Algorithmica (2003) 35: 176–191
DOI: 10.1007/s00453-002-0992-3

Algorithmica
© 2002 Springer-Verlag New York Inc.

On-Line Edge-Coloring with a Fixed Number of Colors[1]

Lene Monrad Favrholdt[2] and Morten Nyhave Nielsen[2]

Abstract. We investigate a variant of on-line edge-coloring in which there is a fixed number of colors available and the aim is to color as many edges as possible. We prove upper and lower bounds on the performance of different classes of algorithms for the problem. Moreover, we determine the performance of two specific algorithms, *First-Fit* and *Next-Fit*.

Specifically, algorithms that never reject edges that they are able to color are called fair algorithms. We consider the four combinations of fair/not fair and deterministic/randomized.

We show that the competitive ratio of deterministic fair algorithms can vary only between approximately 0.4641 and $\frac{1}{2}$, and that *Next-Fit* is worst possible among fair algorithms. Moreover, we show that no algorithm is better than $\frac{4}{7}$-competitive.

If the graphs are all k-colorable, any fair algorithm is at least $\frac{1}{2}$-competitive. Again, this performance is matched by *Next-Fit* while the competitive ratio for *First-Fit* is shown to be $k/(2k-1)$, which is significantly better, as long as k is not too large.

Key Words. Edge-coloring, On-line algorithms, Competitive analysis, Fixed number of colors, Maximization problem, Fair algorithms, k-Colorable graphs, Accommodating sequences, Restricted adversary, Randomization.

1. Introduction

The Problem. In this paper we investigate the on-line problem EDGE-COLORING defined in the following way. A number k of colors is given. The algorithm is given the edges of a graph one by one, each one specified by its endpoints. For each edge, the algorithm must either color the edge with one of the k colors or reject it, before seeing the next edge. Once an edge has been colored the color cannot be altered and a rejected edge cannot be colored later. The aim is to color as many edges as possible under the constraint that no two adjacent edges receive the same color.

Note that the problem investigated here is different from the classical version of the edge coloring problem, which is to color *all* edges with as *few* colors as possible. In [2] it is shown that, for the on-line version of the classical edge coloring problem, the greedy algorithm (the one that we call *First-Fit*) is optimal.

The Measures. To measure the quality of the algorithms, we use the competitive ratio which was introduced in [5] and has become a standard measure for on-line algorithms.

---

[1] A preliminary version of this paper appeared in *Proceedings of the Twentieth Conference on the Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, vol. 1974, pp. 106–116, Springer-Verlag, Berlin, 2000. This research was supported in part by the Danish Natural Science Research Council (SNF) and in part by the Future and Emerging Technologies program of the EU under Contract Number IST-1999-14186 (ALCOM-FT).

[2] Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark. {lenem,nyhave}@imada.sdu.dk.

Received June 28, 2001; revised March 7, 2002. Communicated by A. B. Borodin.
Online publication October 29, 2002.

For the problem EDGE-COLORING addressed in this paper, the competitive ratio of an algorithm $\mathbb{A}$ is the worst case ratio, over all possible input sequences, of the number of edges colored by $\mathbb{A}$ to the number of edges colored by an optimal off-line algorithm.

In some cases it may be realistic to assume that the input graphs are all k-colorable. Therefore, we also investigate the competitive ratio in the special case where it is known that the input graphs are k-colorable. This idea is similar to what was done in [1] and [3]. In these papers the competitive ratio is investigated on input sequences that can be fully accommodated by an optimal off-line algorithm with the resources available (in this paper the resource is, of course, the colors). Such sequences are called accommodating sequences. This is generalized in [4], where the competitive ratio as a function of the amount of resources available is investigated.

This paper illustrates an advantage of analyzing accommodating sequences, apart from tailoring the measure to the type of input. A common technique when constructing a difficult proof is to start out investigating easier special cases. In our analysis of the general performance guarantee, the case of k-colorable input graphs was used as such a special case.

The Algorithms. We mainly consider fair algorithms. A *fair* algorithm is an algorithm that never rejects an edge, unless all k colors have already been used on edges adjacent to the new edge. Two natural fair algorithms are *Next-Fit* and *First-Fit* described in Sections 3.4 and 3.5, respectively.

The Results. In Section 2.2 we show that any fair algorithm has a competitive ratio no worse than $2\sqrt{3}-3 \approx 0.4641$. Furthermore, we show that no deterministic fair algorithm is better than $\frac{1}{2}$-competitive, and that no algorithm can be better than $\frac{4}{7}$-competitive, even if we allow randomization. In Section 4 we show that, in the case of k-colorable graphs, any fair algorithm is $\frac{1}{2}$-competitive and that no deterministic algorithm is better than $\frac{2}{3}$-competitive.

The performance of the algorithm *Next-Fit* matches the performance guarantee for fair algorithms in both the general case and in the special case where the input graphs are all k-colorable. The algorithm *First-Fit* is only slightly better. It has a competitive ratio no better than $\frac{2}{9}(\sqrt{10} - 1) \approx 0.4805$ in general and exactly $k/(2k - 1)$ on k-colorable graphs.

The Graphs. The performance guarantees proven in this paper are valid even if we allow multigraphs, i.e., graphs that may have parallel edges, but no loops. The adversary graphs used for proving the impossibility results are all simple graphs. Thus, the impossibility results are valid even if we restrict ourselves to simple graphs. Furthermore, the adversary graphs are all bipartite except one which could easily be changed to a bipartite graph. Thus, the results are all valid for bipartite graphs too.

2. Preliminaries

2.1. *Notation and Terminology.* A *k-coloring* is a coloring using at most k colors. We label the colors $1, 2, \ldots, k$. For any $i, j \in \{1, 2, \ldots, k\}$, we let $C_{i,j}$ denote the subset $\{i, i + 1, \ldots, j\}$ of the k colors.

$K_{m,n}$ denotes the complete bipartite graph in which the two independent sets contain m and n vertices, respectively.

An r*-regular* graph is a graph in which every vertex has degree r. A *biregular* graph is a graph in which each vertex has one of two possible vertex degrees.

The terms fair^D, fair^R, $\mathit{on\text{-}line}^D$, and $\mathit{on\text{-}line}^R$ denote arbitrary on-line algorithms from the classes "fair deterministic", "fair randomized", "deterministic", and "randomized," respectively, for the EDGE-COLORING problem. The term *off-line* denotes an optimal off-line algorithm for the problem.

2.2. *The Competitive Ratio*. We give a formal definition of the competitive ratio for the problem EDGE-COLORING. Note that since the EDGE-COLORING problem is a maximization problem, lower bounds on the competitive ratio are performance guarantees and upper bounds are impossibility results.

DEFINITION 2.1. For any algorithm $\mathbb{A}$ and any sequence S of edges, let $\mathbb{A}(S)$ be the number of edges colored by $\mathbb{A}$ and let $\mathrm{OPT}(S)$ be the number of edges colored by an optimal off-line algorithm. Furthermore, let $0 \leq C \leq 1$.

