999; revised July 2000; accepted August 2000
Communicated by M.S. Paterson

**Abstract**

Cellular networks are generally modeled as node-weighted graphs, where the nodes represent cells and the edges represent the possibility of radio interference. An algorithm for the *channel assignment* problem must assign as many channels as the weight indicates to every node, such that any two channels assigned to the same node satisfy the *co-site* constraint, and any two channels assigned to adjacent nodes satisfy the *inter-site* constraint. We describe several approximation algorithms for channel assignment with arbitrary co-site and inter-site constraints for odd cycles and the so-called *hexagon* graphs that are often used to model cellular networks. The algorithms given for odd cycles are optimal for some values of constraints, and have performance ratio at most $1 + 1/(n-1)$ for all other cases, where $n$ is the length of the cycle. Our main result is an algorithm of performance ratio at most $\frac{4}{3} + \frac{1}{100}$ for hexagon graphs with arbitrary co-site and inter-site constraints. 

*Keywords:* Frequency assignment; Approximation algorithms; Radio colouring

## 1. Introduction

The demand for wireless telephony and other services has been growing dramatically over the last decade. As a result of this, radio spectrum resources are scarce, and their efficient use becomes of critical importance. The cellular concept was proposed as an early solution to the problem of spectrum congestion. By dividing the service area into small coverage areas called *cells* served by low power transmitters, it became possible to *reuse* the same frequencies in different cells, provided they are far enough apart.

☆ Research supported in part by NSERC, Canada.

* Corresponding author. Fax: +(902)-494-5130.
*E-mail addresses:* janssen@mscs.dal.ca (J. Janssen), lata@cs.concordia.ca (L. Narayanan).


PII: S0304-3975(00)00388-1

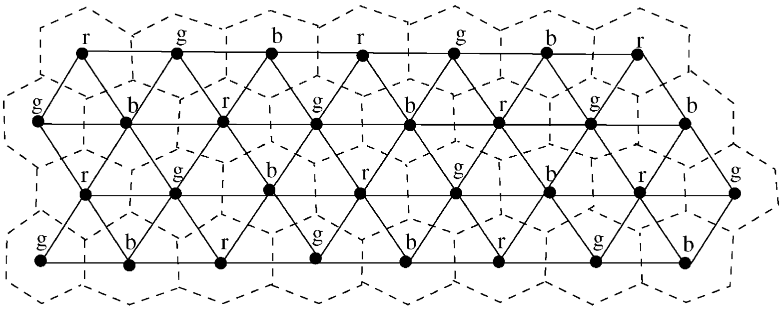


Fig. 1. A hexagon graph and a 3-coloring of the nodes of the graph. The hexagonal area around each node represents the calling area it serves.

With growing demand, it is necessary to perform this reuse as efficiently as possible, while ensuring that radio interference is at acceptable levels.

Cellular networks are generally modeled as node-weighted graphs, where the nodes represent the cells, and the edges represent the possibility of radio frequency interference. The weight on a node represents the number of calls originating in the cell represented by the node. The base station in a cell must assign frequency channels to each call originating in the cell. However, this assignment of channels must satisfy certain interference constraints. In particular, channels that are too close together may interfere with each other when they are assigned to calls that originate in the same or adjacent cells. These interference constraints can be represented by a set of integers $c_0 \geqslant c_1 \geqslant c_2, \ldots,$ where $c_i$ is the minimum separation required between two channels assigned to calls in cells that are distance $i$ apart in the network. The parameter $c_0$ which is the minimum gap between two channels assigned to the same cell is called the *co-site* constraint and the other constraints are called *inter-site* constraints. In order to optimize the use of the spectrum, the objective of the channel assignment algorithm is to minimize the *span* of the assignment, that is, the difference between the largest numbered channel used and the smallest channel used.

When $c_0 = c_1 = 1$, and $c_i = 0$ for all $i > 1$, the problem reduces to the multicoloring problem, which has been widely studied. The problem is NP-hard for many classes of graphs, including the so-called hexagon graphs, which have traditionally been used to represent cellular networks [5]. Hexagon graphs are subgraphs of the triangular lattice (see Fig. 1). They are particularly relevant to channel assignment, since they represent a regular cellular layout where the cells are hexagonal, and interference only occurs between neighboring cells. Optimal solutions for this restricted channel assignment problem are possible for some classes of graphs including complete graphs, bipartite graphs, odd cycles, and outerplanar graphs [6]. When the chromatic number of the underlying graph is $k$, an approximation algorithm with a performance ratio of $k/2$ has been shown [4]. For hexagon graphs, approximation algorithms with performance ratio of $\frac{4}{3}$ are known [5, 6]. For the case $c_0 = c_1 = c_2 = 1$, and $c_i = 0$ for all $i > 2$, an approximation algorithm with performance ratio $\frac{7}{3}$ is given in [8].

In this paper, we study the more general case $c_0 \geqslant c_1$ and $c_i = 0$ for all $i > 1$. Thus, channels assigned to the same cell must differ by at least $c_0$ and those assigned to adjacent cells must differ by at least $c_1$. For the case $c_1 = 1$, Schabanel et al. [7] give a $\frac{4}{3}$ approximation algorithm for hexagon graphs. For arbitrary $c_0$ and $c_1$, a tight bound for cliques was given by Gamst [2], and an optimal algorithm for bipartite graphs was given by Gerke [3].

We give the first known algorithms with provable performance guarantees for channel assignment with arbitrary constraints in odd cycles and hexagon graphs. The performance of our algorithms is evaluated using known lower bounds based on the maximum weight on a node, the total weight and the maximal number of nodes that can receive the same channel, or the weights on a clique and their distribution. We first show six simple algorithms for bipartite graphs, odd cycles, and 3-colorable graphs. Using these as building blocks, we derive an optimal algorithm for odd cycles when $c_0 \geqslant (2n/(n-1))c_1$, where $n$ is the length of the cycle. For the case where $c_0 < (2n/(n-1))c_1$ we give near-optimal algorithms with performance ratio at most $1 + 1/(n-1)$.

For hexagon graphs, we give approximation algorithms with performance ratio at most $\frac{4}{3}$, when $c_1 \leqslant c_0 \leqslant 2c_1$, and $9c_1/4 \leqslant c_0 < 3c_1$. For the intermediate case $2c_1 < c_0 < 9c_1/4$, the performance ratio of the algorithm is less than $\frac{4}{3} + \frac{1}{100}$. There is a straightforward optimal algorithm for hexagon graphs when $c_0 \geqslant 3c_1$. Thus for arbitrary co-site and inter-site constraints, our algorithms nearly match the performance of the best-known algorithm for the case when co-site and inter-site constraints are both exactly equal to 1.

The rest of the paper is organized as follows. We define the problem formally in Section 2. We give simple (but not necessarily optimal) algorithms for channel assignment in bipartite graphs, odd cycles, and hexagon graphs in Section 3. Near-optimal algorithms for odd cycles are given in Section 4 and approximation algorithms for hexagon graphs are then given in Section 5.

## 2. Preliminaries

For the basic definitions of graph theory we refer to [1]. A *stable set* in a graph is a set of nodes of which no pair is adjacent. A *clique* in a graph is a set of nodes of which every pair is adjacent.

A *constrained graph* $G = (V, E, c_0, c_1)$ is a graph $G = (V, E)$ and positive integer parameters $c_0$ and $c_1$ representing the reuse differences prescribed between pairs of channels assigned to the same node and adjacent nodes, respectively. A *constrained, weighted graph* is a pair $(G, w)$ where $G$ is a constrained graph and $w$ is a positive integral weight vector indexed by the nodes of $G$. The component of $w$ corresponding to node $u$ is denoted by $w(u)$ and called the *weight* of node $u$. The weight of node $u$ represents the number of calls to be serviced at node $u$. We use $w_{\max}$ to denote $\max\{w(v) \mid v \in V\}$ and $w_{\min}$ to denote the corresponding minimum weight of any node in the graph.

