Edge-Coloring Sparse Graphs with Δ Colors in Quasilinear Time

Łukasz Kowalik
Institute of Informatics, University of Warsaw, Poland

Abstract

In this paper we show that every graph G of bounded maximum average degree $\operatorname{mad}(G)$ and with maximum degree Δ can be edge-colored using the optimal number of Δ colors in quasilinear time, whenever $\Delta \geq 2\operatorname{mad}(G)$. The maximum average degree is within a multiplicative constant of other popular graph sparsity parameters like arboricity, degeneracy or maximum density. Our algorithm extends previous results of Chrobak and Nishizeki [10] and Bhattacharya, Costa, Panski and Solomon [5].

2012 ACM Subject Classification Mathematics of computing → Graph coloring; Theory of computation → Graph algorithms analysis

Keywords and phrases edge coloring, algorithm, sparse, graph, quasilinear

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.81

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2401.13839 [22]

Funding This work is a part of project BOBR that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 948057).

Acknowledgements The author wishes to thank Bartłomiej Bosek, Jadwiga Czyżewska, Konrad Majewski and Anna Zych-Pawlewicz for helpful discussions on related topics. The author is also grateful to the reviewers for helpful comments.

1 Introduction

Algorithms for edge-coloring is a classic research topic which has become active again recently. Let us recall that edge-coloring of graph G is a function $\pi : E(G) \to \mathbb{N}$ which assigns different values (called *colors*) to incident edges. By *k-edge-coloring* we mean an edge coloring which uses at most k colors. The minimum number of colors that suffice to color graph G is called the *chromatic index* of G and denoted $\chi'(G)$. For a maximum degree of G denoted $\Delta(G)$ it is clear that $\chi'(G) \geq \Delta(G)$, while the classic theorem of Vizing states that $\chi'(G) \leq \Delta(G) + 1$. For general graphs, determining whether $\chi'(G) = \Delta(G)$ is NP-complete, as shown by Holyer [20].

1.1 Algorithms for general graphs

Vizing's proof can be easily transformed to an algorithm for edge-coloring graphs in $\Delta + 1$ colors running in time $O(nm)$, where n and m denote the number of vertices and edges, respectively, of the input graph (we use this notation throughout the whole paper). Gabow, Nishizeki, Kariv, Leven, and Terada [18] were the first to show an algorithmic progress: they designed two algorithms, one running in time $O(m\sqrt{n \log n})$ and another in time $O(m\Delta \log n)$. Both these algorithms saw improvements recently: Sinnamon [24] obtained deterministic time $O(m\sqrt{n})$, Bhattacharya, Carmon, Costa, Solomon and Zhang [4] have randomized time $O(mn^{1/3})$, while Bernshteyn and Dhawan [3] randomized time $\Delta^{O(1)}m$. Another direction of research is improving the running time at the price of an increased number of colors: very

© Łukasz Kowalik;
licensed under Creative Commons License CC-BY 4.0
32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 81; pp. 81:1–81:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Summary of state-of-the-art edge-coloring algorithms. Note that $\alpha = \Theta(\mathrm{mad}(G))$. The dice symbol ⚅ denotes a randomized algorithm.

Number of colors	Graph class	Time	Reference
$(1+\epsilon)\Delta$	general	$O(m\epsilon^{-1}\log n)$	Elkin and Khuzman [16]
$(1+\epsilon)\Delta$	general	$O(m\log(1/\epsilon))$ ⚅	Assadi [1]
$\Delta + O(\log n)$	general	$O(m\log\Delta)$ ⚅	Assadi [1]
$\Delta + 2\alpha - 2$	arboricity α	$O(m\log\Delta)$	Christiansen et al. [8]
$\Delta + 1$	general	$O(m\sqrt{n})$	Sinnamon [24]
$\Delta + 1$	general	$O(mn^{1/3})$ ⚅	Bhattacharya et al. [4]
$\Delta + 1$	general	$O(m\Delta\log n)$	Gabow et al. [18]
$\Delta + 1$	general	$O(m\Delta^{18})$ ⚅	Bernshteyn and Dhawan [3]
$\Delta + 1$	general	$O(n^2\log n)$ ⚅	Assadi [1]
$\Delta + 1$	arboricity α	$O(m\alpha\log n)$ ⚅	Bhattacharya et al. [5]
$\Delta + 1$	arboricity α	$O(m\alpha^7\log n)$	**This work**
$\Delta + 1$	arboricity α	$\tilde{O}(m\sqrt{n}\,\frac{\alpha}{\Delta})$ ⚅	Bhattacharya et al. [5]
$\chi'(G)$	bounded treewidth	$O(n)$	Zhou et al. [31]
Δ	bipartite	$O(m\log\Delta)$	Cole, Ost and Shirra [13]
Δ	bounded genus, $\Delta \geq 19$	$O(n)$	Chrobak and Yung [11]
Δ	bounded genus, $\Delta \geq 9$	$O(n\log n)$	Chrobak and Nishizeki [10]
Δ	planar, $\Delta \geq 9$	$O(n)$	Cole and Kowalik [12]
Δ	$\Delta \geq 2\,\mathrm{mad}(G)$	$O(m\alpha^3\log n)$ ⚅	**This work**
Δ	$\Delta \geq 2\,\mathrm{mad}(G)$	$O(m\alpha^7\log n)$	**This work**