An on-line algorithm $\mathbb{A}$ is C*-competitive* if $\mathbb{A}(S) \geq C \cdot \mathrm{OPT}(S)$, for any sequence S of edges.

The *competitive ratio* of $\mathbb{A}$ is $C_{\mathbb{A}} = \sup\{C \mid \mathbb{A} \text{ is } C\text{-competitive}\}$.

3. General Graphs

3.1. *A Tight Performance Guarantee for Fair Algorithms*. In this section a tight performance guarantee for fair algorithms is given. Actually, Theorem 3.1 as well as the performance guarantee in Section 4.1 holds with the weaker assumption that the algorithm never rejects an edge e, unless there are at least k colored edges adjacent to e.

The idea behind the proof is the following. For each edge that fair^R colors, it earns one unit of some value. If, for some fraction C of a unit, fair^R can buy all edges colored by *off-line*, paying at least C for each of them, the number of edges colored by fair^R is at least the fraction C of the number of edges colored by *off-line*. If this is the case for any sequence of edges, fair^R is C-competitive.

THEOREM 3.1. *For any fair on-line algorithm* fair^R *for* EDGE-COLORING,

$$C_{\mathit{fair}^R}(k) \geq \min_{d \in C_{1,k}} \left\{ \frac{k^2 + d^2 - kd}{2k^2 - kd} \right\} \geq 2\sqrt{3} - 3 \approx 0.4641.$$

PROOF. Let E_{c} denote the set of edges colored by fair^R, let E_{u} denote the set of edges colored by *off-line* and not by fair^R, and let E_{d} denote the set of edges colored by both *off-line* and fair^R. Thus, $E_{\mathrm{u}} \cup E_{\mathrm{d}}$ are the edges colored by *off-line*, and $E_{\mathrm{d}} \subseteq E_{\mathrm{c}}$. Similarly, for any vertex x, let $d_{\mathrm{c}}(x)$, $d_{\mathrm{u}}(x)$, and $d_{\mathrm{d}}(x)$ denote the number of edges incident to x colored by fair^R, *not* colored by fair^R, and colored by both fair^R and *off-line*, respectively.

Assume that, for each edge $e \in E_{\mathrm{c}}$, fair^R earns one unit of some value. We determine a C, $0 < C < \frac{1}{2}$, such that, for any sequence of edges, the total value earned by fair^R

suffices to buy all edges colored by *off-line*, paying at least C for each. Since E_c are the edges colored by *fair*^{R}, and $E_d \cup E_u$ are the edges colored by *off-line*, this can be expressed as $|E_c| \geq C(|E_d| + |E_u|)$.

Assume that *fair*^{R} starts out buying all edges in E_d, paying C for each. This is clearly possible, since $E_d \subseteq E_c$. The remaining value is distributed to the edges in E_u in two steps. In the first step, each vertex x receives the value $m(x) = \frac{1}{2}(d_c(x) - Cd_d(x))$. Note that $\sum_{x \in V} m(x) = |E_c| - C|E_d|$. In the next step, the value on each vertex is distributed equally among the edges in E_u incident to it. Thus, each vertex x with $d_u(x) \geq 1$ gives the value $m_u(x) = m(x)/d_u(x)$ to each edge in E_u incident to it.

Note that

$$\begin{aligned} \sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) &\leq \sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) + \sum_{d_u(x)=0} m(x) \\ &= \sum_{x \in V} m(x) = |E_c| - C|E_d|. \end{aligned}$$

Thus, if $m_u(x) + m_u(y) \geq C$ for any edge $(x, y) \in E_u$, then

$$C|E_u| \leq \sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) \leq |E_c| - C|E_d|,$$

yielding $|E_c| \geq C(|E_u| + |E_d|)$.

What remains to be done is to find a value of C such that $m_u(x) + m_u(y) \geq C$ for any edge $(x, y) \in E_u$. This is done using calculations based on two simple observations:

(1) For any vertex $x \in V$, $d_d(x) + d_u(x) \leq k$, since *off-line* can color at most k edges incident to x.
(2) For each edge $(x, y) \in E_u$, $d_c(x) + d_c(y) \geq k$, since *fair*^{R} is a fair algorithm.

For any edge $(x, y) \in E_u$,

$$\begin{aligned} m_u(x) + m_u(y) &= \frac{1}{2}\left(\frac{d_c(x) - Cd_d(x)}{d_u(x)} + \frac{d_c(y) - Cd_d(y)}{d_u(y)}\right) \\ &\overset{(1)}{\geq} \frac{1}{2}\left(\frac{d_c(x) - Cd_d(x)}{k - d_d(x)} + \frac{d_c(y) - Cd_d(y)}{k - d_d(y)}\right). \end{aligned}$$

Let

$$m_x = \frac{d_c(x) - Cd_d(x)}{k - d_d(x)} \quad \text{and} \quad m_y = \frac{d_c(y) - Cd_d(y)}{k - d_d(y)}.$$

By (2) it can be assumed without loss of generality that $d_c(y) \geq k/2$. For $d_c(x) \leq kC$, m_x is a monotonically decreasing function of $d_d(x)$, and, for $d_c(x) > kC$, m_x is a monotonically increasing function of $d_d(x)$. Similarly, since $d_c(y) \geq k/2 > kC$, m_y is a monotonically increasing function of $d_d(y)$.

We can now conclude that,

for $d_c(x) > kC$,

$$m_u(x) + m_u(y) \geq \frac{1}{2}\left(\frac{d_c(x)}{k} + \frac{d_c(y)}{k}\right) \overset{(2)}{\geq} \frac{1}{2} > C,$$

and for $d_c(x) \le kC$,

$$\begin{aligned} m_u(x) + m_u(y) &\ge \frac{1}{2}\left(\frac{d_c(x) - Cd_c(x)}{k - d_c(x)} + \frac{d_c(y)}{k}\right) \qquad (\text{since } d_d(x) \le d_c(x)) \\ &\overset{(2)}{\ge} \frac{1}{2}\left(\frac{d_c(x) - Cd_c(x)}{k - d_c(x)} + \frac{k - d_c(x)}{k}\right). \end{aligned}$$

Now,

$$\frac{1}{2}\left(\frac{d_c(x) - Cd_c(x)}{k - d_c(x)} + \frac{k - d_c(x)}{k}\right) \ge C \quad \Longleftrightarrow \quad \frac{k^2 + (d_c(x))^2 - kd_c(x)}{2k^2 - kd_c(x)} \ge C.$$

Thus,

$$C_{fair^R} \ge \min_{d \in C_{1,k}} \left\{\frac{k^2 + d^2 - kd}{2k^2 - kd}\right\} \ge \min_{d \in (0;k]} \left\{\frac{k^2 + d^2 - kd}{2k^2 - kd}\right\} = 2\sqrt{3} - 3. \qquad \square$$

In Section 3.4 it is shown that the competitive ratio of *Next-Fit* exactly matches the performance guarantee of Theorem 3.1.