A *channel assignment* for a constrained, weighted graph $(G,w)$, where $G=(V,E,c_0,c_1)$ is an assignment $f$ of sets of non-negative integers (which will represent the channels) to the nodes of $G$ which satisfies the conditions:

$$|f(u)| = w(u) \qquad (u \in V),$$

$$i \in f(u) \quad \text{and} \quad j \in f(v) \Rightarrow |i-j| \geqslant c_1 \qquad ((u,v) \in E,\ u \neq v),$$

$$i,j \in f(u) \quad \text{and} \quad i \neq j \Rightarrow |i-j| \geqslant c_0 \qquad (u \in V).$$

The *span* $S(f)$ of a channel assignment $f$ of a constrained weighted graph is the difference between the lowest and the highest channel assigned by $f$, in other words, $S(f) = \max f(V) - \min f(V)$, where $f(V) = \bigcup_{u\in V} f(u)$. The span $S(G,w)$ of a constrained, weighted graph $G$ and a positive integer vector $w$ indexed by the nodes of $G$ is the minimum span of any channel assignment for $(G,w)$. We use $\chi(G,w)$ to denote the minimal *number* of channels needed for an assignment of the weighted, unconstrained graph $G$. Note that $\chi((V,E),w) = S((V,E,1,1),w)+1$, where the additive term is due to the fact that $k$ consecutive channels have a span of $k-1$.

A channel assignment $f$ is said to be *optimal* for a weighted constrained graph $G$ if $S(f) = S(G,w) + \Theta(1)$. Here we consider the span to be a function of the weights and the size of the graph, so the $\Theta(1)$ term can include terms that depend on the constraints $c_0$ and $c_1$. An approximation algorithm for channel assignment has performance ratio $k$ when the span of the assignment produced by the algorithm on $(G,w)$ is at most $kS(G,w) + \Theta(1)$.

The following lower bounds will be used to evaluate our algorithms and calculate the performance ratio. The first bound derives from the fact that any two channels on the same node must be at least $c_0$ apart. The next two bounds are based on weights and their distribution on cliques in the graph, and are derived from a bound for cliques given by Gamst [2]. The last two bounds use the fact that, because of the inter-site constraint, all nodes that receive channels from any particular channel interval of length $c_1$ must form a stable set.

**Theorem 2.1** (Known lower bounds). *Let $G=(V,E,c_0,c_1)$ be a constrained graph, and $w \in \mathbf{Z}_+^V$ a weight vector for $G$. Then*

$$S(G,w) \geqslant c_0 w_{\max} - c_0, \tag{1}$$

$$S(G,w) \geqslant \max\{c_0 w(u) + (2c_1 - c_0)w(v) \mid (u,v) \in E\} - c_0 \quad \textit{when } c_0 \leqslant 2c_1, \tag{2}$$

$$S(G,w) \geqslant \max\{c_0 w(u) + (2c_1 - c_0)(w(v)+w(t)) \mid \{u,v,t\} \textit{ a clique}\} - c_0 \quad \textit{when } c_0 \leqslant 2c_1, \tag{3}$$

$$S(G,w) \geqslant c_1 \max\{w(u)+w(v)+w(t) \mid \{u,v,t\} \textit{ a clique}\} - c_1, \tag{4}$$

$$S(G,w) \geqslant \frac{2c_1}{n-1} \sum_{v\in V} w(v) - c_1 \quad \textit{when } G \textit{ is an odd cycle of length } n. \tag{5}$$

## 3. Basic algorithms for channel assignment

In this section, we provide six simple algorithms for channel assignment in specific situations. The first two, Algorithms A and B, are optimal algorithms for bipartite graphs for the cases $c_0 \geqslant 2c_1$ and $c_1 \leqslant c_0 < 2c_1$, respectively. Note that essentially the same algorithms are given and proved to be exactly optimal (without a constant additive term) in [3]. We give them here for completeness, as we use these algorithms to prove further results in the next section. Also, our exposition is simpler as we ignore constant additive terms in our definition of optimality.

Algorithms C and D both perform channel assignment for odd cycles. Neither of these algorithms is optimal, but we use them in the next section in combination with other algorithms to obtain optimal and near-optimal bounds for odd cycles. It is easy to check that all the algorithms given in this section have linear-time implementations.

Algorithms E and F are for 3-colorable graphs. While Algorithm E's performance is always at least as good as that of Algorithm F, the latter has some room for channel borrowing. Thus, when used in combination with other algorithms, it can have an advantage over Algorithm E. We will use these two algorithms combined with modification techniques and with the algorithms for bipartite graphs, to derive near-optimal algorithms for hexagon graphs.

**Algorithm 3.1.** Algorithm A for bipartite graphs when $c_0 \geqslant 2c_1$
Let $G=(V,E,c_0,c_1)$ be a constrained bipartite graph of $n$ nodes, where $c_0 \geqslant 2c_1$ and $w$ an arbitrary weight vector. Let each node be colored red or green according to the bipartition. Red nodes use as many colors as necessary from the set $0, c_0, 2c_0, \ldots, (w_{\max}-1)c_0$. Green nodes use as many colors as necessary from the set $c_1, c_0+c_1, \ldots, (w_{\max}-1)c_0+c_1$. It is easy to see that the span of the assignment is no more than $c_0 w_{\max} - c_0 + c_1$.

**Algorithm 3.2.** Algorithm B for bipartite graphs when $c_1 \leqslant c_0 \leqslant 2c_1$
Let $G=(V,E,c_0,c_1)$ be a constrained bipartite graph of $n$ nodes, where $c_1 \leqslant c_0 \leqslant 2c_1$, and $w$ an arbitrary weight vector. Let each node be colored red or green according to the bipartition.

Given a node $v$, define $p(v) = \max\{w(u) \mid (u,v) \in E\}$. The general idea is that red nodes always get channels starting from 0 and the green nodes get channels starting from $c_1$. If a node has demand greater than any of its neighbors then it initially gets some channels that are $2c_1$ apart (in order to allow interspersing the channels of its neighbors) and the remaining distance $c_0$ apart. More precisely, we consider the following cases:

$w(v) > p(v)$ **and** $v$ **red:** Assign the channels $\{0, 2c_1, \ldots, 2p(v)c_1\} \cup \{2p(v)c_1 + c_0, \ldots, 2p(v)c_1 + (w(v) - p(v) - 1)c_0\}$. The span of the channels assigned to such a node is $p(v)(2c_1 - c_0) + w(v)c_0 - c_0$.

$w(v) > p(v)$ **and** $v$ **green:** Assign the channels $\{c_1, 3c_1, \ldots, (2p(v)-1)c_1\} \cup \{(2p(v)-1)c_1 + c_0, \ldots, (2p(v)-1)c_1 + (w(v) - p(v))c_0\}$. The span of the channels assigned to

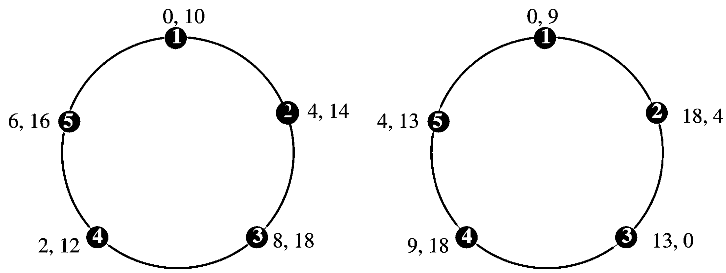


Fig. 2. Channel assignments by Algorithms C (left) and D (right) on a 5-cycle where all nodes have weight 2 and $(c_0, c_1) = (9, 4)$.

such a node is $p(v)(2c_1 - c_0) + w(v)c_0 - c_1$.