recently, Elkin and Khuzman [16] presented a deterministic algorithm that uses at most $(1+\epsilon)\Delta$ colors for any $\epsilon \geq \frac{1}{\Delta}$ and works in time $O(m\epsilon^{-1}\log n)$, while Assadi [1] was able to get the improved time of $O(m\log(1/\epsilon))$, at the price of randomization; see also Duan et al. [15] for an earlier result of this kind.

1.2 Algorithms for graph classes

Restricting the input graph may allow for algorithms with improved performance. By the classical Kőnig's Theorem, for bipartite graphs only Δ colors suffice, and Cole, Ost and Shirra [13] provided an $O(m\log\Delta)$-time algorithm. Zhou, Nakano and Nishizeki [31] gave an algorithm that uses $\chi'(G)$ colors and runs in linear time for graphs of bounded treewidth. Chrobak and Yung [11] gave a linear time algorithm that uses only $\max\{\Delta, 19\}$ colors and works for graphs of bounded genus. Graphs from this class can be also colored using only $\max\{\Delta, 9\}$ colors by an algorithm of Chrobak and Nishizeki [10] that runs in time $O(n\log n)$. For the special case of planar graphs, Cole and Kowalik [12] improved that to linear time.

1.3 Uniformly sparse graphs and our result

Maximum average degree is defined as $\mathrm{mad}(G) = \max_H \mathrm{ad}(H)$, where the maximum is over all nonempty subgraphs of G and $\mathrm{ad}(H) = \frac{1}{s}\sum_{v\in V(H)} d_H(v)$ is the average degree of H. Hence, when $\mathrm{mad}(G)$ is low, graph G is uniformly sparse. It is easy to see that $\mathrm{mad}(G)$ equals exactly twice the *maximum density* of G, i.e., $\max_S |E(S)|/|V(S)|$, which is close to the *arboricity* defined as $\alpha(G) = \max_{S,|V(S)|>1}\lceil |E(S)|/(|V(S)|-1)\rceil$. Indeed, all these three parameters (and also the degeneracy), are within a small constant of each other (see e.g. [21]). In particular, when one of them is bounded by a constant, so are all the others. Moreover, planarity, bounded treewidth or bounded genus all imply bounded maximum average degree, arboricity, etc.

Very recently, Bhattacharya, Costa, Panski and Solomon [5] obtained two algorithms that use $\Delta + 1$ colors and work in randomized time $O(m\sqrt{n}\Delta/\alpha(G))$ and $O(m\alpha(G)\log n)$. In particular, the latter one is quasilinear for bounded arboricity, and both algorithms improve over the results of Gabow et al. [18] when $\Delta(G) = \omega(\alpha(G))$. Another recent result is an algorithm of Christiansen, Rotenberg and Vlieghe [8] that uses $\Delta + 2\alpha(G) - 2$ colors and runs in deterministic time $O(m \log \Delta)$.

In this work we continue this line of research with the goal of *improving the number of colors used*. Note that the quasilinear algorithm of Chrobak and Nishizeki [10] uses only Δ colors for the special case of graph G of bounded genus g and such that $\Delta \geq 9$. Since then $\mathrm{mad}(G) \leq 6 + O(g/n)$ it is natural to ask if it is possible to generalize it to any class of uniformly sparse graphs. We answer this question in the affirmative by showing the following main result.

▶ **Theorem 1.** *Every graph G with n vertices and m edges such that $\Delta(G) \geq 2\,\mathrm{mad}(G)$ can be $\Delta(G)$-edge-colored*

1. *by a randomized algorithm running in time $O(m\,\mathrm{mad}(G)^3 \log n)$ in the expectation and with high probability,*
2. *by a deterministic algorithm in time $O(m\,\mathrm{mad}(G)^7 \log n)$.*

In particular, whenever $\mathrm{mad}(G) = \log^{O(1)} n$, the algorithms work in quasilinear time. Here, by *high probability* we mean probability at least $1 - n^{-k}$, for an arbitrary constant $k > 0$. Note that as a corollary we get also the first *deterministic* quasi-linear time algorithm for $(\Delta+1)$-edge-coloring graphs of bounded $\mathrm{mad}(G)$ (or arboricity, etc.), thus derandomizing the result of Bhattacharya et al. [5]. Indeed, if $\Delta(G) \geq 2\,\mathrm{mad}(G)$ we apply our result, and otherwise just apply the $O(m\Delta \log n)$-time algorithm of Gabow et al. [18].