The next section in conjunction with Theorem 3.1 shows that all deterministic fair algorithms must have very similar competitive ratios.

3.2. *An Impossibility Result for Fair Deterministic Algorithms*

THEOREM 3.2. *Any deterministic fair algorithm for* EDGE-COLORING *is at most* $\frac{1}{2}$-*competitive.*

PROOF. The adversary constructs a simple graph $G = (V_1 \cup V_2, E)$ in two phases. In Phase 1 only vertices in V_1 are connected. In Phase 2 vertices in V_2 are connected to vertices in V_1. Let $|V_1| = |V_2| = n$ for some large integer n.

In Phase 1 the adversary gives an edge between two unconnected vertices $x, y \in V_1$ with a common unused color. Since the edge can be colored, $fair^D$ will do so. This process is repeated until no two unconnected vertices with a common unused color can be found. At that point Phase 1 ends.

For any vertex x, let $\bar{C}(x)$ denote the set of colors not represented at x. At the end of Phase 1, the following holds true. For each color c and each vertex x such that $c \in \bar{C}(x)$, x is already connected to all other vertices y with $c \in \bar{C}(y)$. Since $c \in \bar{C}(x)$, x is connected to at most $k - 1$ other vertices. Thus, each of the k colors are missing at at most k vertices: $\sum_{x \in V_1} \bar{C}(x) \le k^2$.

The edges given in Phase 2 are the edges of a k-regular bipartite graph with V_1 and V_2 forming the two independent sets. Note that, by König's theorem [6, p. 209], such a graph can be k-colored.

In Phase 2 $fair^D$ colors at most k^2 edges, but *off-line* rejects all edges from Phase 1 and colors all edges from Phase 2, giving a performance ratio of at most

$$\frac{\frac{1}{2}(nk - k^2) + k^2}{nk} = \frac{nk + k^2}{2nk} = \frac{1}{2} + \frac{k}{2n}.$$

If we allow n to be arbitrarily large, this can be arbitrarily close to $\frac{1}{2}$.

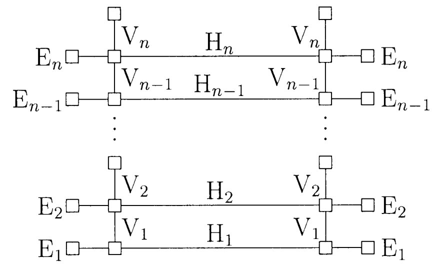

Fig. 1. Structure of the adversary graph for the general impossibility result.

Note that for $k = 1$, Phase 1 may include only one edge. Thus, for $k = 1$, a graph with only three edges gives a ratio of exactly $\frac{1}{2}$. $\square$

Note that the proof of Theorem 3.2 can be easily modified to be valid even if the input graphs are restricted to being bipartite. The vertex set V_1 should be replaced by two sets V_1^{L} and V_1^{R}, and the edges of Phase 1 should connect vertices in V_1^{L} to vertices in V_1^{R}. In this case, at the end of Phase 1, each color is missing at at most $2k-2$ vertices, because, if a color is missing at a vertex in V_1^{L}, then it can be missing at at most $k-1$ vertices in V_1^{R} and vice versa. Clearly, half of the vertices of Phase 2 should be connected to V_1^{L}, the other half to V_1^{R}.

3.3. *A General Impossibility Result*. Now follows an impossibility result for any type of algorithm for EDGE-COLORING, fair or not fair, deterministic or randomized.

THEOREM 3.3. *Any algorithm for* EDGE-COLORING *is at most* $\frac{4}{7}$*-competitive*.

PROOF. The structure of the adversary graph is depicted in Figure 1. Each box contains k vertices. When two boxes are connected, there are k^2 edges in a complete bipartite graph between the $2k$ vertices inside the boxes. Note that this bipartite graph can be k-colored. The edges of the graph are divided into n levels, level $1, \ldots, n$. The adversary gives the edges, one level at a time, according to the numbering of the levels. The edges of level i are given in three consecutive phases:

1. H_i: Internal (horizontal) edges at level i. In total k^2 edges.
2. V_i: Internal (vertical) edges between level i and level $i+1$. In total $2k^2$ edges.
3. E_i: External edges at level i. In total $2k^2$ edges.

Vertices that are endpoints of internal edges are called internal vertices.

Let X_{H_i} be a random variable counting how many edges *on-line*^{R} will color from the set H_i, and let X_{V_i} and X_{E_i} count the colored edges from V_i and E_i, respectively.

For $i = 0, \ldots, n$, let EXT_i and INT_i be random variables counting the sum of all external and internal edges, respectively, colored by *on-line*^{R} after level i is given, i.e., $\mathrm{EXT}_i = \sum_{j=1}^{i} X_{\mathrm{E}_j}$ and $\mathrm{INT}_i = \sum_{j=1}^{i} (X_{\mathrm{V}_j} + X_{\mathrm{H}_j})$. Note that $\mathrm{EXT}_0 = \mathrm{INT}_0 = 0$.

If the adversary stops giving edges after Phase 1 of level i, *off-line* will color $k^2(2i-1)$ edges in total. These are the edges in the sets $\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_{i-1}$, and H_i. If the adversary stops giving edges after Phase 2 (or 3) of level i, *off-line* will color $2k^2 i$ edges. These are the edges in the sets $\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_{i-1}$, and V_i.

To prove the bound we use the following observations. Let G_i denote the graph consisting of the first i levels. Consider the subgraph G'_i of G_i colored by *on-line*^{R}. Summing over all *internal* vertices, the total vertex degree in G'_i is at most $2k^2 i$. An internal edge (excluding V_i) contributes two to this number, whereas an external edge (plus edges in V_i) will only contribute one. Thus, the expected number of edges in G_i colored by *on-line*^{R} is

$$(1)\quad \begin{aligned} E[\mathrm{INT}_i]+E[\mathrm{EXT}_i] &= (E[\mathrm{INT}_i]-E[X_{\mathrm{V}_i}])+(E[\mathrm{EXT}_i]+E[X_{\mathrm{V}_i}]) \\ &\leq \tfrac{1}{2}(2k^2 i - E[\mathrm{EXT}_i] - E[X_{\mathrm{V}_i}]) + (E[\mathrm{EXT}_i] + E[X_{\mathrm{V}_i}]) \\ &= k^2 i + \tfrac{1}{2}(E[\mathrm{EXT}_i] + E[X_{\mathrm{V}_i}]). \end{aligned}$$

The rest of the proof is divided into two cases.