$w(v) \leqslant p(v)$ **and** $v$ **red:** Assign the channels $\{0, 2c_1, \ldots, (w(v) - 1)2c_1\}$.

$w(v) \leqslant p(v)$ **and** $v$ **green:** Assign the channels $\{c_1, 3c_1, \ldots, (2w(v) - 1)c_1\}$.

It is not difficult to see that the span of the assignment above is at most $\max_{(u,v) \in E} \{c_0 w(u) + (2c_1 - c_0)w(v)\}$ (see [3] for a complete explanation).

**Algorithm 3.3.** Algorithm C for odd cycles
Let $G = (V, E, c_0, c_1)$ be a constrained cycle of $n$ nodes, where $n > 3$ is odd, and $w$ be an arbitrary weight vector. Fix $s = \max\{c_0, c_R\}$ where $c_R = 2nc_1/(n-1)$. For convenience, the nodes of the cycle are $\{1, \ldots, n\}$, numbered in cyclic order, where node 1 is a node of maximum weight in the cycle.

The algorithm is based on an initial basic assignment of one channel per node. Additional channels are then given to each node by adding the appropriate number of multiples of $s$ to the basic assigned channel of the node. The basic assignment uses a spectrum $[0, s - 1]$. Initially, it will proceed by assigning the channel obtained by adding $c_1$ (modulo $s$) to the previously assigned channel to the next node in the cycle. At a certain point, it will switch to an alternating assignment.

More precisely, let $m > 1$ be the smallest odd integer such that $s \geqslant (2m/(m - 1))c_1$. Since $s \geqslant c_R$, a value of $m \leqslant n$ satisfying this must exist, and $m$ is well-defined. Note that the definition of $m$ implies that $s < (2(m-2)/(m-3))c_1$. Let $b$ be the basic assignment assigned as follows:

$$b(i) = \begin{cases} (i-1)c_1 \bmod s & \text{when } 1 \leqslant i \leqslant m, \\ 0 & \text{when } i > m \text{ and } i \text{ is even,} \\ (m-1)c_1 \bmod s & \text{when } i > m \text{ and } i \text{ is odd.} \end{cases}$$

To each node $i$, the algorithm assigns the channels $b(i) + js$, where $j = 0, \ldots, w(i) - 1$.

See Fig. 2 for an example of channel assignment using Algorithm C on a cycle of 5 nodes, where each node has weight 2, and $c_0 = 9$, $c_1 = 4$. It can be verified that $m = 5$ in this case.

*Correctness*: Any two channels assigned to the same node differ by at least $s$, and $s \geqslant c_0$, so the co-site constraint is satisfied.

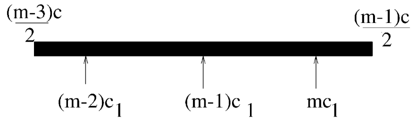


Fig. 3. Why there is no conflict between nodes $b(i)$ and $b(i+1)$ for $i \geqslant m$?

Consider two channels $\gamma_1$ and $\gamma_2$ assigned, respectively, to consecutive nodes $i$ and $i+1$. If $i<m$, then $\gamma_1 - \gamma_2 \equiv c_1 \bmod s$. If $\gamma_1$ and $\gamma_2$ differ by less than $s$, then $|\gamma_1 - \gamma_2|$ equals $c_1$ or $s - c_1$. Now, $s \geqslant c_R > 2c_1$, so $s - c_1 \geqslant c_1$, and the inter-site constraint is respected in this case.

If $i \geqslant m$, then, without loss of generality, $b(i) = 0$ and $b(i+1) = (m-1)c_1$. So $|\gamma_1 - \gamma_2| \geqslant \min\{b(i+1), s - b(i+1)\}$. From the definition of $m$, it follows that $(m-2)c_1 > ((m-3)/2)s$ and $mc_1 \leqslant ((m-1)/2)s$, so the largest multiple of $s$ smaller than $(m-1)c_1$ is $((m-3)/2)s$. Therefore, $b(i+1) = (m-1)c_1 - ((m-3)/2)s > (m-1)c_1 - (m-2)c_1 = c_1$ and $s - b(i+1) = ((m-1)/2)s - (m-1)c_1 \geqslant mc_1 - (m-1)c_1 = c_1$ (see Fig. 3). So in this case also, the inter-site constraint is respected.

*Span of the assignment*: The span of the channel assignment described here is at most $w_{\max} \max\{c_0, c_R\}$, where $c_R = 2nc_1/(n-1)$.

**Algorithm 3.4.** Algorithm D for odd cycles
Let $G = (V, E, c_0, c_1)$ be a constrained cycle of $n$ nodes, where $n > 3$ is odd, and $w$ be an arbitrary weight vector. We state the following fact about $\chi(G, w)$.

**Fact 3.5.** *For G an odd cycle of n nodes,*

$$\chi(G,w) = \max\left\{2\sum_{v \in V} w(v)/(n-1), \max\{w(u) + w(v) \mid (u,v) \in E\}\right\}.$$

This algorithm is a straightforward adaptation of the optimal algorithm for multicoloring an odd cycle (without constraints) given in [6].

Fix $s = \max\{c_0, 2c_1\}$, and $\omega = \max\{\chi(G,w), 2w_{\max}\}$. We use the spectrum $[0, \ldots, s\lceil \omega/2 \rceil]$, and describe a fixed sequence of channels to be used from this spectrum in that order by the multicoloring algorithm. The sequence is

$$\left(0, s, 2s, \ldots, \left(\left\lceil \frac{\omega}{2} \right\rceil - 1\right) s, c_1, s + c_1, 2s + c_1, \ldots, \left(\left\lfloor \frac{\omega}{2} \right\rfloor - 1\right) s + c_1\right).$$

It is straightforward to verify that there are exactly $\omega$ channels in this sequence. We now proceed as for multicoloring, using the sequence given here to assign channels rather than a continuous part of the spectrum.

Precisely, let $k$ be the smallest integer such that $\sum_{i=1}^{2k+1} w(i) \leqslant k\omega$. Nodes 1 through $2k$ are assigned *contiguous* channels in a cyclic manner from the spectrum given. Specifically, for $1 \leqslant j \leqslant 2k$, node $j$ is assigned the $\ell$th through $m$th channel of the spectrum, where $\ell$ and $m$ are between 0 and $\omega - 1$, and $\ell \equiv 1 + \sum_{i=1}^{j-1} w(i) \bmod \omega$

and $m \equiv \sum_{i=1}^{j} w(i) \bmod \omega$. This assignment is done cyclically, so if $\ell > m$, then the channels "wrap around" from the $\ell$th channel to the end of the spectrum, and back from the beginning of the spectrum to the $m$th channel. The assignment for nodes $2k+1$ through $n$ is based on their parity; for $2k+1 \leqslant i \leqslant n$, node $i$ is assigned the first $w(i)$ channels of the spectrum if $i$ is even, or the last $w(i)$ channels if $i$ is odd.

See Fig. 2 for an example of channel assignment using Algorithm D on a cycle of 5 nodes, where each node has weight 2, and $c_0 = 9$, $c_1 = 4$. In this case, $\omega = \chi(G, w) = 5$. The reader can verify that the sequence of channels to be used is $(0, 9, 18, 4, 13)$.