Let us comment on the assumption $\Delta(G) \geq 2\,\mathrm{mad}(G)$. Clearly, some kind of assumption is needed because there are sparse graphs of chromatic index $\Delta(G) + 1$. Vizing [28] (see also [25]) conjectured[1] that for $\chi'(G) = \Delta(G)$ it suffices that $\mathrm{mad}(G) \leq \Delta - 1$. While this conjecture is still open, there has been a substantial progress. Fiorini [17], Haile [19] and Sanders and Zhao [23] obtained sufficient conditions of the form $\mathrm{mad}(G) \leq \frac{1}{2}\Delta + o(\Delta)$. Next, it was improved to $\mathrm{mad}(G) < \frac{2}{3}(\Delta + 1)$ by Woodall [29, 30], and further to $\mathrm{mad}(G) < \min\{\frac{3}{4}\Delta - 2, 0.738\Delta - 1.153\}$ by Cao, Chen, Jiang, Liu and Lu [7]. All these results are in fact constructive and correspond to polynomial-time algorithms (but *not* quasilinear[2]). However, while the proof of Sanders and Zhao [23] relies on a simple procedure for extending a partial coloring called Vizing Adjacency Lemma (VAL, see Theorem 2 below), the later proofs result in considerably more complicated algorithms. In fact we show that the constant 2 is optimal if we restrict ourselves to algorithms that color the graph edge by edge using VAL (see Theorem 7 for a formal statement). Since we prefer to keep this article short, we leave the task of improving the assumption as an open problem.

[1] The original conjecture states that in every critial graph (in a sense, a minimal graph with chromatic index $\Delta + 1$) we have $|E(G)| \geq \frac{1}{2}((\Delta(G) - 1)|V(G)| + 3)$.

[2] The algorithm corresponding to the proof of Sanders and Zhao [23] can be implemented in time $O(nm)$; for the other the worst time complexity is at least that large.

1.4 Sketch of our approach

Let D be the number of available colors. Let us recall the following classical result (usually stated in terms of critical graphs). Here, for a statement P the expression $[P]$ equals 1 if P holds and 0 otherwise.

▶ **Theorem 2** (Vizing Adjacency Lemma, VAL [27]). *Let G be a simple graph and let $e = xy$ be an edge such that x has at most $D - d(y) + [d(y) = D]$ neighbors of degree D. Then any partial D-edge-coloring of G which colors a subset E_c of edges of G, $e \notin E_c$, can be extended to a partial D-edge-coloring that colors $E_c \cup \{e\}$.*

In this work, an edge xy that satisfies the assumption of VAL is called (D, x)*-weak*, or simply x*-weak* when D is known. Moreover, xy is *weak* if it is x-weak or y-weak.

Note that for $D = \Delta(G) + 1$, every edge is weak, just because then every vertex has no neighbors of degree D. Hence, in a simple implementation of standard proofs of Vizing Theorem, we color the input graph G edge by edge, using the algorithm originating from the Vizing's proof of VAL. This algorithm uses so-called fans and alternating paths. Given an edge $e = xy$, an e-fan is roughly a certain sequence of neighbors of x (equivalently, incident edges) – a precise definition will be given later. An *alternating path* is just a maximal path with edges colored alternately in two colors. Of course the algorithm behind VAL may need to recolor some of the already colored edges. However, it turns out that it always suffices to recolor some edges of an xy-fan and a single alternating path P_{xy}.

When we aim at a quasilinear algorithm, even for $D = \Delta(G)+1$, the approach above poses the following problem: even if coloring edge xy takes just $O(1)$ time per a recolored edge, our bound on the number of those edges is $\deg(x) + |P_{xy}|$, and the sum $\sum_{xy \in E(G)}(\deg(x) + |P_{xy}|)$ can be as large as $\Theta(nm)$, for $n := |V(G)|$, $m := |E(G)|$. The problem with long alternating paths has been resolved in an elegant way by Sinnamon [24]: he shows, roughly, that if we pick a *random* uncolored edge xy out of all ℓ uncolored edges, then the probability that a fixed alternating path is chosen is $O(\frac{1}{\ell})$. Then he argues that the *total* length of alternating paths is only $O(\Delta m)$, so $\mathrm{E}(|P_{xy}|) = O(\frac{1}{\ell}\Delta m)$. This means that when ℓ is large, i.e., most of the time, the path is usually short, and the total expected length of the alternating paths used by the algorithm is $O(H_\ell \Delta m)$, where $H_\ell = O(\log \ell)$ is the ℓ-th harmonic number.

Another solution for the same problem was used by Chrobak and Nishizeki [10]. They also need a large number ℓ of uncolored edges. They show that then we can pick a set I of size $\Omega(\ell/\Delta(G)^4)$ of these edges so that all the corresponding alternating paths are of the same *type* (alternate the same pair of colors) and moreover coloring one of edges from I using VAL does not interact with coloring the others, intuitively: the VAL algorithm for one of these edges works the same, independent of whether it was invoked for some other edges from I before or not. Hence we can apply VAL to all the edges of I. The total length of the alternating paths is then linear, since they all have the same type and hence they are disjoint. A simple calculation shows that after iterating this $O(\log n\, \Delta(G)^4)$ times all edges get colored, and the total running time of coloring the ℓ edges is $O(m \log n\, \Delta(G)^{O(1)})$.