Case 1: *There exists a level* $i \leq n$, *where* $E[\mathrm{EXT}_i] > \frac{2}{7}k^2 i$. We will show by contradiction that in this case *on-line*^{R} is not $\frac{4}{7}$-competitive. Let i denote the first level such that $E[\mathrm{EXT}_i] > \frac{2}{7}k^2 i$. Then

$$(2)\qquad E[\mathrm{EXT}_{i-1}] \leq \tfrac{2}{7}k^2(i-1)$$

and

$$(3)\qquad E[X_{\mathrm{E}_i}] > \tfrac{2}{7}k^2.$$

Assume that the number of edges colored by *on-line*^{R} is at least $\frac{4}{7}$ of the number of edges colored by *off-line*. If the adversary stops the sequence after Phase 1 of level i, the following inequality must hold:

$$(4)\qquad E[\mathrm{INT}_{i-1}] + E[\mathrm{EXT}_{i-1}] + E[X_{\mathrm{H}_i}] \geq \tfrac{4}{7}k^2(2i-1).$$

If the adversary stops the sequence after Phase 2 of level i, the following inequality must hold:

$$(5)\qquad E[\mathrm{INT}_{i-1}] + E[\mathrm{EXT}_{i-1}] + E[X_{\mathrm{H}_i}] + E[X_{\mathrm{V}_i}] \geq \tfrac{4}{7}k^2 2i.$$

If *on-line*^{R} is $\frac{4}{7}$-competitive, both inequalities must hold. Adding inequalities (4) and (5) yields

$$(6)\qquad 2(E[\mathrm{INT}_{i-1}] + E[\mathrm{EXT}_{i-1}]) + 2E[X_{\mathrm{H}_i}] + E[X_{\mathrm{V}_i}] \geq \tfrac{16}{7}k^2 i - \tfrac{4}{7}k^2.$$

Furthermore,

$$(7)\qquad E[X_{\mathrm{V}_{i-1}}] + 2E[X_{\mathrm{H}_i}] + E[X_{\mathrm{V}_i}] \leq 2k^2 - E[X_{\mathrm{E}_i}] < \tfrac{12}{7}k^2,$$

where the first inequality follows from the fact that the number of colored edges incident to the internal vertices at level i is at most $2k^2$, and the second inequality follows from (3). Combining inequality (1) (for $i-1$) with (6) and then using (7) yields a contradiction with (2). Thus, in this case *on-line*^{R} is not $\frac{4}{7}$-competitive.

Case 2: *For all* $i \leq n$, $E[\text{EXT}_i] \leq \frac{2}{7}k^2 i$. By (1), the expected number of edges colored by *on-line*^{R} is

$$\begin{aligned} E[\text{INT}_n] + E[\text{EXT}_n] &\leq k^2 n + \tfrac{1}{2}(E[\text{EXT}_n] + E[X_{\text{V}_n}]) \\ &= k^2 n + \tfrac{1}{2}E[\text{EXT}_{n-1}] + \tfrac{1}{2}(E[X_{\text{E}_n}] + E[X_{\text{V}_n}]) \\ &\leq k^2 n + \tfrac{1}{7}k^2(n-1) + \tfrac{1}{2}2k^2 \\ &= \tfrac{8}{7}k^2 n + \tfrac{6}{7}k^2. \end{aligned}$$

Thus, we get an upper bound on the performance ratio of $(\frac{8}{7}k^2 n + \frac{6}{7}k^2)/2nk^2 = \frac{4}{7} + 3/7n$, which can be arbitrarily close to $\frac{4}{7}$, if we allow n to be arbitrarily large. □

Thus, even if we allow randomized algorithms that are not necessarily fair, no algorithm is more than 0.11 apart from the worst fair algorithm when comparing competitive ratios.

3.4. *The Algorithm Next-Fit*. The algorithm *Next-Fit* (*NF*) is a fair algorithm that uses the colors in a cyclic order. *Next-Fit* colors the first edge with the color 1 and keeps track of the last used color c_{last}. When coloring an edge (u, v) it uses the first color in the sequence $\langle c_{\text{last}} + 1, c_{\text{last}} + 2, \ldots, k, 1, 2, \ldots, c_{\text{last}} \rangle$ that is not yet used on any edge incident to u or v, if any.

Intuitively, this is a poor strategy and it turns out that its worst case performance matches the performance guarantee of Section 3.1. Thus, this algorithm is mainly described here to show that the performance guarantee cannot be improved.

When proving impossibility results for *Next-Fit*, the following claim is useful.

CLAIM 3.4. *Any coloring in which each color is used on exactly* n *or* $n + 1$ *edges*, *for some* $n \in \mathbb{N}$, *can be produced by Next-Fit*, *for some ordering of the input sequence*. *The colors just need to be permuted so that the colors used on* $n + 1$ *edges are the lowest numbered colors*. *With the colors permuted this way*, *the adversary can give an edge with color* 1 *followed by an edge with color* 2, *and so on until all* k *colors have been used*. *This pattern is followed* n *times and*, *finally*, *remaining edges are given*, *again ordered according to color.*

Now follows a theorem showing that *Next-Fit* is worst possible among fair algorithms.

THEOREM 3.5.

$$C_{NF}(k) \leq \min_{d \in C_{1,k}} \left\{ \frac{k^2 + d^2 - kd}{2k^2 - kd} \right\} \quad \textit{and} \quad \inf_{k \in \mathbb{N}} \{C_{NF}(k)\} \leq 2\sqrt{3} - 3 \approx 0.4641.$$

PROOF. The adversary constructs a graph G_{NF} in the following way. It chooses a $d \in C_{1,k}$ close to $(2 - \sqrt{3})k$ and constructs a d-regular bipartite graph $G_1 = (L_1 \cup R_1, E_1)$ with $|L_1| = |R_1| = k$ and a graph $G_2 = (L_2 \cup R_2, E_2)$ isomorphic to $K_{k-d,k-d}$ ($K_{1,1}$ if $k = 1$). Now, each vertex in R_1 is connected to each vertex in L_2 and each vertex in R_2 is connected to each vertex in L_1. Call these extra edges E_{12}. The graph G_{NF} for $k = 4$

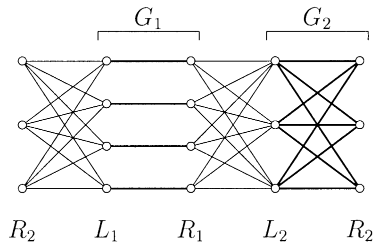

Fig. 2. The graph G_{NF} when $k = 4$ and $d = 1$, showing that $C_{NF}(4) \leq \frac{13}{28} \approx 0.4643$.

is depicted in Figure 2. Note that the three leftmost vertices are the same as the three rightmost vertices.

Note that G_1 is d-colorable and G_2 is $(k-d)$-colorable. If the edges of G_1 are colored with $C_{1,d}$, each of the d colors will be represented at each vertex of G_1. Similarly, if G_2 is colored with $C_{d+1,k}$, each color in $C_{d+1,k}$ will be represented at each vertex of G_2. After this, none of the edges in E_{12} can be colored. However, the edge set $E_1 \cup E_{12}$ can be k-colored.