*Correctness*: First, we have to show that an integer $k$ that satisfies the condition always exists. Suppose instead that $\sum_{i=1}^{2k+1} w(i) > k\omega$ for all $k$. Then, for $k = (n - 1)/2$ in particular, we have that $\sum_{i=1}^{n} w(i) > (n-1)/2\omega \geqslant \chi(G, w)$, a contradiction with Fact 3.5.

It is easy to see that any two consecutive members of the sequence have channels that are at least $c_0$ apart. Further, since $w_{\max} \leqslant \omega/2$ and any subsequence of length at most $\omega/2$ from the above sequence has the property that any pair of colors is at least $c_0$ apart, there can never be conflict between two colors assigned to the same node due to "wrapping around" the spectrum. Thus the co-site constraint is always respected.

Since any pair of channels in the sequence has a difference of at least $c_1$, the inter-site constraint can only be violated if neighboring nodes are assigned the same channels. For nodes 1 through $2k$ and $2k+1$ through $n$ this does not happen because the spectrum contains more than $w(i)+w(i+1)$ channels for any $i$. There is also no conflict between node $2k$ and $2k+1$: the last channel from the spectrum assigned to node $2k$ is channel number $\sum_{i=1}^{2k} w(i) - (k-1)\omega$. Since $2k+1$ is odd, the lowest channel assigned to node $2k+1$ is channel number$(\omega - w(2k+1))$. The rest follows from the definition of $k$. Finally, there is no conflict between nodes $n$ and 1 because $n$ is odd, so node $n$ receives the last channels from the spectrum, whereas node 1 is assigned the first channels.

*Span of the assignment*: The span of the assignment is no more than $\max\{c_0, 2c_1\} \cdot \max\{w_{\max}, \lceil \chi(G, w)/2 \rceil\}$.

**Algorithm 3.6.** Algorithm E for 3-colorable graphs
Let $G = (V, E, c_0, c_1)$ be a 3-colorable graph, and $w$ be an arbitrary weight vector. Fix $s = \max\{3c_1, c_0\}$.

This algorithm uses a coloring of the nodes of the graph with colors red, blue and green. It assigns at most $w_{\max}$ channels to each node. For $j = 0, \ldots, w_{\max} - 1$, the channels $js$ are reserved for red nodes, the channels $js + c_1$ are reserved for green nodes, and the channels $js + 2c_1$ are reserved for blue nodes. Each node $v$ is assigned $w(v)$ channels from its own reserved sets of channels.

*Correctness*: Since any two assigned channels have a separation of $c_1$ or $s - c_1 \geqslant c_1$, the inter-site constraints are respected. Any two channels assigned to the same node have a separation of $s \geqslant c_0$, so the co-site constraint is respected as well.

*Span of the assignment*: This algorithm produces an assignment of span at most $sw_{\max} - c_1 = \max\{3c_1, c_0\}w_{\max} - c_1$.

**Algorithm 3.7.** Algorithm F for 3-colorable graphs
Let $G=(V,E,c_0,c_1)$ be a 3-colorable graph, and $w$ be an arbitrary weight vector. Fix $s=\max\{c_1,c_0/2\}$ and $T\geqslant 3w_{\max}$.

We use a spectrum of $T$ channels, with consecutive channels separated by $s$, where channels reserved for different colors are interspersed. (This alternation of channels was first used in [7].) We assume for ease of explanation that $T$ is a multiple of 6. Precisely, the red channels consist of a first set $R_1=[0,2s,\ldots,(T/3-2)s]$ and a second set $R_2=[(T/3+1)s+c_0,(T/3+3)s+c_0,\ldots,(2T/3-1)s+c_0]$. The blue channels consist of first set $B_1=[(T/3)s+c_0,(T/3+2)s+c_0,\ldots,(2T/3-2)s+c_0]$ and second set $B_2=[(2T/3+1)s+2c_0,(2T/3+3)s+2c_0,\ldots,(T-1)s+2c_0]$, and the green channels consist of first set $G_1=[(2T/3)s+2c_0,(2T/3+2)s+2c_0,\ldots,(T-2)s+2c_0]$ and second set $G_2=[s,3s,\ldots,(T/3-1)s]$. Thus, we can think of the spectrum as being divided into three parts, each containing $T/3$ channels, with a separation of $s$ between consecutive channels. The first part of the spectrum consists of alternating channels from $R_1$ and $G_2$, the second part has alternating channels from $B_1$ and $R_2$, and the third part has alternating channels from $G_1$ and $B_2$.

Note that there are extra gaps of $c_0$ between these three parts of the spectrum. This gap is not needed for the correctness of Algorithm F when used exactly as described here. However, we have included this gap to anticipate the use of this algorithm in combination with subsequent borrowing and combination phases, as described in the next section. Note that the extra gaps only add a constant of $2c_0$ to the span of the assignment.

Each node $v$ is assigned $w(v)$ channels from those of its color class, where the first set is exhausted before starting on the second set, and lowest numbered channels are always used first within each set.

*Correctness*: Any two assigned channels have separation at least $s\geqslant c_1$, so the intersite constraint is satisfied. Any two channels within the two channel sets of a color are separated by at least $2s\geqslant c_0$, so the co-site constraint is satisfied.

*Span of the assignment*: The span equals $sT+2c_0=\max\{c_1,c_0/2\}T+2c_0$, where $T$ is at least $3w_{\max}$.

## 4. Near-optimal algorithms for odd cycles

In this section, we will describe how variations and combinations of the algorithms described in Section 3 can be used to derive optimal and near-optimal algorithms for channel assignment in odd cycles. As stated earlier, optimal algorithms for channel assignment in even cycles were given in [3]. We deal with the cases $c_0\geqslant c_R$, $2c_1\leqslant c_0<c_R$, and $c_0<2c_1$ separately. In the first case, we give an optimal algorithm, and in the second and third cases, approximation algorithms with performance ratio $1+1/(4n-3)$ or $1+1/(n-1)$, respectively ($n$ is the length of the cycle).

**Theorem 4.1.** *For any $c_0 \geqslant c_R = 2nc_1/(n-1)$, $G=(V,E,c_0,c_1)$ a constrained odd cycle of length n, and w an arbitrary weight vector, there is a linear time optimal algorithm for channel assignment in $(G,w)$.*

**Proof.** Since $c_0 \geqslant c_R$, Algorithm C gives an assignment using $c_0 w_{\max}$ channels, and it follows from lower bound (1) of Theorem 2.1 that it is optimal. □

**Theorem 4.2.** *For any $2c_1 \leqslant c_0 < c_R = 2nc_1/(n-1)$, $G=(V,E,c_0,c_1)$ a constrained odd cycle of length n, and w an arbitrary weight vector, there is a linear time approximation algorithm for channel assignment in $(G,w)$ that has performance ratio $1+1/(4n-3)$ (or, in an alternative formulation, of performance ratio $1+1/(4n)+\mathcal{O}(1/n^2)$), where n is the number of nodes in the cycle.*

**Proof.** Compute $\delta = \sum_{v\in V} w(v) - (n-1)w_{\max}$. If $\delta \leqslant 0$, it follows as a consequence of Fact 3.5 that $\chi(G,w) \leqslant 2w_{\max}$. Therefore, we use Algorithm D with spectrum $[0, c_0 w_{\max}]$. The span is at most $c_0 w_{\max}$, which is within a constant of lower bound (1) of Theorem 2.1, so the assignment is optimal.

If instead $\delta > 0$, we combine Algorithm C with either Algorithm A or D to derive an assignment. Denote by $f_1$ the assignment computed by Algorithm C for $(G,w')$ where $w'(v) = \min\{w(v),\delta\}$. This assignment has span at most $c_R\delta$.

