Theoretical Computer Science 411 (2010) 1734–1741

Contents lists available at ScienceDirect

Theoretical Computer Science

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

# Comparing First-Fit and Next-Fit for online edge coloring

Martin R. Ehmsen [a,*], Lene M. Favrholdt [a], Jens S. Kohrt [a], Rodica Mihai [b]

[a] Department of Mathematics and Computer Science, University of Southern Denmark, Denmark

[b] Department of Informatics, University of Bergen, Norway

ARTICLE INFO

*Article history:*
Received 4 February 2009
Received in revised form 27 November 2009
Accepted 19 January 2010
Communicated by A. Fiat

*Keywords:*
Online algorithms
Edge coloring
Competitive ratio
Relative worst-order ratio
Next-Fit
First-Fit

ABSTRACT

We study the performance of the algorithms First-Fit and Next-Fit for two online edge coloring problems. In the min-coloring problem, all edges must be colored using as few colors as possible. In the max-coloring problem, a fixed number of colors is given, and as many edges as possible should be colored. Previous analysis using the competitive ratio has not separated the performance of First-Fit and Next-Fit, but intuition suggests that First-Fit should be better than Next-Fit. We compare First-Fit and Next-Fit using the relative worst-order ratio, and show that First-Fit is better than Next-Fit for the min-coloring problem. For the max-coloring problem, we show that First-Fit and Next-Fit are not strictly comparable, i.e., there are graphs for which First-Fit is significantly better than Next-Fit and graphs where Next-Fit is slightly better than First-Fit.



## 1. Introduction

In edge coloring, the edges of a graph must be colored such that no two adjacent edges receive the same color. This paper studies two variants of online edge coloring, min-coloring and max-coloring. For both problems, the algorithm is given the edges of a graph one by one, each one specified by its endpoints.

In the *min-coloring* problem, each edge must be colored before the next edge is received, and once an edge has been colored, its color cannot be changed. The aim is to color all edges *using as few colors as possible*.

For the *max-coloring* problem, a limited number $k$ of colors is given. Each edge must be either colored or rejected before the next edge arrives. Once an edge has been colored, its color cannot be changed and it cannot be rejected. Similarly, once an edge has been rejected, it cannot be colored. In this problem, the aim is to *color as many edges as possible*.

For both problems we study the following two algorithms. *First-Fit* is the natural greedy algorithm which colors each edge using the lowest possible color. *Next-Fit* uses the colors in a cyclic order. It colors the first edge with the color 1 and keeps track of the last used color $c_{\text{last}}$. For the max-coloring problem, 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 $v$. For the min-coloring problem, it only cycles through the set of colors that it has used so far, and a new color is only selected if all colors used so far are present on edges incident to either $u$ or $v$.

Both algorithms are members of more general families of algorithms. For the max-coloring problem, we define the class of *fair* algorithms that never reject an edge, unless all $k$ colors are already represented at adjacent edges. For the min-coloring problem, we define the class of *Any-Fit* algorithms that do not take a new color into use, unless necessary. In [1], the term

* Corresponding author. Tel.: +45 6550 2499; fax: +45 6550 2325.
*E-mail addresses:* ehmsen@imada.sdu.dk (M.R. Ehmsen), lenem@imada.sdu.dk (L.M. Favrholdt), svalle@imada.sdu.dk (J.S. Kohrt), rodica.mihai@ii.uib.no (R. Mihai).

0304-3975/$ – see front matter 
doi:10.1016/j.tcs.2010.01.015

"parsimonious" was used for this type of algorithm. In this paper, we use the term "Any-Fit" because of the similarity with the class of Any-Fit algorithms for bin packing.

The standard quality measure for online algorithms is the competitive ratio. Roughly speaking, the *competitive ratio* of an online algorithm A is the worst-case ratio of the performance of A to the performance of an optimal offline algorithm over all possible request sequences [2,3]. In the following, note that online algorithms for the min-problem have competitive ratios greater than 1, and online algorithms for the max-problem have competitive ratios less than 1.

The min-problem has previously been studied in [4], where the main result is that for any online algorithm A there exists a graph $G$ with maximum vertex degree $\Delta$, such that $G$ can be $\Delta$-colored, but A uses $2\Delta - 1$ colors. On the other hand, since no edge is adjacent to more than $2\Delta - 2$ other edges, no Any-Fit algorithm will use more than $2\Delta - 1$ colors. Hence, the result in [4] implies that all Any-Fit algorithms have competitive ratio $2 - \frac{1}{\Delta}$. Thus, the competitive ratio does not distinguish between First-Fit and Next-Fit.