Now we turn to the setting of this work, where $D = \Delta(G)$ and $\Delta(G) \geq 2\,\mathrm{mad}(G)$. For simplicity assume that $\mathrm{mad}(G)$ is bounded. The problem we encounter is that in our case not all edges are weak. However, Sanders and Zhao [23] show that there is always at least one weak edge. Their proof can be transformed to an algorithm that finds a weak edge xy, removes it, colors the graph recursively, and then extends the coloring. Unfortunately, then the path P_{xy} may be long.

In this work, we show that when average density is bounded, then at least a constant fraction of edges are weak, thus generalizing a similar lower bound of Chrobak and Nishizeki [10] for bounded genus graphs. This allows for a recursive procedure as follows: find weak edges, color the other edges recursively, and then color the weak edges in time $O(m \log n\, \Delta(G)^{O(1)})$ using the approach of Sinnamon or Chrobak-Nishizeki. Since the number of weak edges is $\Omega(m)$, this is only $O(\log n\, \Delta(G)^{O(1)})$ per edge.

The last missing piece in this construction is getting rid of the dependency on $\Delta(G)$. The solution of this particular issue can be found by adapting a technique of Zhou, Nakano and Nishizeki [31]. Namely, one can show that edges of the input graph can be partitioned in such a way that each of the resulting induced graphs has maximum degree at least $2\,\mathrm{mad}(G)$ and at most $4\,\mathrm{mad}(G) + 2$, and moreover the maximum degrees of all the individual graphs sum up to $\Delta(G)$, so we can just color each of them separately using disjoint palettes of colors, and merge the results. In this way, we trade the dependency on $\Delta(G)$ for the dependency on $\mathrm{mad}(G)$. Having $\Delta(G)$ bounded makes many issues easy, for example controlling the running time of the algorithm behind VAL.

1.5 Organization of the paper

This paper is organized as follows. In Section 2 we show a lower bound for the number of weak edges for graphs with $\Delta \geq 2\,\mathrm{ad}(G)$. At the end of the section we will also prove that the constant 2 in the assumption cannot be improved if we want to get any positive lower bound. Next, in Section 3 we show an implementation of the Vizing Adjacency Lemma stating some of its properties that we need. In Section 4 we describe the complete algorithm and we prove Theorem 1. Finally, in Section 5 we summarize the paper and suggest some open problems.

2 Lower bound for the number of weak edges

By a d-vertex (resp. d-neighbor) we mean a vertex (resp. neighbor) of degree d. For an integer i, let $d^i(v)$ be the number of i-neighbors of v. Let D be an integer. In what follows, we call an edge uv (D,u)*-weak* if $d^D(u) \leq D - d(v) + [d(v) = D]$. Moreover, uv is D-weak if it is (D,u)-weak or (D,v)-weak. Note that for $D \geq \Delta(G) + 1$, every edge is D-weak. For simplicity, whenever D is known, we write that an edge is u*-weak* or *weak* when it is (D,u)-weak or weak, respectively. An edge uv is *strong* if it is not weak, i.e., it is neither u-weak nor v-weak. A vertex is called *weak* if it is incident to at least one weak edge, otherwise it is called *strong*.

In this section we focus on the case $D = \Delta(G)$ and we show that for bounded average degree, if $\Delta(G)$ is large enough, then at least a constant fraction of edges are weak.

▶ **Theorem 3.** *Let $G = (V,E)$ be a nonempty graph with no isolated vertices and such that $\Delta(G) \geq 2\,\mathrm{ad}(G)$. Then the number of $\Delta(G)$-weak edges in G is at least*

$$\frac{|E|}{2\,\mathrm{ad}(G)^2}.$$

Proof. For brevity, we denote $\tilde{d} = \mathrm{ad}(G)$. We use the discharging method [14], in a similar spirit as Sanders and Zhao [23], with a crucial difference that here we need a linear lower bound, and not just an existence of a weak edge. We assign an initial *charge* to each vertex v as follows:

$$\mathrm{ch}(v) = d(v) - \tilde{d}.$$

Note that the total charge is zero:

$$\sum_{v \in V} \mathrm{ch}(v) = \sum_{v \in V} d(v) - |V| \sum_{v \in V} d(v)/|V| = 0.$$

Next, we move the charge as follows: every vertex x of degree $d(x) \geq \tilde{d} + \frac{1}{2}$, sends

$$\frac{\mathrm{ch}(x) - \frac{1}{2}}{d(x) - \Delta + d(y) - 1}$$

units of charge to every neighbor y of degree $< \tilde{d} + \frac{1}{2}$ such that edge xy is strong. We claim that the expression above is always well defined and positive. Indeed, since xy is strong, $d(x) \geq [d(y) < \Delta] + d^{\Delta}(x) \geq [d(y) < \Delta] + \Delta - d(y) + [d(y) = \Delta] + 1 = \Delta - d(y) + 2$. This also implies that $d(y) \geq \Delta - d(x) + 2 \geq 2$, which we state below as a claim for later use.