We consider the remaining weight $\bar{w}$ after this assignment. Clearly, $\bar{w}_{\max} = w_{\max} - \delta$. We will denote by $f_2$ the assignment for $(G,\bar{w})$, and compute it in two different ways depending on a key property of $\bar{w}$. If there is a node $v$ with $\bar{w}(v)=0$ at this stage, we have a bipartite graph left. Then $f_2$ is the assignment computed by Algorithm A for $(G,\bar{w})$. This assignment has a span of at most $c_0\bar{w}_{\max}$.

If all nodes have non-zero weight, then $\sum_{v\in V} \bar{w}(v) = \sum_{v\in V} w(v) - n\delta$. We claim that $\bar{w}_{\max} = \sum_{v\in V} \bar{w}(v)/(n-1) = \chi(G,\bar{w})/2$, as shown below:

$$\sum_{v\in V} \bar{w}(v) = \sum_{v\in V} w(v) - n\delta$$

$$= (n-1)\left(nw_{\max} - \sum_{v\in V} w(v)\right)$$

$$= (n-1)\bar{w}_{\max}.$$

Thus, we can use Algorithm D (using $\omega = 2\bar{w}_{\max}$) to compute $f_2$, the assignment for $(G,\bar{w})$. Note that this also has span $c_0\omega/2 = c_0\bar{w}_{\max}$. Thus in either case, $f_2$ has span at most $c_0\bar{w}_{\max}$.

The final assignment computed for $(G,w)$ is as follows: for any node $v$, $f(v) = f_1(v) \cup \{y + c_R\delta + c_0 \mid y \in f_2(v)\}$. This is a valid assignment, since every channel derived from the assignment $f_2$ has a difference of at least $c_0$ from the highest channel assigned by $f_1$. The span of the assignment is $c_R\delta + c_0\bar{w}_{\max} + c_0$. Since $\bar{w}_{\max} = w_{\max} - \delta$, the span is at most $(c_R - c_0)\delta + c_0 w_{\max} + c_0$. Now

$$(c_R - c_0)\left(\sum_{v\in V} w(v) - (n-1)w_{\max}\right) + c_0 w_{\max}$$

$$= \left(\frac{2nc_1}{n-1} - c_0\right)\sum_{v\in V} w(v) + (nc_0 - 2nc_1)w_{\max}$$

$$= \left(n - c_0\frac{n-1}{2c_1}\right)\frac{c_1}{n-1}\sum_{v\in V} w(v) + \left(\frac{n(c_0-2c_1)}{c_0}\right)c_0 w_{\max}.$$

By the lower bounds (1) and (5) of Theorem 2.1, this implies that the performance ratio of this algorithm is at most $n-(c_0/2c_1)(n-1)+(1-2c_1/c_0)n+\Theta(1)$. Now, the function $f(y)=n-y(n-1)+(1-1/y)n$ achieves its maximum for positive values of $y$ at $y=\sqrt{n/(n-1)}$, and hence never exceeds $f(\sqrt{n/(n-1)})=1+1/4n+\mathcal{O}(1/n^2)$. Also, for $n\geqslant 3$ we have that $f(\sqrt{n/(n-1)}) \leqslant 1+1/(4n-3)$. Thus the performance ratio of the algorithm is as claimed. □

**Theorem 4.3.** *For any* $c_0<2c_1$, $G=(V,E,c_0,c_1)$ *a constrained odd cycle of length n, and w an arbitrary weight vector, there is a linear time approximation algorithm for channel assignment in* $(G,w)$ *that has performance ratio* $1+1/(n-1)$ *where n is the number of nodes in the cycle.*

**Proof.** In this case, we combine Algorithms C and B in the following manner. Denote by $f_1$ the assignment computed by Algorithm C for $(G,w')$ where $w'(v)=w_{\min}$ for every node $v$. Denote by $f_2$ the assignment computed by Algorithm B for $(G,w'')$ where $w''(v)=w(v)-w_{\min}$.

Let $L=\max\{c_0 w(u)+(2c_1-c_0)w(v)\,|\,(u,v)\in E\}$, the lower bound (2) given by Theorem 2.1 for $S(G,w)$. Algorithm $B$ for $(G,w'')$ produces an assignment of span $max\{c_0 w''(u)+(2c_1-c_0)w''(v)\,|\,(u,v)\in E\}$, which equals $L-2c_1 w_{\min}$.

The final assignment for $(G,w)$ is as follows: for any node $v$, $f(v)=f_1(v)\cup\{y+c_R w_{\min}+c_0\,|\,y\in f_2(v)\}$. The span of this assignment is $c_R w_{\min}+L-2c_1 w_{\min}+c_0=L+2c_1 w_{\min}/(n-1)+c_0\leqslant L(1+1/(n-1))+\Theta(1)$. It follows from lower bound (2) that the algorithm has performance ratio $1+1/(n-1)$ as claimed. □

The results of this section are summarized in Table 1.

Table 1
Upper bounds on performance ratio on odd cycles for different values of $c_0$ and $c_1$

| | $c_1\leqslant c_0<2c_1$ | $2c_1\leqslant c_0<2nc_1/(n-1)$ | $c_0\geqslant 2nc_1/(n-1)$ |
|---|---|---|---|
| Perf. ratio | $1+1/(n-1)$ | $1+1/(4n-3)$ | 1 |

## 5. Approximation algorithms for hexagon graphs

In this section, we describe approximation algorithms for channel assignment with constraints in hexagon graphs. The algorithms we describe use a standard 3-coloring of hexagon graphs, which gives a partition of the nodes into red, blue, and green nodes (see Fig. 1). The first two theorems are based on Algorithm E and give results for the cases $c_0 \geqslant 3c_1$, where the algorithm is optimal, and for $c_1 \leqslant c_0 < 3c_1$. The last two theorems use a combination of the algorithms given in Section 3 with additional modifications, and deal with the cases, where $2c_1 \leqslant c_0 \leqslant (9/4)c_1$ and $c_1 \leqslant c_0 \leqslant 2c_1$, respectively.

**Theorem 5.1.** *For any $c_0 \geqslant 3c_1$, $G=(V,E,c_0,c_1)$ a constrained hexagon graph, and $w$ an arbitrary weight vector, there is an optimal linear time approximation algorithm for channel assignment in $(G,w)$.*

**Proof.** Since $c_0 \geqslant 3c_1$, Algorithm E gives an assignment of span at most $c_0 w_{\max}$, and it follows from lower bound (1) of Theorem 2.1 that this is an optimal assignment. □

**Theorem 5.2.** *For any $c_1 \leqslant c_0 < 3c_1$, $G=(V,E,c_0,c_1)$ a constrained hexagon graph, and $w$ an arbitrary weight vector, there is a linear time approximation algorithm for channel assignment in $(G,w)$ that has performance ratio $3c_1/c_0$.*

**Proof.** Since $3c_1 \geqslant c_0$, Algorithm E gives an assignment of span at most $3c_1 w_{\max}$. By lower bound (1), $S(G,w) \geqslant c_0 w_{\max} - c_0$, so this span is at most $(3c_1/c_0)S(G,w) + \Theta(1)$, as claimed. □

Note that when $c_0 \geqslant (\frac{9}{4})c_1$, the performance ratio of the above algorithm is at most $\frac{4}{3}$. In the following two theorems, we give algorithms that improve on the performance ratio given in Theorem 5.2 for values of $c_0 < (\frac{9}{4})c_1$.

The two remaining algorithms in this section use the same type of strategy. First, each node is assigned enough channels from those assigned to its color class to guarantee that there are no triangles left in the graph. Next, each node *borrows* any available channels of a designated borrowing color. The resulting graph is then shown to be a bipartite graph, for which an optimal channel assignment is found. (This general approach was first used for multicoloring of hexagon graphs in [5, 6].) The algorithms differ in the initial separation of channels into different color classes. The algorithm of Theorem 5.4 uses an additional technique of *squeezing* channels when possible. Both borrowing and squeezing make use of already assigned parts of the spectrum, and thus do not add to the total span of the assignment.