The adversary uses k copies of G_{NF}, $G^1_{NF}, \ldots, G^k_{NF}$. Consider a coloring where G^1_1 is colored with $C_{1,d}$ and G^1_2 is colored with $C_{d+1,k}$, G^2_1 is colored with $C_{2,d+1}$ and G^2_2 is colored with $C_{d+2,k} \cup \{1\}, \ldots, G^k_1$ is colored with $\{k\} \cup C_{1,d-1}$ and G^k_2 is colored with $C_{d,k-1}$. That is, to obtain the coloring of G^{i+1}_j from G^i_j, the colors are shifted once. In this coloring, each color is used the same number of times, so, by Claim 3.4, it can be produced by *Next-Fit*. Hence, for any $d \in C_{1,k}$, the competitive ratio of *Next-Fit* can be no more than

$$\frac{|E_1| + |E_2|}{|E_1| + |E_{12}|} = \frac{kd + (k-d)^2}{kd + 2k(k-d)} = \frac{k^2 - kd + d^2}{2k^2 - kd}.$$

This ratio attains its minimum value of $2\sqrt{3}-3$ when $d = (2-\sqrt{3})k$. Thus, by allowing arbitrarily large values of k, it can be arbitrarily close to $2\sqrt{3} - 3$. □

3.5. *The Algorithm First-Fit*. The algorithm *First-Fit* (*FF*) is a fair algorithm. For each edge e that it is able to color, it colors e with the lowest numbered color possible.

The following theorem gives an impossibility result for *First-Fit*.

THEOREM 3.6.

$$C_{FF}(k) \leq \min_{d \in C_{1,k}} \left\{ \frac{2k^2 - 2kd + d^2}{4k^2 - 3kd} \right\} \quad \textit{and} \quad \inf_{k \in \mathbb{N}} \{C_{FF}(k)\} \leq \tfrac{2}{9}(\sqrt{10} - 1) \approx 0.4805.$$

PROOF. The adversary graph G_{FF} of this proof is inspired by the graph G_{NF}. It is not possible, though, to make *First-Fit* color the subgraph G_2 of G_{NF} with $C_{d+1,k}$. Therefore, the graph is extended to contain an extra copy of G_2, G'_2. Each vertex in R_2 is connected to exactly d vertices in L'_2 and vice versa. Now, E_2 denotes the edges in G_2 and G'_2 and the edges connecting them. Finally, $2k(k-d)$ new vertices are added, and each vertex in $R_2 \cup L'_2$ is connected to k of these vertices. Let E_3 denote the set of these extra edges. The graph G_{FF} for $k = 4$ is depicted in Figure 3.

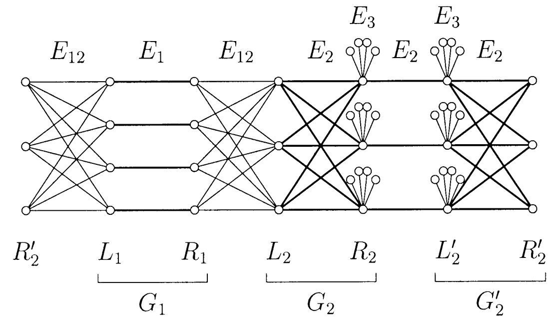

Fig. 3. The graph G_{FF} when $k = 4$, showing that $C_{FF}(4) \leq \frac{25}{52} \approx 0.4808$.

If the edges in G_1 and the edges between R_2 and L'_2 are given first (one perfect matching at a time), followed by the edges in G_2 and G'_2 (one perfect matching at a time), *First-Fit* will color E_1 and the edges between R_2 and L'_2 with $C_{1,d}$ and the remaining edges in E_2 with $C_{d+1,k}$. After this, *First-Fit* will not be able to color any more edges of G_{FF}. On the other hand, it is possible to k-color the set $E_1 \cup E_{12} \cup E_3$ of edges. Thus, the competitive ratio of *First-Fit* can be no more than

$$\frac{|E_1| + |E_2|}{|E_1| + |E_{12}| + |E_3|} = \frac{kd + 2(k-d)^2 + (k-d)d}{kd + 2k(k-d) + 2k(k-d)} = \frac{2k^2 - 2kd + d^2}{4k^2 - 3kd}.$$

This ratio attains its minimum value of $\frac{2}{9}(\sqrt{10} - 1)$, when $d = \frac{1}{3}(4 - \sqrt{10})k$. Thus, for the graph G_{FF}, we choose d to be an integer close to $\frac{1}{3}(\sqrt{10} - 1)k$, and by allowing arbitrarily large values of k, the ratio can be arbitrarily close to $\frac{2}{9}(\sqrt{10} - 1)$. $\square$

4. *k*-Colorable Graphs.

Now that we know that the competitive ratio cannot vary much between different kinds of algorithms for the EDGE-COLORING problem, it would be interesting to see what happens if we know something about the input graphs—for instance that they are all k-colorable. In this section we investigate the competitive ratio in the case where the input graphs are known to be k-colorable.

4.1. *A Performance Guarantee for Fair Algorithms.*

In this section a performance guarantee for any *fair* algorithm is given. As in the proof of Theorem 3.1 the idea is that each colored edge is worth one unit of some value. Again the value of each colored edge e is distributed equally among its endpoints and, from there, redistributed to the uncolored edges adjacent to e. If each uncolored edge receives a total value of at least one, then there are at least as many colored edges as uncolored edges.

THEOREM 4.1. *On k-colorable graphs, any fair algorithm for* EDGE-COLORING *is* $\frac{1}{2}$-*competitive.*

PROOF. Let $G = (V, E)$ be an arbitrary k-colorable graph. Let E_c denote the set of edges colored by $fair^R$, and let E_u denote the set of edges *not* colored by $fair^R$. Similarly, for any vertex x, let $d_c(x)$ denote the number of edges incident to x that are colored by *fair*^{R}, and let $d_u(x)$ denote the number of edges incident to x that are not colored by *fair*^{R}. Then *fair*^{R} is $\frac{1}{2}$-competitive if $|E_c| \geq |E_u|$.

Now, for each vertex $x \in V$, let $m(x) = \frac{1}{2}d_c(x)$. Note that $\sum_{x\in V} m(x) = |E_c|$. For each vertex $x \in V$ such that $d_u(x) \geq 1$, define $m_u(x) = m(x)/d_u(x)$. Then

$$\begin{aligned}\sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) &\leq \sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) + \sum_{d_u(x)=0} m(x)\\ &= \sum_{x\in V} m(x) = |E_c|.\end{aligned}$$

In what follows we will prove that $m_u(x) + m_u(y) \geq 1$ for every edge $(x, y) \in E_u$, giving the desired inequality:

$$|E_u| \leq \sum_{(x,y)\in E_u} (m_u(x) + m_u(y)) \leq |E_c|.$$

Let $(x, y) \in E_u$. Since G is k-colorable, $d_c(x)+d_u(x) \leq k$, yielding the first inequality below. Note that, since $d_u(x), d_u(y) \geq 1$, this means that $d_c(x), d_c(y) \leq k - 1$. Thus, the two divisions on the right-hand side of the inequality are not divisions by zero. The second inequality follows from the fact that $d_c(x) + d_c(y) \geq k$, since *fair*^{R} is a fair algorithm. Finally, the last inequality holds, since $x + 1/x \geq 2$, for any $x > 0$.

$$\begin{aligned}m_u(x) + m_u(y) &= \frac{1}{2}\left(\frac{d_c(x)}{d_u(x)} + \frac{d_c(y)}{d_u(y)}\right) \geq \frac{1}{2}\left(\frac{d_c(x)}{k - d_c(x)} + \frac{d_c(y)}{k - d_c(y)}\right)\\ &\geq \frac{1}{2}\left(\frac{d_c(x)}{k - d_c(x)} + \frac{k - d_c(x)}{d_c(x)}\right) \geq 1.\end{aligned}$$ □

In Section 4.3 it is shown that, on k-colorable graphs, the competitive ratio of the algorithm *Next-Fit* is $\frac{1}{2}$ for all even k. Thus, the result in Theorem 4.1 is tight.