▷ Claim 4. If a vertex receives charge, then it has degree at least 2.

For a vertex v, let $\mathrm{ch}'(v)$ denote the charge of v after moving the charge. Since the charge is only being moved between vertices, $\sum_{v \in V} \mathrm{ch}'(v) = \sum_{v \in V} \mathrm{ch}(v) = 0$. Our goal is to show that

(i) for every strong vertex the final charge is at least $\frac{1}{2}$, and

(ii) for every weak vertex the final charge is at least $1 - \tilde{d}$.

Then, denoting the set of weak vertices by V_w, we have

$$0 = \sum_{v \in V} \mathrm{ch}'(v) \geq |V_w| \cdot (1 - \tilde{d}) + (|V| - |V_w|) \cdot \tfrac{1}{2},$$

which implies $(\tilde{d} - \frac{1}{2})|V_w| \geq |V|/2$. Then, since G has no isolated vertices, $\tilde{d} \geq 1$, and hence $|V_w| \geq |V|/(2(\tilde{d} - \frac{1}{2})) > |V|/(2\tilde{d})$. Since every weak vertex is incident with at least one weak edge, we get at least $|V|/(4\tilde{d}) = |E|/(2\tilde{d}^2)$ weak edges, as required. Hence, indeed it suffices to show (i) and (ii).

▷ Claim 5. For any vertex v of degree $d(v) \geq \tilde{d} + \frac{1}{2}$ we have $\mathrm{ch}'(v) \geq \frac{1}{2}$.

Proof. Since $\mathrm{ch}(v) \geq \frac{1}{2}$, the claim holds trivially when v does not send charge. Hence in what follows assume v sends charge. Then v is incident to at least one strong edge. Let w be a neighbor of v such that vw is strong and $d(w)$ is minimum among such neighbors. Note that $d(w) < \tilde{d} + \frac{1}{2}$, for otherwise v does not send charge to any vertex. Then v sends at most

$$\frac{d(v) - \tilde{d} - \frac{1}{2}}{d(v) - \Delta + d(w) - 1}$$

to each neighbor u of v such that vu is strong and $d(u) < \tilde{d} + \frac{1}{2}$, and does not send to other vertices. Note that v does not send to Δ-vertices, because $\tilde{d} + \frac{1}{2} \leq \frac{1}{2}\Delta + \frac{1}{2} = \Delta - \frac{1}{2}(\Delta - 1) \leq \Delta$. Since vw is strong, $d^{\Delta}(v) \geq \Delta - d(w) + 1$. Hence v has at most $d(v) - \Delta + d(w) - 1$ neighbors of degree smaller than Δ and this is also an upper bound for the number of neighbors it sends charge to. It follows that in total v sends at most $d(v) - \tilde{d} - \frac{1}{2}$ charge, and hence it keeps at least $\frac{1}{2}$ of charge, as required. ◁

▷ Claim 6. For any strong vertex v of degree $d(v) < \tilde{d} + \frac{1}{2}$ we have $\mathrm{ch}'(v) \geq \frac{3}{2}$.

Proof. Let w be a neighbor of v of minimum degree. Since vw is strong, $d(w) \geq d^{\Delta}(w) \geq \Delta - d(v) + 1 > 2\tilde{d} - (\tilde{d} + \frac{1}{2}) + 1 = \tilde{d} + \frac{1}{2}$. It follows that all neighbors of v send charge to v, and

$$\mathrm{ch}'(v) = d(v) - \tilde{d} + \sum_{i=d(w)}^{\Delta} n_i \cdot \frac{i - \tilde{d} - \frac{1}{2}}{i - \Delta + d(v) - 1},$$

where n_i is the number of i-neighbors of v. Since $\Delta - d(v) + 1 > 2\tilde{d} - (\tilde{d} + \frac{1}{2}) + 1 = \tilde{d} + \frac{1}{2}$, we infer that $\frac{i-\tilde{d}-\frac{1}{2}}{i-\Delta+d(v)-1} = 1 + \frac{\Delta-d(v)+1-\left(\tilde{d}+\frac{1}{2}\right)}{i-\Delta+d(v)-1}$ is decreasing with increasing i, so

$$\mathrm{ch}'(v) \geq d(v) - \tilde{d} + d(v) \cdot \frac{\Delta - \tilde{d} - \frac{1}{2}}{d(v) - 1}.$$

Since $d(v) \geq 2$ by Claim 4 and $\Delta \geq 2\tilde{d}$, we get $\mathrm{ch}'(v) \geq \frac{3}{2}$, as required. ◁

To finish the proof, note that (i) follows from claims 5 and 6. For (ii), a weak vertex of degree at least $\tilde{d} + \frac{1}{2}$ has final charge at least $\frac{1}{2}$ by Claim 5, which is more than $1 - \tilde{d}$ since $\tilde{d} \geq 1$ (no isolated vertices). Finally, a weak vertex v of degree $< \tilde{d} + \frac{1}{2}$ does not send charge, so it has final charge $\mathrm{ch}'(v) \geq \mathrm{ch}(v) = d(v) - \tilde{d} \geq 1 - \tilde{d}$, as required. This ends the proof. ◀

We conclude this section by showing that the assumption $\Delta(G) \geq 2\,\mathrm{ad}(G)$ in the formulation of Theorem 3 is the best possible.