In the following, let $D$ represent the maximum weight of any maximal clique (edge or triangle) in the graph. It follows from lower bound (4) of Theorem 2.1 that $S(G,w) \geqslant c_1 D - c_1$. For ease of explanation, we assume that $D$ is a multiple

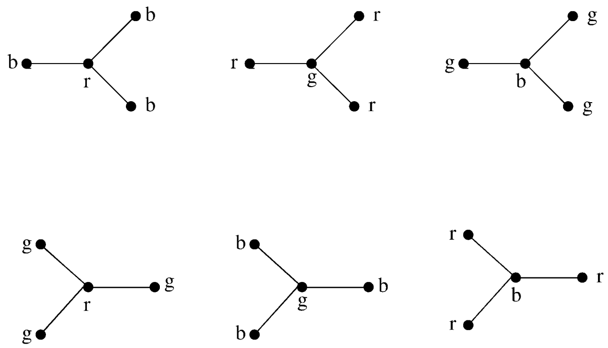


Fig. 4. The orientations of corner nodes. After the borrowing phase, corner nodes of the type in the second row drop out.

of 6; it is easy to verify that a more careful consideration would only add a constant additive term to the span of the assignment.

We define a *corner* node to be a node which has at least two neighbors of the same color class, and no neighbors of the remaining third color class. Based on the color of the corner node itself and the color of its 2 or 3 neighbors, we can identify six types of corner nodes (see Fig. 4). Furthermore, we fix the designated borrowing colors as follows. For red, the borrowing color is blue, for blue it is green, and for green, it is red.

**Theorem 5.3.** *For any* $2c_1 < c_0 \leqslant (\frac{9}{4})c_1$, $G = (V, E, c_0, c_1)$ *a constrained hexagon graph, and* $w$ *an arbitrary weight vector, there is a linear time approximation algorithm for channel assignment in* $(G, w)$ *that has performance ratio* $1 + 3(c_0 - 2c_1)/c_0 + (9c_1 - 4c_0)/3c_1$.

**Proof.** The algorithm proceeds in four phases. The first two phases assign a total of $D$ channels, partly according to Algorithm E (Phase 1) and partly according to Algorithm F (Phase 2). The next phase is a borrowing phase in which nodes borrow any available channels of the borrowing color, i.e., any channels that are unused by all of its neighbors of the borrowing color. The resulting graph is shown to be a bipartite graph, for which an assignment is found in Phase 4 using Algorithm A.

*Phase* 1: If $D > 2w_{\max}$, use Algorithm E on $(G, w')$ where $w'(v) = \min\{w(v), D - 2w_{\max}\}$. In this case the parameter $s$ prescribing the separation between consecutive channels at a node equals $3c_1$, so the span of the assignment is at most $(D - 2w_{\max})3c_1 - c_1$. If $D \leqslant 2w_{\max}$, skip this phase, and take $w'(v) = 0$ for all $v$. The span needed for this phase is no more than $\max\{0, D - 2w_{\max}\}3c_1$.

*Phase* 2: Let $T = \min\{2w_{\max}, 6w_{\max} - 2D\}$. Use Algorithm F on $(G, w'')$, where $w''(v) = \min\{w(v) - w'(v), T/3\}$, taking $T$ as defined.

Note that in this case the parameter $s$ representing the separation gap between consecutive channels of the spectrum equals $c_0/2$, so the span of the assignment is $\min\{2w_{\max}, (6w_{\max} - 2D)\}c_0/2 + 2c_0$.

We claim that after this phase, there are no triangles left in the resulting graph. This is because $\max\{0,(D-2w_{\max})\}+\min\{2w_{\max}/3,2w_{\max}-2D/3\}\geqslant D/3$, so in Phases 1 and 2, at least $\min\{w(v),D/3\}$ channels are assigned to each node $v$. Since in any triangle, the sum of the weights is at most $D$, the claim follows.

*Phase* 3: Any node which has still unfulfilled demand tries to borrow channels assigned in Phase 2 from its neighbors of the borrowing color. Precisely, let $v$ be a node with $w(v)>w'(v)+w''(v)$, and let $w_{\mathrm{B}}(v)$ be the maximum number of channels used during Phase 2 by any neighbor of $v$ of its borrowing color, so $w_{\mathrm{B}}(v)$ is the maximum of $w''(u)$ over all three neighbors $u$ of $v$ which are of $v$'s borrowing color. Then we assign an additional $\min\{w(v)-w'(v)-w''(v),T/3-w_{\mathrm{B}}(v),T/6\}$ channels to $v$ from the second channel set of the borrowing color of $v$, starting from the highest channels in the set. ($T=\min\{2w_{\max},6w_{\max}-2D\}$, as defined in Phase 2.)

*Correctness of this phase*: Since any two channels assigned to $v$ in this phase are at least $c_0$ apart, and are not from the two of the three parts of the spectrum where channels were assigned to $v$ in earlier phases, the co-site constraint is respected. Since only channels unused by any neighbor are used, and all channels are at least $c_1$ apart, the inter-site constraint is respected as well. Thus there are no conflicts caused by the assignment in this phase.

We claim that the resulting graph is bipartite. We show this by arguing that, at this point, no corner nodes that have neighbors of the non-borrowing color are left. The argument is only given for red nodes but by symmetry it translates easily into an analogous argument for blue or green nodes. Recall that for a red node, the borrowing color is blue.

Let $v$ be a red corner node with at least two green neighbors, that all still survive in the graph after Phases 1 and 2. Let $\alpha=w(v)-(w'(v)+w''(v))$, and assume that $\alpha>0$. It suffices to show that $v$ can borrow $\alpha$ channels in Phase 3. In other words, we need to show that $\alpha\leqslant\min\{T/3-w_{\mathrm{B}}(v),T/6\}$.

Node $v$ has at least two green neighbors of initial weight greater than $D/3$, so any blue neighbor of $v$ is contained in a triangle with $v$ and a green node of weight greater than $D/3$. It follows that any blue neighbor $u$ of $v$ has initial weight $w(u)\leqslant D-D/3-w(v)$. For any blue neighbor $u$ with $w''(u)>0$, $w''(u)=w(u)-w'(u)\leqslant 2D/3-w(v)-\max\{0,(D-2w_{\max})\}$. If $D\leqslant 2w_{\max}$, then $T=2w_{\max}$ and $\alpha=w(v)-T/3$, so $w''(u)\leqslant 2D/3-w(v)\leqslant 4w_{\max}/3-w(v)=T/3-\alpha$. Also, $\alpha\leqslant w_{\max}-T/3=T/6$.

If instead $D>2w_{\max}$, then $T=6w_{\max}-2D$ and $\alpha=w(v)-D/3$, so $w''(u)\leqslant 2D/3-w(v)-(D-2w_{\max})=T/3-\alpha$. Also, since $w(v)-(D-2w_{\max}+T/3)\leqslant 3w_{\max}-D-T/3=T/2-T/3$, we have that $\alpha\leqslant T/6$.

Therefore, in both cases $w_{\mathrm{B}}(v)\leqslant T/3-\alpha$ and $\alpha\leqslant\min\{T/3-w_{\mathrm{B}}(v),T/6\}$ as required. Since node $v$ could borrow at least $\alpha$ colors in Phase 2, its demand is completely fulfilled after the first three phases.

An analogous argument for the blue and green corner nodes shows that all corner nodes with at least two neighbors from their non-borrowing color receive enough channels and drop out before this phase. Since all the remaining corner nodes have the same "orientation" (see Fig. 4), there can be no cycles in the graph.