4.2. *An Impossibility Result for Deterministic Algorithms*. If $k = 1$, any fair algorithm is clearly 1-competitive on k-colorable graphs. The following theorem gives an impossibility result for all other values of k.

THEOREM 4.2. *When $k \geq 2$, any deterministic algorithm for* EDGE-COLORING *is at most $\frac{2}{3}$-competitive, even on k-colorable graphs.*

PROOF. The adversary gives a $\lceil k/2\rceil$-regular bipartite graph $G = (L \cup R, E)$ with $|L| = |R| = N$, for some large integer N. For each vertex $x \in L \cup R$, let $C(x)$ be the set of colors with which *on-line*^{D} has colored the edges incident to x. Let $p = \sum_{i=0}^{\lceil k/2\rceil} \binom{k}{i}$. Then there are p possibilities $C_1, C_2, \ldots, C_p$ for $C(x)$. Let $S_i^L = \{x \in L \mid C(x) = C_i\}$ and $S_i^R = \{x \in R \mid C(x) = C_i\}$. For each i, the vertices in S_i^L are partitioned into $\lfloor |S_i^L|/k\rfloor$ subsets of size k and at most one subset of size $|S_i^L| - k\lfloor |S_i^L|/k\rfloor$. The same is done to S_i^R. Let $\mathcal{S}$ be the family of all these subsets. Then $|\mathcal{S}| \geq 2(N - (k - 1)p)/k$. Thus, if N is chosen sufficiently large, the number of vertices contained in the sets in $\mathcal{S}$

will be much larger than the number of vertices not contained in the sets in $\mathcal{S}$. Thus, we can ignore the edges not contained in the sets in $\mathcal{S}$.

Now, for each set $S \in \mathcal{S}$, $\lfloor k/2 \rfloor$ new vertices are created, and each of these $\lfloor k/2 \rfloor$ vertices are connected to each vertex in S. Assume that for each vertex $x \in S$, $|C(x)| = d$. Note that $d \leq \lceil k/2 \rceil$. Then *on-line*^{D} can color at most $k - d$ edges incident to each of the new vertices. Now, looking at the subgraph colored by *on-line*^{D} and summing the vertex degrees of the vertices in S and the $\lfloor k/2 \rfloor$ new vertices, we get at most $kd + 2 \cdot \lfloor k/2 \rfloor (k - d)$, which reduces to k^2 if k is even and to $k^2 - k + d \leq k^2 - \frac{1}{2}k + \frac{1}{2}$ if k is odd. Since $S \subseteq L$ or $S \subseteq R$, the whole graph is bipartite. Furthermore, it has maximum degree k. Thus, by König's theorem, it can be k-colored off-line. Looking at the whole graph, and summing the vertex degrees of the vertices in S and the $\lfloor k/2 \rfloor$ new vertices, we get $k^2 + \lfloor k/2 \rfloor \cdot k$ which reduces to $\frac{3}{2}k^2$, if k is even, and to $\frac{3}{2}k^2 - \frac{1}{2}k$, if k is odd. Thus, for any deterministic algorithm $\mathbb{A}$ for EDGE-COLORING,

$$C_{\mathbb{A}}(k) \leq \begin{cases} \dfrac{k^2}{\frac{3}{2}k^2} = \dfrac{2}{3} & \text{if } k \text{ is even,} \\[2ex] \dfrac{k^2 - \frac{1}{2}k - \frac{1}{2}}{\frac{3}{2}k^2 - \frac{1}{2}k} = \dfrac{2}{3} - \dfrac{k-3}{9k^2 - 3k} \leq \dfrac{2}{3} & \text{if } k \geq 3 \text{ and odd.} \end{cases} \qquad \square$$

4.3. *The Algorithm Next-Fit*. The following theorem shows that *Next-Fit* is worst possible among fair algorithms.

THEOREM 4.3. *On k-colorable graphs*,

$$C_{NF}(k) \leq \begin{cases} \dfrac{1}{2} & \textit{if } k \textit{ is even,} \\[2ex] \dfrac{1}{2} + \dfrac{1}{2k^2} & \textit{if } k \textit{ is odd.} \end{cases}$$

PROOF. The adversary constructs a graph G_{NF} in the following way. First it constructs two complete bipartite graphs $G_1 = (L_1 \cup R_1, E_1)$ with $|L_1| = |R_1| = \lceil k/2 \rceil$ and $G_2 = (L_2 \cup R_2, E_2)$ with $|L_2| = |R_2| = \lfloor k/2 \rfloor$ (see Figure 4). G_1 can be colored with $\lceil k/2 \rceil$ colors using each color $\lceil k/2 \rceil$ times, and G_2 can be colored with $\lfloor k/2 \rfloor$ colors using each color $\lfloor k/2 \rfloor$ times. The edges in these two graphs are given in an order such that *Next-Fit* colors G_1 with $C_{1,\lceil k/2 \rceil}$ and G_2 with $C_{\lceil k/2 \rceil + 1, k}$. Now, each vertex in R_1 is connected to each vertex in L_2 and each vertex in R_2 is connected to each vertex in L_1. Let E_{12} denote these edges connecting G_1 and G_2. *Next-Fit* is not able to color any of the edges in E_{12}. It is, however, possible to color all edges in G_{NF} with $C_{1,k}$, since the graph is bipartite and has maximum degree k. Thus, even in the case where the input graphs are all k-colorable, the competitive ratio of *Next-Fit* can be no more than

$$\frac{|E_1| + |E_2|}{|E_1| + |E_2| + |E_{12}|} = \frac{\lceil k/2 \rceil^2 + \lfloor k/2 \rfloor^2}{\lceil k/2 \rceil^2 + \lfloor k/2 \rfloor^2 + 2\lceil k/2 \rceil \lfloor k/2 \rfloor},$$

which reduces to $\frac{1}{2}$ when k is even, and to $\frac{1}{2} + 1/2k^2$ when k is odd. $\square$

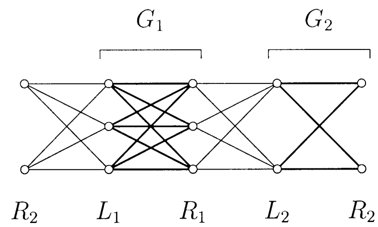

Fig. 4. The graph G_{NF} when $k = 5$.