The max-problem was studied in [1]. For $k$-colorable graphs, First-Fit and Next-Fit have very similar competitive ratios of $1/2$ and $k/(2k-1)$. For general graphs, there is an upper bound on the competitive ratio of First-Fit of $\frac{2}{9}(\sqrt{10}-1) \approx 0.48$, and the competitive ratio of Next-Fit exactly matches the general lower bound for fair algorithms of $2\sqrt{3}-3 \approx 0.46$. No fair algorithm can be better than 0.5-competitive, so the competitive ratio cannot vary much between fair algorithms. Moreover, there is a general upper bound (even for randomized algorithms) of $4/7 \approx 0.57$.

General intuition suggests that First-Fit should be better than Next-Fit, and thus comes the motivation to study the performance of the two algorithms using some other measure than the competitive ratio. There are previous problems, such as paging [5,6], bin packing [7], scheduling [8], and seat reservation [9], where the relative worst-order ratio was successfully applied and separated algorithms that the competitive ratio could not. The relative worst-order ratio is a quality measure that compares two online algorithms directly, without an indirect comparison via an optimal offline algorithm. Thus, the relative worst-order ratio in many cases gives more detailed information than the competitive ratio.

Previous results on the relative worst-order ratio indicate that, when separating algorithms, the measure favors the algorithm which is better according to intuition and/or practical results [5–7,9,8]. However, the concrete value of the ratio is usually of less importance. This is not different from other measures; as an example, the competitive ratios of paging algorithms are much larger than the ratios observed in practice [10], and the actual values are usually only used to compare algorithms. Thus, in this paper, we focus on separating algorithms, not on finding exact ratios. However, observe that separation results with the relative worst-order ratio are stronger results than with the competitive ratio: If algorithm A is better than algorithm B according to the relative worst-order ratio, then A is always at least as good as B, up to permutations of the input sequence, and there is at least one sequence where A is better than B (even without considering permutations[1]). This is in contrast to the competitive ratio: even if A has a better competitive ratio than B, B can be better than A in most cases.

For the min-problem, we prove that the two algorithms are comparable, and First-Fit is $2 - \frac{1}{\Delta}$ times better than Next-Fit. For the max-problem, surprisingly, we conclude that First-Fit and Next-Fit are not comparable using the relative worst-order ratio, i.e., there are graphs for which First-Fit is significantly better than Next-Fit and graphs where Next-Fit is slightly better than First-Fit.

## 2. The relative worst order ratio

The relative worst-order ratio was first introduced in [7] in an effort to combine the desirable properties of the max/max ratio [11] and the random-order ratio [12]. The measure was later refined in [5]. We describe the measure using the terminology of the coloring problems. Let $E$ be a sequence of $n$ edges. If $\sigma$ is a permutation on $n$ elements, then $\sigma(E)$ denotes $E$ permuted by $\sigma$.

For the max-coloring problem, $\mathrm{A}(E)$ is the number of edges colored by algorithm A, and

$$\mathrm{A_W}(E) = \min_{\sigma}\{\mathrm{A}(\sigma(E))\}.$$

For the min-coloring problem, $\mathrm{A}(E)$ is the number of colors used by A, and

$$\mathrm{A_W}(E) = \max_{\sigma}\{\mathrm{A}(\sigma(E))\}.$$

Thus, in both cases, $\mathrm{A_W}(E)$ is the performance of A on a worst possible permutation of $E$.

**Definition 1.** For any pair of algorithms A and B, we define

$$\begin{aligned} c_u(A, B) &= \inf\{c \mid \exists b\colon \forall E\colon\ \mathrm{A_W}(E) \le cB_W(E) + b\} \quad \text{and} \\ c_l(A, B) &= \sup\{c \mid \exists b\colon \forall E\colon\ \mathrm{A_W}(E) \ge cB_W(E) - b\}\,. \end{aligned}$$

[1] If A is better than B according to the relative worst-order ratio, there is an input sequence $I$ where A's performance on its worst permutation of $I$ is better than B's performance on its worst permutation of $I$. Hence, A is better than B on B's worst permutation of $I$.