*Phase* 4: Let $\bar{w}$ denote the weight left on the nodes after the assignments of the first three phases. Use Algorithm A to find an assignment for $(G,\bar{w})$, which has a span of $c_0\bar{w}_{\max}$.

The assignments of all four phases are then combined without causing conflicts, in the same way as in the theorems for odd cycles. Note that the assignment of Phase 3 does not add to the span. If $D>2w_{\max}$, then $T=6w_{\max}-2D$ and $\bar{w}_{\max}\leqslant w_{\max}-D/3$. So the final assignment has span $(3D-6w_{\max})c_1+(6w_{\max}-2D)c_0/2+c_0(w_{\max}-D/3)+\Theta(1)=(4c_0-6c_1)w_{\max}+(3c_1-4c_0/3)D+\Theta(1)$.

If $D\leqslant 2w_{\max}$, then Phase 1 is skipped, and in Phase 2 each node of maximum weight receives $T/3=2w_{\max}/3$ channels. So in this case, $\bar{w}_{\max}\leqslant w_{\max}-T/3=w_{\max}/3$, and the final assignment has span at most $(2w_{\max})c_0/2+c_0(w_{\max}/3)+\Theta(1)=(4/3)c_0w_{\max}+\Theta(1)$. The result then follows from the lower bounds (1) and (4) of Theorem 2.1, and the fact that $(4c_0-6c_1)c_0+(3c_1-4c_0/3)/c_1\geqslant\frac{4}{3}$ when $2c_1\leqslant c_0\leqslant(\frac{9}{4})c_1$. □

The above theorem yields an algorithm with performance ratio that is always less than $\frac{4}{3}+\frac{1}{100}$. In particular, the maximum value of the performance ratio is reached when $c_0/c_1=3/\sqrt{2}$. When $c_0=2c_1$ or $c_0=9c_1/4$, the performance ratio is exactly $\frac{4}{3}$.

**Theorem 5.4.** *For any $c_0\leqslant 2c_1$, $G=(V,E,c_0,c_1)$ a constrained hexagon graph, and $w$ an arbitrary weight vector, there is a linear time approximation algorithm for channel assignment in $(G,w)$ that has performance ratio $\frac{4}{3}$.*

**Proof.** The algorithm consists of four phases. First, Algorithm F is used with a spectrum of $D$ channels. In Phase 2, nodes try to borrow available channels of the borrowing color. In the next phase, some of the channels assigned are “squeezed” closer together where possible to accommodate more channels. The remaining graph is then bipartite, and Algorithm B is used. The precise description of the phases follows. In the following, let

$$L=\max\{c_0w(u)+(2c_1-c_0)(w(v)+w(r))\mid\{u,v,r\}\text{ a triangle}\}$$

and let $T$ be the smallest multiple of 6 larger than $\max\{L,Dc_1\}/c_1$. It follows from lower bounds (3) and (4) of Theorem 2.1 that $Tc_1-\Theta(1)$ is a lower bound for the span of any assignment.

*Phase* 1: Use Algorithm F on $(G,w')$, where $w'(v)=\min\{w(v),T/3\}$ and $T$ is defined above. In this case $s$, the separation between channels, equals $c_1$, so the span of the assignment is $Tc_1$.

*Phase* 2: Any node $v$ of weight greater than $T/3$ borrows $\min\{w(v)-T/3,T/3-w_{\mathrm{B}}(v),T/6\}$ channels from the borrowing color set where $w_{\mathrm{B}}(v)$ is the maximum weight of any neighbor of $v$ of the borrowing color of $v$. The channels assigned in this phase are taken only from the second channels set of the borrowing color, and start with the highest channels. The correctness of this phase follows from the fact that only channels unused by neighbors are used, and all channels are at least $c_0$ apart from each other, and at least $c_1$ apart from any neighbor.

*Phase* 3: Any remaining node for which all neighbors of its borrowing color have used less than $T/6$ channels from their first color set, will squeeze their assigned channels from their second set as much as possible. Precisely, let $v$ be a red node with weight more than $T/3$, such that each of its blue neighbors has weight at most $T/6$. Then replace the last $T/6 - w_B(v)$ channels from $R_2$ by $\min\{w(v) - T/3 - w_B(v), (2c_1/c_0)(T/6 - w_B(v))\}$ channels with separation $c_0$ which fill the part (of length $2(T/6 - w_B(v))c_1$) of the spectrum occupied by the last $T/6 - w_B(v)$ channels of $R_2$.

For example, let $T = 24$, $c_0 = 3$, and $c_1 = 2$. Suppose $v$ is a red corner node with at least two green neighbors, where $w(v) = 13$ and let $w_B(v) = 1$. In Phase 1, $v$ received the channels $21, 25, 29, 33$ from the set $R_2$, whereas at least one blue neighbor of $v$ received the channel 19 from $B_1$ and no other channels from $B_1$ or $B_2$ were used by any neighbor of $v$. Then in Phase 2, $v$ borrows all four blue channels in $B_2$, and in Phase 3, squeezes the part of the spectrum $[21, 33]$ of $R_2$ to get five channels. In particular, it uses the channels $21, 24, 27, 30, 33$ instead of the four channels mentioned above. The reader can verify that in this example, co-site and inter-site constraints are respected.

*Correctness of this phase*: Since we only squeeze that part of $R_2$ where the interspersed channels of $B_1$ are not being used, the inter-site constraint is not violated. Since channels with separation $c_0$ are used, the co-site constraint is respected. Thus, there are no conflicts caused by the assignment in this phase.

Furthermore, we claim that the resulting graph is bipartite, by showing that, at this point, no corner nodes with non-zero weight are left that have two or more neighbors of non-borrowing color. Consider a red corner node $v$ and suppose that it still has two or more green neighbors. Then node $v$ had initial weight at most $2D/3 \leqslant 2T/3$, since it has green neighbors of initial weight at least $T/3 \geqslant D/3$. Also, since every blue neighbor of $v$ is contained in a triangle with $v$ and such a green neighbor, $w_B(v) \leqslant D - w(v) - T/3 \leqslant 2T/3 - w(v)$.

Suppose, first that $w(v) = T/3 + \alpha$ with $0 < \alpha \leqslant T/6$. We then show that $v$ can borrow $\alpha$ channels in Phase 2. In this case, $w_B(v) \leqslant 2T/3 - w(v) = T/3 - \alpha$, so $\alpha \leqslant \min\{T/3 - w_B(v), T/6\}$, as required. So no weight is left on $v$ after Phase 2.

Suppose instead that $w(v) = T/2 + \alpha$, where $0 < \alpha \leqslant T/6$, and $w_B(v) = T/6 - \beta$, where $0 \leqslant \beta \leqslant T/6$. Then $\min\{w(v) - T/3, T/3 - w_B(v), T/6\} = T/6$, so in Phase 2, node $v$ borrows $T/6$ channels from $B_2$. In Phase 3, node $v$ can replace $\beta$ of its channels from $R_2$ by $\min\{\alpha + \beta, (2c_1/c_0)\beta\}$ channels with separation $c_0$. But

$$\begin{aligned} Tc_1 \geqslant L &\geqslant c_0 w(v) + (2c_1 - c_0)(w_B + T/3) \\ &= c_0(T/2 + \alpha) + (2c_1 - c_0)(T/2 - \beta) \\ &= Tc_1 + c_0\alpha - (2c_1 - c_0)\beta, \end{aligned}$$

so $\alpha \leqslant ((2c_1 - c_0)/c_0)\beta$, and $\alpha + \beta \leqslant (2c_1/c_0)\beta$, so in Phase 3, $\beta$ channels previously assigned to $v$ are replaced by $\alpha + \beta$ new channels. So $v$ receives $T/3$ channels in Phase 1, $T/6$ channels in Phase 2, and replaces $\beta$ channels by $\alpha + \beta$ channels in

Phase 3. Therefore, the weight left on $v$ after this phase is $w(v) - (T/3 + T/6 - \beta + (\alpha + \beta)) = 0$.