4.4. *The Algorithm First-Fit*. We now show that for small values of k, the competitive ratio of *First-Fit* on k-colorable graphs is significantly better than that of *Next-Fit*, but the difference tends to zero as k approaches infinity.

THEOREM 4.4. *On k-colorable graphs*, $C_{FF}(k) = k/(2k-1)$.

The theorem is an immediate consequence of the next two lemmas. First, the performance guarantee.

LEMMA 4.5. *On k-colorable graphs*, $C_{FF}(k) \geq k/(2k-1)$.

PROOF. Let E be the edge set of an arbitrary k-colorable graph G. For $c \in C_{1,k}$, let E_c denote the set of edges that *First-Fit* colors with the color c. We will prove by induction on c that, for all $c \in C_{1,k}$, $\sum_{i=1}^{c} |E_i| \geq (c/(2k-1))|E|$.

For the base case, consider $c = 1$. By the definition of *First-Fit*, each edge in $E \setminus E_1$ is adjacent to at least one edge in E_1. Furthermore, since G is k-colorable, each edge in E_1 is adjacent to at most $2(k-1)$ other edges. Thus, $|E| \leq 2(k-1)|E_1| + |E_1|$, or $|E_1| \geq (1/(2k-1))|E|$.

For the induction step, let $c \in C_{1,k}$. Since each edge in E_c is adjacent to at least $c-1$ edges in $\bigcup_{i=1}^{c-1} E_i$, each edge in E_c is adjacent to at most $2(k-1)-(c-1) = 2k-c-1$ edges in $E \setminus \bigcup_{i=1}^{c} E_i$. On the other hand, each edge in $E \setminus \bigcup_{i=1}^{c} E_i$ is adjacent to at least one edge in E_c. Therefore, $|E \setminus \bigcup_{i=1}^{c-1} E_i| \leq (2k-c-1)|E_{\mathrm{c}}| + |E_{\mathrm{c}}|$, or $|E_c| \geq (1/(2k-c))|E \setminus \bigcup_{i=1}^{c-1} E_i|$. Thus,

$$\begin{aligned}
\sum_{i=1}^{c} |E_i| &\geq \sum_{i=1}^{c-1} |E_i| + \frac{|E| - \sum_{i=1}^{c-1} |E_i|}{2k-c} = \frac{|E| + (2k-c-1)\sum_{i=1}^{c-1} |E_i|}{2k-c} \\
&\geq \frac{|E| + (2k-c-1)((c-1)/(2k-1))|E|}{2k-c} \qquad \text{(by the induction hypothesis)} \\
&= \frac{|E| - ((c-1)/(2k-1))|E|}{2k-c} + \frac{c-1}{2k-1}|E| \\
&= \frac{(2k-1)-(c-1)}{(2k-1)(2k-c)}|E| + \frac{c-1}{2k-1}|E| \\
&= \frac{c}{2k-1}|E|.
\end{aligned}$$

□

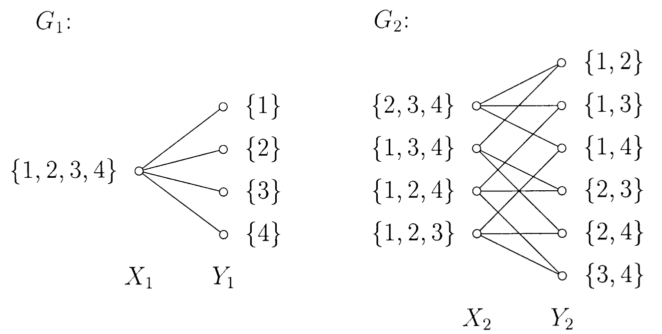

Fig. 5. The graphs G_1 and G_2 when $k = 4$. Next to each vertex v the color set $C(v)$ is shown.

Next, the matching impossibility result.

LEMMA 4.6. *On k-colorable graphs*, $C_{FF}(k) \leq k/(2k-1)$.

PROOF. The edges of the adversary graph are given in two phases. In Phase 1 the edges of k bipartite biregular graphs are given in an order such that *First-Fit* will color all of them. In Phase 2 the bipartite graphs are connected through edges that *First-Fit* cannot color. The resulting graph is called G. In G every vertex has degree k, and no edge is adjacent to more than one edge of each color in the *First-Fit* coloring. For such a graph the analysis in the proof of Lemma 4.5 is tight, meaning that *First-Fit* colors exactly $k/(2k-1)$ of the edges. Furthermore, G is bipartite. Thus, *off-line* colors all of the edges.

Phase 1. The building blocks are $\lceil k/2 \rceil$ bipartite biregular graphs, $G_1, G_2, \ldots, G_{\lceil k/2 \rceil}$. For each i, G_i has vertex partition (X_i, Y_i). X_i has one vertex corresponding to each subset of $C_{1,k}$ of size $k+1-i$, and Y_i has one vertex corresponding to each subset of $C_{1,k}$ of size i (see Figure 5). For each vertex v in G_i, let $C(v)$ denote the set of colors corresponding to v and let $\bar{C}(v) = C_{1,k} \backslash C(v)$. Each vertex $x \in X_i$ is connected to every vertex in $\{y \in Y_i \mid C(x) \cup C(y) = C_{1,k}\}$. Note that, for each edge (x, y), $|C(x)| + |C(y)| = (k+1-i) + i = k+1$. Thus, $|C(x) \cap C(y)| = 1$. We now investigate the coloring in which each edge (x, y) receives the color in $C(x) \cap C(y)$.

Let $x \in X_i$, for some i. Then, for each $c \in C(x)$, there is exactly one vertex $y \in Y_i$ such that $C(x) \cap C(y) = \{c\}$. Similarly, if $y \in Y_i$, then, for each $c \in C(y)$, there is exactly one vertex $x \in X_i$ such that $C(x) \cap C(y) = \{c\}$. This shows that no two adjacent edges are given the same color. It also shows that each vertex $x \in X_i$ has degree $|C(x)|$ and each vertex $y \in Y_i$ has degree $|C(y)|$.

Every edge (x, y) is adjacent to an edge of each color $c \in C_{1,k} \backslash (C(x) \cap C(y))$. Thus, the coloring is obtained if *First-Fit* is given the edges in order of nondecreasing color.

Finally, no edge (x, y) is adjacent to more than one edge of each color, since $|C(x)| + |C(y) \backslash (C(x) \cap C(y))| = k$ and $C(x) \cup C(y) = C_{1,k}$.