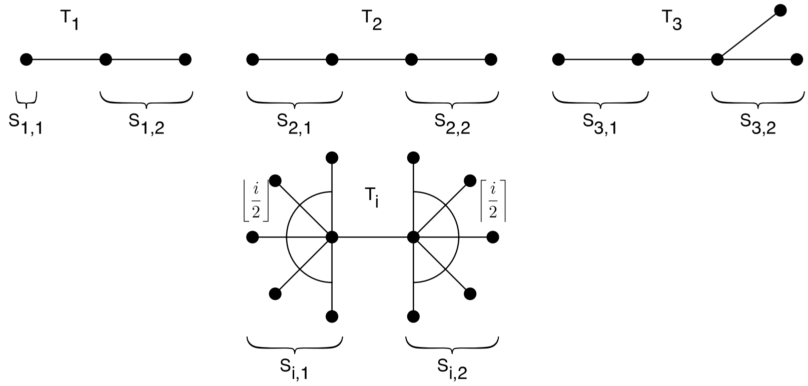


**Fig. 1.** The graph $G$ used in the proof of Lemma 4.

If $c_l(A, B) \geq 1$ or $c_u(A, B) \leq 1$, the algorithms are said to be *comparable* and the *relative worst-order ratio* $\mathrm{WR}_{A,B}$ of algorithm $A$ to algorithm $B$ is defined. Otherwise, $\mathrm{WR}_{A,B}$ is undefined.

$$\text{If } c_u(A, B) \leq 1, \text{ then } \mathrm{WR}_{A,B} = c_l(A, B), \quad \text{and}$$
$$\text{if } c_l(A, B) \geq 1, \text{ then } \mathrm{WR}_{A,B} = c_u(A, B)\,.$$

Intuitively, $c_l$ and $c_u$ can be thought of as tight lower and upper bounds, respectively, on the performance of A relative to B.

## 3. Min-coloring problem

We first study the min-coloring problem, where all edges of a graph must be colored using as few colors as possible.

As we observed in the introduction, the competitive ratio cannot distinguish between First-Fit and Next-Fit. However, with the relative worst-order ratio, we get the result that First-Fit is better than Next-Fit.

**Theorem 2.** *For the min-coloring problem*

$$\mathrm{WR}_{\mathrm{NF,FF}} = 2 - \frac{1}{\Delta}$$

*where $\Delta$ is the maximum degree of the graph.*

The theorem follows directly from Lemma 3, 4 and the following general observation. In the introduction, we showed that no Any-Fit algorithm will use more than $2\Delta - 1$ colors. On the other hand, $\Delta$ is clearly a lower bound on the number of colors that must be used. Hence, First-Fit and Next-Fit cannot be more than a factor of $2 - \frac{1}{\Delta}$ apart in the number of colors they use, independently of the given order of the edges.

We first show that, according to the relative worst-order ratio, First-Fit is a best possible Any-Fit algorithm.

**Lemma 3.** *Given any graph $G$ with edges $E$ and any Any-Fit algorithm* A, $\mathrm{A_W}(E) \geq \mathrm{FF_W}(E)$.

**Proof.** For any ordering $\sigma(E)$ of the edges, we construct an ordering $\sigma'(E)$ of the edges so that A does the same coloring on $\sigma'(E)$ as First-Fit does on $\sigma(E)$.

Assume that First-Fit uses $k$ colors and let $C_i$ denote the set of edges that First-Fit colors with color $i$. The ordering of the edges given to A consists of all the edges from $C_1$, then from $C_2$ and further till $C_k$. The edges in each set are given in an arbitrary order. By the First-Fit policy, each edge in $C_i$ is adjacent to edges of $C_1, \ldots, C_{i-1}$. Hence, by the Any-Fit property, A produces exactly the First-Fit coloring.

This proves that, for any ordering $\sigma(E)$ of the edges (in particular, the worst ordering for First-Fit), we can construct an ordering $\sigma'(E)$ of the edges such that $\mathrm{A}(\sigma'(E)) = \mathrm{FF}(\sigma(E))$. The result follows. □

**Lemma 4.** *For any integer $\Delta > 0$, there exists a graph with edges $E$ and maximum degree $\Delta$ such that* $\mathrm{NF_W}(E) \geq (2 - \frac{1}{\Delta})\, \mathrm{FF_W}(E)$.

**Proof.** Let $G$ be an unconnected graph with components $T_1, T_2, \ldots, T_{2\Delta-2}$, where $T_i$ consists of two stars $S_{i,1}$ and $S_{i,2}$ with an additional edge connecting their centers; see Fig. 1. The star $S_{i,1}$ has $\lfloor \frac{i}{2} \rfloor$ edges and $S_{i,2}$ has $\lceil \frac{i}{2} \rceil$ edges. Thus, $T_i$ has $i + 1$ edges in total, and its maximum degree is $\lceil \frac{i}{2} \rceil + 1$. Hence, the maximum degree of $G$ is $\lceil \frac{2\Delta-2}{2} \rceil + 1 = \Delta$.