▶ **Theorem 7.** *For every $\epsilon > 0$ there are arbitrarily large graphs G with no isolated vertices, with no weak edges and such that $\Delta(G) \geq (2 - \epsilon)\,\mathrm{ad}(G)$.*

Proof. Let d be an integer such that $d \geq \max\{\frac{3}{\epsilon} - 1, 3\}$ and set $\Delta = d(d-1)$. Set also $n_d = (d-1)(\Delta - d + 2)$ and $n_\Delta = d(\Delta - d + 2)$. Construct graph G as follows. Begin with $\Delta - d + 2$ copies of the complete bipartite graph $K_{d-1,d}$ (note that since $d \geq 2$, $\Delta - d + 2 = d(d-2) + 2 \geq 5$). Let V_d be the set of the vertices of the resulting graph of degree d, and let V_Δ be the remaining vertices, i.e., the vertices of degree $d - 1$. Note that $|V_d| = n_d$ and $|V_\Delta| = n_\Delta$. Next, we partition V_Δ arbitrarily into d disjoint sets of size $(\Delta - d + 2)$ and for each such set we add all edges between its verties so that it induces the clique $K_{\Delta-d+2}$. Observe that then every vertex in V_Δ has degree $d - 1 + \Delta - d + 1 = \Delta$. Note that since $d \geq 2$, we have $\Delta \geq d$ and hence G has maximum degree Δ. This completes the construction of G.

Then, we have

$$\mathrm{ad}(G) = \frac{dn_d + \Delta n_\Delta}{n_d + n_\Delta} = \frac{\overbrace{d(d-1)}^{\Delta} + \Delta d}{d - 1 + d} = \frac{d+1}{2d-1}\Delta.$$

It follows that

$$\Delta = \frac{2d-1}{d+1}\,\mathrm{ad}(G) = \left(2 - \frac{3}{d+1}\right)\mathrm{ad}(G) \geq (2 - \epsilon)\,\mathrm{ad}(G).$$

Now it suffices to check that G has no weak edges. Let uv be an edge such that $u \in V_d$ and $v \in V_\Delta$. Then uv is not u-weak because $d^\Delta(u) = d - 1 > 1 = \Delta - d(v) + [d(v) = \Delta]$. Also, uv is not v-weak because $d^\Delta(v) = \Delta - d + 1 > \Delta - d = \Delta - d(u) + [d(u) = \Delta]$. Finally, consider an edge uv between two vertices of V_Δ. Then $d^\Delta(u) = \Delta - d + 1 > 1 = \Delta - d(v) + [d(v) = \Delta]$, so by symmetry uv is not weak. ◀

3 Algorithmic Vizing Adjacency Lemma

The goal of this section is to provide an efficient algorithm which implements the Vizing Adjacency Lemma (Theorem 2). The contents of this section is mostly standard and well-known.

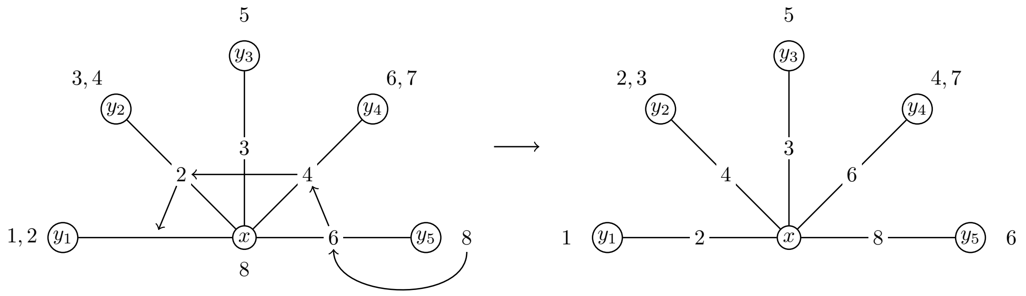

Figure 1 Rotating a fan. The numbers next to vertices denote free colors.

Throughout this section by D we denote the number of available colors. Of course in this paper we are mostly interested in the case when D is the maximum degree Δ, but it is convenient to state results also for the case $D > \Delta$, since our algorithm colors subgraphs of the input graph recursively and at some point the maximum degree drops. A *partial edge-coloring* of graph G is a function $\pi : E(G) \to \{1, \ldots, D\} \cup \{\perp\}$ such that it is an edge-coloring when restricted to edges colored by $\{1, \ldots, D\}$. For a vertex x we define $\pi(x)$ as the set of colors used at edges incident with x, i.e., $\pi(x) = \{\pi(xy) \mid y \in N_G(x)\} \setminus \{\perp\}$. Moreover, $\overline{\pi}(x)$ denotes the set of colors free at x, i.e., $\overline{\pi}(x) = \{1, \ldots, D\} \setminus \pi(x)$. An (a, b)-*alternating path* is a path P in G such that its edges are colored alternately in a and b. We call P *maximal* if it cannot be extended to a longer (a, b)-alternating path. The unordered pair $\{a, b\}$ is called the *type* of P.