Now, k bipartite biregular graphs $G_1^{\mathrm{L}}, G_2^{\mathrm{L}}, \ldots, G_{\lceil k/2 \rceil}^{\mathrm{L}}$, and $G_1^{\mathrm{R}}, G_2^{\mathrm{R}}, \ldots, G_{\lfloor k/2 \rfloor}^{\mathrm{R}}$ are constructed. For $i \in \{1, 2 \ldots, \lceil k/2 \rceil\}$, G_i^{L} consists of a number of copies of G_i. Let

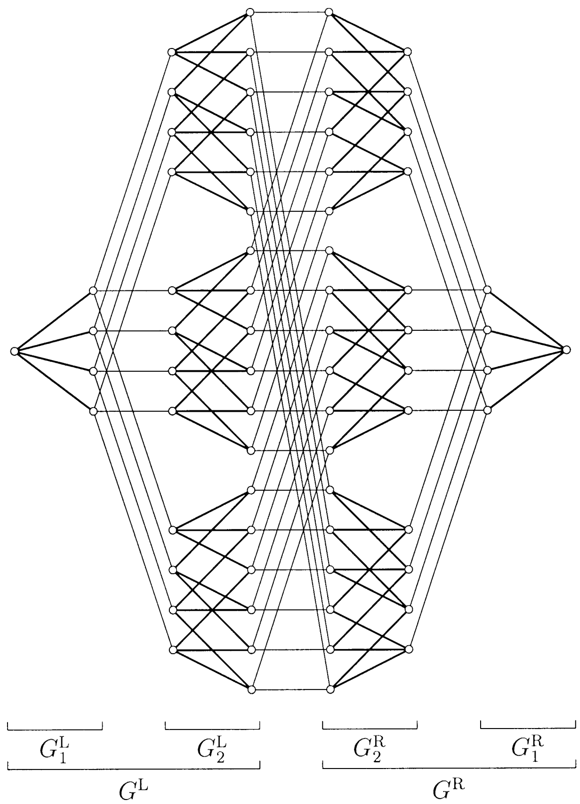

Fig. 6. The graph G when $k = 4$.

n_i be the number of copies of G_i in G_i^{L}. Then $n_1 = 1$ and $n_{i+1} = ((k-i)/i)n_i$, for $i \in \{1, 2 \ldots, \lceil k/2 \rceil - 1\}$. For $i \in \{1, 2 \ldots, \lfloor k/2 \rfloor\}$, G_i^{R} is isomorphic to G_i^{L} (see Figure 6).

Phase 2. Let $i \in \{1, 2, \ldots, \lceil k/2 \rceil - 1\}$. For each pair of vertices $y \in Y_i$ and $x \in X_{i+1}$, $|C(y)| + |C(x)| = i + k + 1 - (i+1) = k$. Thus, for each vertex $y \in Y_i$, there is exactly one vertex $x \in X_{i+1}$ such that $C(y) \cup C(x) = C_{1,k}$. Since G_{i+1}^{L} contains at least $k - i$ copies of G_{i+1}, each vertex $y \in Y_i^{\mathrm{L}}$ can be connected to $k - i$ vertices in $\{x \in X_{i+1}^{\mathrm{L}} \mid C(x) \cup C(y) = C_{1,k}\}$. Since the number of copies of G_i in G_i^{L} is exactly $i/(k-i)$ times the number of copies of G_{i+1} in G_{i+1}^{L}, this can be done such that every vertex in X_{i+1}^{L} is connected to exactly i vertices in Y_i^{L}. All these edges are now added, yielding a connected graph G^{L} in which every vertex, except the vertices in $Y_{\lceil k/2 \rceil}^{\mathrm{L}}$, has degree k. A graph G^{R} is constructed from $G_1^{\mathrm{R}}, G_2^{\mathrm{R}}, \ldots, G_{\lfloor k/2 \rfloor}^{\mathrm{R}}$ in the same way. Note that in G^{R}, $Y_{\lfloor k/2 \rfloor}$ plays the role of $Y_{\lceil k/2 \rceil}$ in G^{L}. Finally, G^{L} and G^{R} are connected through edges connecting pairs of vertices $y^{\mathrm{L}} \in Y_{\lceil k/2 \rceil}^{\mathrm{L}}$ and $y^{\mathrm{R}} \in Y_{\lfloor k/2 \rfloor}^{\mathrm{R}}$ such that $C(y^{\mathrm{L}}) \cup C(y^{\mathrm{R}}) = C_{1,k}$ and in a way so that each vertex in $Y_{\lceil k/2 \rceil}^{\mathrm{L}} \cup Y_{\lfloor k/2 \rfloor}^{\mathrm{R}}$ ends up having degree k. The resulting graph is denoted by G.

For each edge (x, y) given in Phase 2, $C(x) \cup C(y) = C_{1,k}$. Thus, *First-Fit* cannot color the edge. Furthermore, $|C(x)| + |C(y)| = k$. Thus, (x, y) is not connected to more than one edge of each color. This completes the proof. □

5. Conclusion. We have proven that the competitive ratios of algorithms for EDGE-COLORING can vary only between approximately 0.46 and 0.5 for fair deterministic algorithms and between 0.46 and 0.57 for randomized algorithms (it can, of course, be worse for algorithms that are not fair). Thus, we cannot hope for algorithms with competitive ratios much better than those of *Next-Fit* and *First-Fit*. In the case of k-colorable graphs the gap is somewhat larger: the (tight) performance guarantee for fair algorithms is $\frac{1}{2}$ and the impossibility result for deterministic algorithms is $\frac{2}{3}$. In this case we have no impossibility result for randomized algorithms.

We have shown that *Next-Fit* is worst possible among fair algorithms in both the general case and in the special case of k-colorable graphs. Furthermore, we have found the exact competitive ratio of *First-Fit* on k-colorable graphs. For small values of k it is significantly better than that of *Next-Fit*, but for large values of k they can hardly be distinguished. In the general case, the competitive ratios of *First-Fit* and *Next-Fit* are very close. We believe that the competitive ratio of *First-Fit* is a little better than that of *Next-Fit* but we have not proven it.

References

[1] Yossi Azar, Joan Boyar, Lene M. Favrholdt, Kim S. Larsen, and Morten N. Nielsen. Fair versus Unrestricted Bin Packing. In *Proceedings of the Seventh Scandinavian Workshop on Algorithm Theory*, volume 1851 of Lecture Notes in Computer Science, pages 200–213. Springer-Verlag, Berlin, 2000.

[2] Amotz Bar-Noy, Rajeev Motwani, and Joseph Naor. The Greedy Algorithm Is Optimal for On-Line Edge Coloring. *Information Processing Letters*, 44(5):251–253, 1992.

[3] Joan Boyar and Kim S. Larsen. The Seat Reservation Problem. *Algorithmica*, 25(4):403–417, 1999.

[4] Joan Boyar, Kim S. Larsen, and Morten N. Nielsen. The Accommodating Function—a Generalization of the Competitive Ratio. *SIAM Journal on Computing*, 31(1):233–258, 2001.

[5] Anna R. Karlin, Mark S. Manasse, Larry Rudolph, and Daniel D. Sleator. Competitive Snoopy Caching. *Algorithmica*, 3:79–119, 1988.

[6] Douglas B. West. *Introduction to Graph Theory*, page 209. Prentice-Hall, Englewood Cliffs, NJ, 1996.