Consider any ordering of the edges of $G$ and the resulting First-Fit coloring. In $T_i$, each star edge is adjacent to at most $\lceil \frac{i}{2} \rceil$ other edges. Thus, no star edge in $T_i$ is colored with a color larger than $\lceil \frac{i}{2} \rceil + 1$. Furthermore, a star edge in $T_i$ can be colored with color $\lceil \frac{i}{2} \rceil + 1$, only if the edge connecting $S_{i,1}$ and $S_{i,2}$ has already been colored with one of the colors $1, 2, \ldots, \lceil \frac{i}{2} \rceil$. Hence, when First-Fit colors $T_i$, it uses at most $\lceil \frac{i}{2} \rceil + 1$ colors. It follows that the largest color used by First-Fit is no more than $\lceil \frac{2\Delta-2}{2} \rceil + 1 = \Delta$.

We now show that Next-Fit uses $2\Delta - 1$ colors, if the edges are given in the following order: The components are given in the order $T_1, T_2, \ldots, T_{2\Delta-2}$. In each of the $T_i$'s, the star edges are given first, followed by the edge connecting the two stars. The two edges of $T_1$ will be colored with colors 1 and 2. It follows by induction that, for $2 \le i \le 2\Delta - 2$, the star edges of $T_i$ will be colored with colors $1, 2, \ldots, i$ and the connecting edge will receive color $i + 1$. Thus, $T_{2\Delta-2}$ will be colored with the colors $1, 2, \ldots, 2\Delta - 1$, and the result follows. □

## 4. Max-coloring problem

In this section, we study the max-coloring problem, where a limited number $k$ of colors are given, and as many edges as possible should be colored. We first describe a bipartite graph with edges $E$, such that

$$\mathrm{FF_W}(E) \ge \frac{9}{8} \cdot \mathrm{NF_W}(E).$$

Then, we describe a family of graphs with edge set $E_n$ such that

$$\mathrm{NF_W}(E_n) = \left(1 + \Omega\left(\frac{1}{k^2}\right)\right) \cdot \mathrm{FF_W}(E_n).$$

Thus, the two algorithms are not comparable.

### 4.1. First-Fit can be better than Next-Fit

Let $B_{k,k} = (X, Y, E)$ be a complete bipartite graph with $|X| = |Y| = k$. For simplicity, we assume that 4 evenly divides $k$. For other values of $k$, we get similar results, but the calculations are more involved.

We denote by $C_i$ the edges that First-Fit colors with color $i$.

**Proposition 5.** *For any ordering of the edges of $B_{k,k}$,*

$$|C_i| \ge k - i + 1, \quad i = 1, \ldots, k.$$

**Proof.** Assume that color $i$ has been used $j \le k - i$ times. The induced subgraph containing all vertices *not* adjacent to an edge colored with color $i$ is the complete bipartite graph $B_{k-j,k-j}$, where $k - j \ge i$. This subgraph cannot be colored with the colors $1, \ldots, i - 1$ only, and since this is a First-Fit coloring, the color $i$ is going to be used. Thus, at least one more edge will be colored with color $i$. □

**Proposition 6.** *If* First-Fit *colors at most $\frac{9}{16}k^2$ edges of $B_{k,k}$, then*

$$|C_i| \ge \frac{7k^2}{16(2k - 1 - i)}, \quad i = 1, \ldots, k.$$

**Proof.** If First-Fit colors at most $9k^2/16$ edges, then it rejects at least $7k^2/16$ edges. Each rejected edge is adjacent to at least one edge of each color $i = 1, \ldots, k$. Each edge colored with color $i$ has $2k - 2$ neighbor edges. Among those, at least $i - 1$ edges are already colored, since each edge colored with $i$ is adjacent to all lower colors $1, 2, \ldots, i-1$. Thus, at most $2k-1-i$ edges can be rejected for each edge colored with $i$. Hence, for First-Fit to reject $7k^2/16$ edges, it has to use color $i$ at least $7k^2/(16(2k - 1 - i))$ times. □

**Lemma 7.** *Given any ordering of the edges of $B_{k,k}$,* First-Fit *colors more than $\frac{9}{16}k^2$ edges.*

**Proof.** The number of edges colored by First-Fit is $\sum_{i=1}^{k} |C_i|$. We assume for the sake of contradiction that First-Fit colors at most $9k^2/16$ edges of $B_{k,k}$. Using Propositions 5 and 6, we get

$$\begin{aligned}
\sum_{i=1}^{k} |C_i| &\ge \sum_{i=1}^{3k/4} (k - i + 1) + \sum_{i=3k/4+1}^{k} \frac{7k^2}{16(2k - 1 - i)} \\
&= \sum_{i=k/4+1}^{k} i + \frac{7}{16}k^2 \sum_{i=k-1}^{5k/4-2} \frac{1}{i} \\
&> \frac{15}{32}k^2 + \frac{7}{16}k^2 \ln\left(\frac{k + k/4 - 2}{k - 2}\right) \\
&> \frac{15}{32}k^2 + \frac{7}{16}k^2 \ln(1 + 1/4) \\
&> \frac{15}{32}k^2 + \frac{7}{16}k^2 \frac{3}{14} \\
&= \frac{9}{16}k^2,
\end{aligned}$$

which is a contradiction. Thus, First-Fit colors more than $\frac{9k^2}{16}$ edges. □