Fans, developed by Vizing [26] are useful structures in edge-colorings. The notion appears is several variations. Here we need a new one, which differs from the standard fan by modifying condition (F3) and adding condition (F4) below. The definition is somewhat inspired by multi-fans in the book of Stiebitz et al. [25].

Let $G = (V, E)$ be a graph and let π be a partial edge-coloring of G. Then, a *fan* is a sequence $F = (x, y_1, \ldots, y_k)$ such that

(F1) $y_1, \ldots, y_k$ are different neighbors of x,
(F2) xy_1 is uncolored by π,
(F3) for every $i = 2, \ldots, k$ edge xy_i is colored by π and $\pi(xy_i) \in \bigcup_{j=1}^{i-1} \overline{\pi}(y_j)$,
(F4) for every $i = 2, \ldots, k$, $d(y_i) < D$.

The number k is called the *size* of fan F. Since in this work we we do not use other variants of fan, we do not give it a special name. The notion of a fan is motivated by the fact that there is a simple procedure, called *rotating* the fan, which, given a color $c \in \overline{\pi}(x) \cap \overline{\pi}(y_k)$ recolors some edges of the form xy_i so that edge xy_1 can be colored with a free color, see Algorithm 1 and Figure 1. Of course, the definition of the fan does not guarantee that such a color c exists.

▶ **Lemma 8.** *Given graph G, its partial coloring π, a fan $F = (x, y_1, \ldots, y_k)$ and color $c \in \overline{\pi}(x) \cap \overline{\pi}(y_k)$, Algorithm 1 finds a new partial coloring, which colors the same edges as π and additionally xy_1.*

Proof. It suffices to observe that at the beginning of the loop the following invariant is satisfied: for some $j = 1, \ldots, i$ we have $c \in \overline{\pi}(x) \cap \overline{\pi}(y_j)$. ◀

We call a fan $F = (x, y_1, \ldots, y_k)$ *active* when one of the two conditions below hold:

(A1) there is $i = 1, \ldots, k$ such that $\overline{\pi}(y_i) \cap \overline{\pi}(x) \neq \emptyset$,
(A2) there are $i, j = 1, \ldots, k$ such that $i \neq j$ and $\overline{\pi}(y_i) \cap \overline{\pi}(y_j) \neq \emptyset$.

If additionally $F' = (x, y_1, \ldots, y_{k-1})$ is not active, we call F a *minimal active fan*.

Algorithm 1 Rotating a fan.

```
input : a fan $F = (x, y_1, \dots, y_k)$, color $c \in \bar{\pi}(x) \cap \bar{\pi}(y_k)$.
$i \leftarrow k$
while $i \geq 1$ do
    // Invariant: for some $j = 1, \dots, i$ we have $c \in \bar{\pi}(x) \cap \bar{\pi}(y_j)$
    if $c \in \bar{\pi}(y_i)$ then
        $c' \leftarrow \pi(xy_i)$
        $\pi(xy_i) \leftarrow c$
        $c \leftarrow c'$
    $i \leftarrow i - 1$
```

We will prove a series of simple lemmas. In all of them we work with a graph G and its partial coloring π.

▶ **Lemma 9.** *If a fan $F = (x, y_1, \dots, y_k)$ is not active, then*

$$\left|\bigcup_{i=1}^{k} \bar{\pi}(y_i)\right| \geq D - d(y_1) + k.$$

Proof. Since xy_1 is uncolored, $|\pi(y_1)| \leq d(y_1) - 1$. Then, we have $|\bar{\pi}(y_1)| = D - |\pi(y_1)| \geq D - (d(y_1) - 1)$ and by (F4), $|\bar{\pi}(y_i)| \geq 1$ for $i = 2, \dots, k$. It follows that

$$\left|\bigcup_{i=1}^{k} \bar{\pi}(y_i)\right| \overset{(A2)}{=} \sum_{i=1}^{k} |\bar{\pi}(y_i)| \geq D - (d(y_1) - 1) + (k - 1) \cdot 1 = D - d(y_1) + k.$$ ◀

▶ **Lemma 10.** *Let xy_1 be an x-weak edge. If a fan $F = (x, y_1, \dots, y_k)$ is not active, then it can be extended to another fan $F' = (x, y_1, \dots, y_{k+1})$.*

Proof. Consider the set $Z = \{z \in N_G(x) \mid \pi(xz) \in \bigcup_{i=1}^{k} \bar{\pi}(y_i)\}$. Then,

$$|Z| \overset{(A1)}{=} \left|\bigcup_{i=1}^{k} \bar{\pi}(y_i)\right| \overset{\text{Lemma 9}}{\geq} D - d(y_1) + k.$$