An analogous argument for the blue and green corner nodes shows that all corner nodes specified in this phase receive enough channels and drop out before this phase. Since all the remaining corner nodes have the same "orientation" (see Fig. 1), there can be no cycles in the graph.

*Phase* 4: Let $\bar{w}$ be the weight vector remaining after Phase 3. We use Algorithm B to find an assignment for $(G, \bar{w})$, which has a span of $L' = \max\{c_0\bar{w}(u) + (2c_1 - c_0)\bar{w}(v) \mid (u, v) \in E\}$.

The bound $L'$ may be achieved at either isolated nodes or edges, and we show that in each case, $L' \leqslant Tc_1/3$. Let $v$ be a red node which has become isolated after Phases 1–3. Note that in the following case analysis, $w(v)$ refers to the *initial* weight of the node, while $\bar{w}(v)$ denotes the weight left on $v$ after Phases 1–3. Recall that $w_B(v)$ is the maximum weight on any blue neighbor of $v$.

$w_B(v) \geqslant T/3$: In this case, $v$ could not borrow or squeeze, so the remaining weight on node $v$ is $\bar{w}(v) = w(v) - T/3$. Since

$$\begin{aligned} Tc_1 \geqslant L &\geqslant c_0 w(v) + (2c_1 - c_0) w_B(v) \\ &\geqslant c_0\bar{w}(v) + c_0 T/3 + (2c_1 - c_0)T/3, \end{aligned}$$

we have that $c_0\bar{w}(v) \leqslant Tc_1/3$.

$T/6 \leqslant w_B(v) < T/3$: In this case, node $v$ could borrow $T/3 - w_B(v)$ channels in the second phase of Algorithm F, so $\bar{w}(v) = w(v) - T/3 - (T/3 - w_B(v))$. Since

$$\begin{aligned} Tc_1 \geqslant L &\geqslant c_0 w(v) + (2c_1 - c_0) w_B(v) \\ &= c_0(\bar{w}(v) + 2T/3 - w_B(v)) + (2c_1 - c_0) w_B(v) \\ &= c_0\bar{w}(v) + c_0(2T/3) - (c_0 - c_1)(2w_B(v)) \\ &\geqslant c_0\bar{w}(v) + c_0(2T/3) - (c_0 - c_1)(2T/3), \end{aligned}$$

we have that $c_0\bar{w}(v) \leqslant Tc_1/3$.

$w_B(v) < T/6$: In this case, node $v$ could borrow $T/6$ channels in the second phase of Algorithm F, and squeeze a portion of its channels in Phase 2. In Phase 2, $\beta = T/6 - w_B$ channels from $R_2$ with separation $2c_1$ are replaced by $(2c_1/c_0)\beta$ channels with separation $c_0$. Therefore, $\bar{w}(v) = w(v) - (T/3 + T/6 + ((2c_1 - c_0)/c_0)\beta)$, and

$$\begin{aligned} Tc_1 \geqslant L &\geqslant c_0 w(v) + (2c_1 - c_0) w_B(v) \\ &= c_0\left(\bar{w}(v) + T/2 + \frac{2c_1 - c_0}{c_0}\beta\right) + (2c_1 - c_0)(T/6 - \beta) \\ &= c_0\bar{w}(v) + c_0(T/2) + (2c_1 - c_0)T/6 \\ &= c_0\bar{w}(v) + (c_0 + c_1)T/3 \\ &\geqslant c_0\bar{w}(v) + (2T/3)c_1, \end{aligned}$$

so $c_0\bar{w}(v) \leqslant Tc_1/3$.

Table 2
Upper bounds on performance ratio on hexagon graphs for different values of $c_0$ and $c_1$

| | $c_1 \leqslant c_0 \leqslant 2c_1$ | $2c_1 < c_0 < 9c_1/4$ | $9c_1/4 \leqslant c_0 < 3c_1$ | $c_0 \geqslant 3c_1$ |
|---|---|---|---|---|
| Perf. ratio | 4/3 | 4/3 + 1/100 | 4/3 | 1 |

It remains to show the upper bound on $L'$ when it is achieved at an edge. Let $u$ and $v$ be the nodes achieving the maximum for $L'$. Then

$$\begin{aligned} L' &= c_0\bar{w}(u) + (2c_1 - c_0)\bar{w}(v) \\ &\leqslant c_0(w(u) - T/3) + (2c_1 - c_0)(w(v) - T/3) \\ &\leqslant L - 2Tc_1/3 \\ &\leqslant Tc_1/3. \end{aligned}$$

The assignments of different phases are then combined without causing conflicts, in the same way as in the previous theorems, to give a final assignment of span at most $(\frac{4}{3})Tc_1 + \Theta(1)$. $c_1 + \Theta(1)$, From the definition of $T$, we have that $Tc_1 - \Theta(1)$ is a lower bound, which gives the required performance ratio of $\frac{4}{3}$. □

Table 2 summarizes the results of this section. Note that the given values are upper bounds; as $c_0$ approaches $3c_1$, the performance ratio approaches 1, and similarly, the performance ratio approaches $\frac{4}{3}$ at both ends of the range $2c_1 < c_0 < 9c_1/4$.

## 6. Conclusions

We described new algorithms for channel assignment with arbitrary co-site and inter-site constraints on odd cycles and hexagon graphs. For odd cycles, our algorithms are optimal or near-optimal. We conjecture that in the cases where we do not achieve optimality, the existing lower bounds for odd cycles are inadequate. For hexagon graphs, for the case $c_0 < 3c_1$, we give approximation algorithms with performance ratios of at most $\frac{4}{3} + \frac{1}{100}$ (when $c_0 \geqslant 3c_1$, there is a straightforward optimal algorithm). We point out that this matches the performance ratio of the best known algorithm for multicoloring on hexagon graphs, for most values in this range, and is only greater by a very small factor for all values (see Table 2).

Our algorithms are centralized and static algorithms. However, in practice, channel allocation is a distributed and online task; future work will involve the investigation of efficient online and distributed algorithms for this problem.

## Acknowledgements

We would like to thank the anonymous referee for suggesting an improvement in the performance ratio of the algorithm described in Theorem 4.2.

## References

[1] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, Macmillan Press Ltd, New York, 1976.
[2] A. Gamst, Some lower bounds for a class of frequency assignment problems, IEEE Trans. Veh. Technol. 35 (1) (1986) 8–14.
[3] S.N.T. Gerke, Colouring weighted bipartite graphs with a co-site constraint, D.Phil. Thesis, Oxford, 2000.
[4] J. Janssen, K. Kilakos, Adaptive multicolouring, Combinatorica 20 (1) (2000) 87–102.
[5] C. McDiarmid, B. Reed, Channel assignment and weighted colouring, Networks, 1997, to appear.
[6] L. Narayanan, S. Shende, Static frequency assignment in cellular networks, Proc. SIROCCO97, Carleton Scientific Press, 1997, pp. 215–227. Algorithmica, to appear.
[7] N. Schabanel, S. Ubeda, J. Zerovnik, A note on upper bounds for the span of frequency planning in cellular networks, Discrete Appl. Math., submitted.
[8] Tomás Feder, Sunil M. Shende, Online channel allocation in fdma networks with reuse constraints, Inform. Process. Lett. 67 (6) (1998) 295–302.