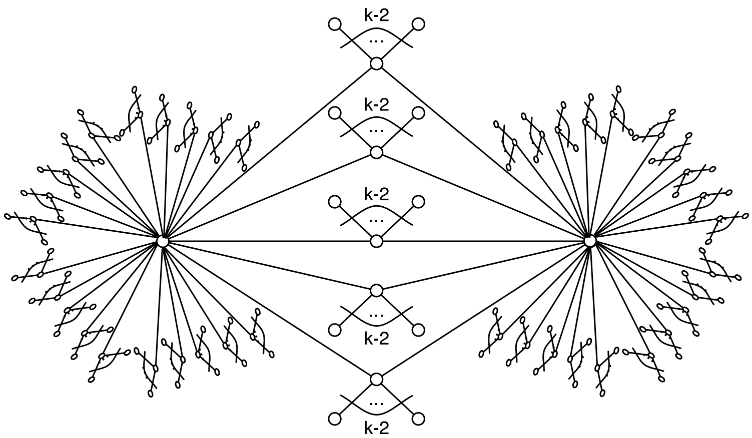


**Fig. 2.** Two superstars, for $k = 25$, connected through a link of five outer stars.

The inequalities in the proof of Lemma 7 are not tight. However, we lose less than 0.02 in the factor separating First-Fit and Next-Fit. Hence, if one wants to separate First-Fit and Next-Fit by a significantly larger ratio, more involved arguments or a different graph are needed.

**Lemma 8.** *Given the worst ordering of the edges of $B_{k,k}$,* Next-Fit *colors at most $k^2/2$ edges.*

**Proof.** We partition the vertex sets $X, Y$ into equal-sized sets $X_1, X_2, Y_1, Y_2$. The induced subgraphs $H_1$ and $H_2$ with vertex sets $X_1, Y_1$ and $X_2, Y_2$, respectively, are complete bipartite graphs.

Clearly, $H_1$ (and $H_2$) can be colored with $k/2$ colors, such that each of the $k/2$ colors is present at each vertex in the subgraph. Since, in such a coloring, each of the $k/2$ colors is used the same number of times, we can give the edges of $H_1$ and $H_2$ alternately such that Next-Fit colors the edges of $H_1$ using colors $1, 2, \ldots, k/2$ and the edges of $H_2$ with colors $k/2 + 1, \ldots, k$. After that, Next-Fit cannot color any of the $k^2/2$ edges between $X_1$ and $Y_2$ and between $X_2$ and $Y_1$. Thus, for such an ordering of the edges, Next-Fit colors at most $k^2/2$ edges of $B_{k,k}$. □

Combining Lemmas 7 and 8, we arrive at the following.

**Corollary 9.** *Given the graph $B_{k,k} = (X, Y, E)$,*

$$\mathrm{FF_W}(E) \geq \frac{9}{8}\,\mathrm{NF_W}(E).$$

*4.2. Next-Fit can be slightly better than First-Fit*

In this section, we prove that there exists a family of graphs where Next-Fit is $1 + \Omega(\frac{1}{k^2})$ times better than First-Fit. We first define the building blocks of the graph family.

**Definition 10.** For any given integer $k \geq 25$, a *superstar $S_k$* is a graph consisting of an *inner star* with $k$ edges, each incident to the center of an *outer star* with $k - 2$ edges of its own.

A *superstar graph* is a graph consisting of superstars. Each pair of superstars in the graph may share a number of outer stars. The set of outer stars shared by a pair of superstars is called the *link* between them. All outer stars are contained in a link. Each link contains at least five outer stars, and each superstar has links to between five and seven other superstars. See Fig. 2 for an incomplete example.

Clearly, fair algorithms never reject outer star edges. However, if all outer stars are colored using the same $k - 2$ colors, at least $k - 2$ edges of each inner star will be rejected. This leads to the following lemma.

**Lemma 11.** *Let $G_{n,k}$ be a superstar graph with $n$ superstars. Then, on its worst ordering of the edges,* First-Fit *rejects at least $n(k - 2)$ edges.*

What remains to be shown is that there exists a family $G_{n,k}$ of superstar graphs, such that, on a worst ordering of the edges of $G_{n,k}$, Next-Fit rejects only $n(k - 2) - \Omega(n)$ edges.

**Proposition 12.** *Consider a superstar graph $G$ colored by a fair algorithm. Any superstar in $G$ has at most $k - 1$ edges rejected. If some superstar $S$ in $G$ has $k - 1$ edges rejected, then each of its neighbor superstars has at most $k - 4$ edges rejected.*

**Proof.** Clearly, outer star edges are not rejected, so we only need to consider the inner star edges. At least one inner star edge will be colored in each superstar, since each inner star edge is only adjacent to $k - 1$ edges that are not inner star edges in the same superstar. Thus, at most $k - 1$ edges are rejected from any superstar in the graph.

Assume that some superstar $S$ has $k-1$ inner star edges rejected. Each of these edges must be adjacent to $k$ colored edges. However, at most $k-1$ of these colored neighbor edges belong to $S$ ($k-2$ from the outer star, and the one colored inner edge of $S$). Hence, the $k$th colored neighbor edge must be an inner star edge in a neighboring superstar. Since each link contains at least five inner edges of $S$ and at most one of them is colored, this completes the proof. □

By Proposition 12, any pair of neighboring superstars has at most $2k-4$ rejected edges in total. A pair of neighboring superstars with $2k-4$ rejected edges in total is called a *bad pair*. Note that, in a bad pair, exactly $k-2$ edges are rejected in each superstar. A pair of neighboring superstars with at most $2k-5$ rejected edges in total is called a *good pair*. A superstar contained only in bad pairs is called a *bad superstar*. A superstar contained in at least one good pair is called a *good superstar*.

Counting the good superstars, the extra colored edge from a good pair is counted at most eight times: once for the superstar $S$ containing it and once for each of the at most seven neighbors of $S$. Thus, the following lemma follows directly from Proposition 12.

**Lemma 13.** *Consider a fair coloring of a superstar graph with n superstars. If there are m good superstars, then at most* $n(k-2)-\frac{m}{8}$ *edges are rejected.*

Consequently, we just need to show that we can connect our building blocks, the superstars, such that, for any ordering of the edges and the resulting Next-Fit coloring, there will be $\Omega(n)$ good superstars. Such a construction is described in the proof of Lemma 16. The proof of Lemma 16 uses Proposition 14 and Lemma 15 below.

The *majority coloring* of a superstar is the set of colors used on the majority of its outer stars, breaking ties arbitrarily. An outer star is *isolated*, if it is not adjacent to at least one colored inner star edge.

**Proposition 14.** *If two neighboring superstars have different majority colorings, one of them is a good superstar.*

**Proof.** We prove the proposition by contraposition. Assume that two superstars $S$ and $S'$ are both bad superstars. Then, by Proposition 12, $S$, $S'$, and their neighbors each have exactly $k-2$ edges rejected. Let $c_1$ and $c_2$ be the two colors used on inner star edges in $S$.

If $S$ has $m$ neighbors, the outer stars of $S$ are adjacent to at most $2m+2$ colored inner star edges. Thus $S$ has at least $k-2m-2$ isolated outer stars. Each of these outer stars must be colored with the $k-2$ colors different from $c_1$ and $c_2$. Hence, the isolated outer stars in $S$ are colored the same, and since $m \geq 5$ and $k \geq 5m$ that coloring is the majority coloring of $S$. The same is true for $S'$. Since $S$ and $S'$ each have exactly two colored edges and there are at least five edges in the link between them, they share at least one isolated outer star. This means that $S$ and $S'$ have the same majority coloring. □

**Lemma 15.** *Assume that* $k \geq 101$. *Consider a* Next-Fit *coloring of a superstar graph* $G_{n,k}$, $n \geq 6$. *Among the bad superstars, there are at most* $\frac{2}{3}n$ *superstars with the same majority coloring.*

**Proof.** Any subgraph of $G_{n,k}$ containing $x$ superstars has at least $x\frac{k}{2}$ outer stars. Thus, in any subgraph $H$ of $G_{n,k}$ consisting of $x$ bad superstars with the same majority coloring $\mathcal{M}$, there are at least $x\frac{k-16}{2} = x\left(\frac{k}{2}-8\right)$ isolated outer stars colored with $\mathcal{M}$. Each time Next-Fit has used the colors in $\mathcal{M}$ once, the two colors $c_1, c_2 \notin \mathcal{M}$ must be used once, before it will use the colors in $\mathcal{M}$ on isolated outer stars again. Thus, an upper bound on the number of times $c_1$ and $c_2$ are used in $G_{n,k}$ gives an upper bound on $x$.

Clearly, $c_1$ and $c_2$ are each used at most once on inner star edges in each superstar. Inside $H$, $c_1$ and $c_2$ are not used on isolated outer stars. Thus, since each bad superstar has at least $k-16$ isolated outer stars, $c_1$ and $c_2$ are used at most $17x$ times on superstars in $H$.

Outside $H$, $c_1$ and $c_2$ can each be used at most once per outer star, since using $c_1$ ($c_2$) on an inner star edge would prohibit the algorithm from using $c_1$ ($c_2$) on the adjacent outer star. Hence, since each superstar outside $H$ share each outer star with another superstar, the superstars outside $H$ can only contribute $(n-x)\frac{k}{2}$.

Thus, to create $x$ bad superstars with majority coloring $\mathcal{M}$, we must have

$$x\left(\frac{k}{2}-8\right)-1 \;\leq\; 17x+(n-x)\,\frac{k}{2}.$$

Solving for $x$, we obtain $x \leq \frac{2}{3}n$, since $k \geq 101$ and $n \geq 6$. □

**Lemma 16.** *For* $k \geq 101$, *there exists a family of superstar graphs* $G_{n,k}$ *where any ordering of the edges results in a* Next-Fit *coloring with* $\Omega(n)$ *good superstars.*

**Proof.** We use a result from expander graphs [13]. Using notation from [14], for any positive integer $m$, there exists an $\left(n=2m^2, 7, \frac{2-\sqrt{3}}{2}\right)$-expander, i.e., a 7-regular bipartite multigraph $G(X \cup Y, E)$ with $|X| = |Y| = \frac{n}{2}$, such that, for any $S \subseteq X$,

$$|\Gamma(S)| \geq \left(1+\frac{2-\sqrt{3}}{2}\left(1-\frac{2|S|}{n}\right)\right)|S|\,,$$

where $\Gamma(S)$ is the set of edges between $S$ and $\overline{S}$. The result also holds for any $S \subseteq Y$. The graph contains parallel edges, but each vertex has at least five neighbors. Replacing each set of parallel edges by one edge, we obtain a simple graph with the same $\Gamma$-function.

Now, we connect the superstars as in the simple expander graph. For any suitable $n$, let each vertex in the expander graph correspond to a superstar. Each edge in the expander graph corresponds to a link between the corresponding superstars. Thus, we obtain a superstar graph where each superstar has links to five, six, or seven other superstars.

Consider any ordering of the edges of this graph with $n$ superstars and the resulting Next-Fit coloring. If there are at least $\frac{1}{3}n$ good superstars, the result follows immediately. Thus, we consider the case where there are at least $\frac{2}{3}n$ bad superstars. By Lemma 15, no majority coloring occurs on more than $\frac{2}{3}n$ bad superstars. Among the bad superstars, let $S$ be the superstars with the most frequently occurring majority coloring. If $|S| < \frac{1}{3}n$, add the bad superstars with the most frequently occurring majority coloring among the superstars not in $S$. Continue doing this until $S$ reaches a size between $\frac{1}{3}n$ and $\frac{2}{3}n$. This is possible, since we consider the case where there are at least $\frac{2}{3}n$ bad superstars.

Define $S_X = S \cap X$ and $S_Y = S \cap Y$, and assume without loss of generality that $|S_X| \geq |S_Y|$. Note that $|S_X| \geq \frac{1}{2}|S| \geq \frac{1}{6}n$. We can bound the size of $\Gamma(S)$ from below by the following:

$$\begin{aligned} |\Gamma(S)| &\geq |\Gamma(S_X)| - |S_Y| \\ &\geq \frac{2-\sqrt{3}}{2}\left(1 - \frac{2|S_X|}{n}\right)|S_X| + (|S_X| - |S_Y|). \end{aligned} \tag{1}$$

We now have two cases depending on the size of $S_X$:

- $\frac{5}{12}n \leq |S_X| \leq \frac{1}{2}n$. Since $|S_X| + |S_Y| \leq \frac{2}{3}n$, we must have $|S_Y| \leq \frac{3}{12}n$. Thus, inequality (1) immediately yields a lower bound of $\frac{5}{12}n - \frac{3}{12}n = \frac{1}{6}n$, since $\frac{2-\sqrt{3}}{2}\left(1 - \frac{2|S_X|}{n}\right)$ is nonnegative.
- $|S_X| < \frac{5}{12}n$. Since $|S_X| - |S_Y| \geq 0$, inequality (1) gives a lower bound of $\left(\frac{2-\sqrt{3}}{2}\right)\frac{1}{6}|S_X| \geq \frac{2-\sqrt{3}}{144}n$.

Hence, in the coloring done by Next-Fit, in both cases we have $\Omega(n)$ links between $S$ and $\overline{S}$. By the construction of $S$, each superstar in $\overline{S}$ linked to a superstar $s$ in $S$ is a good superstar or has a different majority coloring than $s$. Thus, by Proposition 14, there are $\Omega(n)$ good superstars. □

**Corollary 17.** *There exist superstar graphs $G_{n,k}$ such that*

$$\mathrm{NF_W}(G_{n,k}) = \left(1 + \Omega\left(\frac{1}{k^2}\right)\right)\mathrm{FF_W}(G_{n,k}).$$

**Proof.** By Lemma 11, there is an ordering of the edges in any superstar graph with $n$ superstars, such that First-Fit rejects at least $n(k-2)$ edges.

By Lemmas 13 and 16, there are superstar graphs $G_{n,k}$ with $n$ superstars such that, for any ordering of the edges, Next-Fit rejects $n(k-2) - \Omega(n)$ edges. Hence, it follows that $\mathrm{NF_W}(G_{n,k}) = \left(1 + \Omega\left(\frac{1}{k^2}\right)\right)\mathrm{FF_W}(G_{n,k})$, since First-Fit colors $O(nk^2)$ edges. □

This, together with Corollary 9, immediately yields the following theorem.

**Theorem 18.** First-Fit *and* Next-Fit *are not comparable by the relative worst-order ratio.*

## 5. Conclusion

We have proven that, with the relative worst-order ratio, First-Fit is strictly better than Next-Fit for the min-coloring problem. This is in contrast to the competitive ratio which is the same for all Any-Fit algorithms, a class of algorithms to which both First-Fit and Next-Fit belong.

For the max-coloring problem, the answer is not as clear. With the relative worst-order ratio, there are graphs where First-Fit does significantly better than Next-Fit and graphs where Next-Fit does slightly better than First-Fit. This is somewhat in keeping with an earlier result saying that the two algorithms can hardly be distinguished by their competitive ratios.

However, for the max-coloring problem, the two algorithms may be *resource-asymptotically* comparable [5]. Roughly speaking, this means that, as $k$ tends to infinity, the algorithms "become comparable". This is left as an open problem. Note that if one were to prove that the algorithms are *not* asymptotically comparable, another construction than the superstar graphs would be required: even if Next-Fit colored all edges of a superstar graph, it would color only $1 + \Theta(\frac{1}{k})$ times as many edges as First-Fit.

## Acknowledgement

The first three authors were supported in part by the Danish Agency for Science, Technology and Innovation (FNU).

## References

[1] L.M. Favrholdt, M.N. Nielsen, On-line edge coloring with a fixed number of colors, Algorithmica 35 (2) (2003) 176–191.
[2] D.D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Communications of the ACM 28 (2) (1985) 202–208.
[3] A.R. Karlin, M.S. Manasse, L. Rudolph, D.D. Sleator, Competitive snoopy caching, Algorithmica 3 (1988) 79–119.

[4] A. Bar-Noy, R. Motwani, J. Naor, The greedy algorithm is optimal for on-line edge coloring, Information Processing Letters 44 (5) (1992) 251–253.
[5] J. Boyar, L.M. Favrholdt, K.S. Larsen, The relative worst-order ratio applied to paging, Journal of Computer and System Sciences 73 (2007) 818–843.
[6] J. Boyar, M.R. Ehmsen, K.S. Larsen, Theoretical evidence for the superiority of LRU-2 over LRU for the paging problem, in: Approximation and Online Algorithms, 2006, pp. 95–107.
[7] J. Boyar, L.M. Favrholdt, The relative worst order ratio for on-line algorithms, ACM Transactions on Algorithms 3 (2) (2007) article 22, 24 pages.
[8] L. Epstein, L.M. Favrholdt, J. Kohrt, Separating online scheduling algorithms with the relative worst order ratio, Journal of Combinatorial Optimization 12 (4) (2006) 363–386.
[9] J. Boyar, P. Medvedev, The relative worst order ratio applied to seat reservation, ACM Transactions on Algorithms 4 (4) (2008) article 48, 22 pages.
[10] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.
[11] S. Ben-David, A. Borodin, A new measure for the study of on-line algorithms, Algorithmica 11 (1) (1994) 73–91.
[12] C. Kenyon, Best-Fit bin-packing with random order, in: 7th Annual ACM-SIAM Symposium on Discrete Algorithms, 1996, pp. 359–364.
[13] O. Gabber, Z. Galil, Explicit constructions of linear-sized superconcentrators, Journal of Computer and System Sciences 22 (1981) 407–420.
[14] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.