As xy_1 is x-weak, we have $d^D(x) \leq D - d(y_1) + [d(y_1) = D]$. Since $y_1 \notin Z$, because xy_1 is not colored, we know that Z contains at least $D - d(y_1) + k - (D - d(y_1)) = k$ vertices of degree $< D$. It follows that $Z \setminus \{y_i \mid i = 2, \dots, k\}$ has at least one vertex of degree $< D$, call it y_{k+1}. It is easy to check that $(x, y_1, \dots, y_{k+1})$ satisfies conditions (F1)-(F4). ◀

Now we turn to lemmas that describe algorithms. These algorithms use the following data structures that represent the input graph and a partial coloring of it:

- adjacency lists, i.e., array N such that $N[v]$ is a list of neighbors of v;
- for every $v \in V$ we keep an associative array M_v that maps every color $c \in \pi(v)$ to the edge incident with v colored with c. The array is also used to test membership in $\pi(v)$. We use a simple array of size D. Note that the total size of these arrays is $O(nD)$, but we will use it only when D is somewhat small, i.e., $D = O(\mathrm{mad}(G))$.

The lemma below is rather standard (similar statements, but for different variants of the fan notion, can be found for example in Sinnamon [24] or Bhattacharya [5]).

▶ **Lemma 11.** *Given an uncolored x-weak edge xy_1, one can find a minimal active fan $F = (x, y_1, \dots)$ in time $O(D)$. The algorithm also returns the color that makes (A1) or (A2) satisfied.*

Proof. Our algorithm is as follows. We begin with the fan $F_1 = (x, y_1)$. Then we extend it vertex by vertex, until we get an active fan. Let us describe a single step of this process. We will use a set S of colors, initially (i.e., before all steps) empty, implemented using a bitmap (of size $|D|$). We also use a queue Q storing a subset of S. Let $F_i = (x, y_1, \dots, y_i)$ be a fan built so far. We know that $i = 1$ or F_{i-1} is not active (otherwise y_i would not be added). We maintain the invariant that S consists of all the colors free at any y_j, $j < i$. First we check if F_i is active. To this end, for each color $c \in \bar{\pi}(y_i)$ we check if $c \notin \pi(x)$ or $c \in S$ in $O(1)$ time per color c. If one of these checks is positive (A1) or (A2) holds, respectively, so we stop (without checking further colors of $\bar{\pi}(y_i)$) and return F_i. Otherwise we add c both to S and Q and proceed to the next c (or finish the loop). Let us call the loop described above as the *activity checking loop*. When this loop finishes without returning a fan, we know that F_i is not active. At this point we know by Lemma 10 that F_i can be extended by a vertex y_{i+1}. Our algorithm finds y_{i+1} as follows. We remove a color c from Q, and find the edge xz colored with c using M_x. If $d(z) < D$ we set $z = y_{i+1}$, and otherwise we remove another color from Q and so on. This loop will be called the *extension loop*.

Now let us analyze the time complexity. Let $F = (x, y_1, \dots, y_k)$ be the fan returned. Clearly, each iteration of each of the two loops takes constant time, so it suffices to bound the total number of iterations. For the activity checking loop, note that by (A1) each but the last iteration corresponds to a different color that is in $\pi(x)$ (it is not the case for the last iteration only if F was returned at this iteration). Hence the number of these iterations is at most $|\pi(x)| \le d(x) - 1$. Finally, the number of iterations in the extension loop is bounded by the number of colors inserted to Q. However, we insert to Q only in the activity checking loop, one element per iteration. Hence the total number of insertions to Q is bounded by $d(x) - 1$. It follows that the total number of iterations in both loops is $O(d(x))$, and each of them is done in constant time. Since $d(x) \le D$ and initializing bitmap S takes $O(D)$ time, we get the $O(D)$ time bound as required. ◀

Now we are ready to prove the main theorem of this section. Its proof is basically an algorithmization of the proof of Theorem 2.1 from Stiebitz et al. [25], using Lemma 11. Recall that for a vertex v the *closed neighborhood* of v is the set $N[v] = N(v) \cup \{v\}$.

▶ **Theorem 12** (Algorithmic VAL). *Let G be a simple graph and let $e = xy$ be an x-weak edge. Then there is an algorithm which, given any partial D-edge-coloring π of G which colors a subset E_c of edges of G, $e \notin E_c$ finds a partial D-edge-coloring π' that colors $E_c \cup \{e\}$. The algorithm runs in time $O(D + |P|)$, where P is a maximal alternating path with an endpoint in x, possibly $P = \emptyset$. Moreover,*

(V1) *if needed, one can specify the color $c' \in \bar{\pi}(x)$ that alternates in P,*

(V2) *π' may differ from π only on edges incident to x and on edges of P,*

(V3) *if $\pi(v) \neq \pi'(v)$ for a vertex v, then $v \in N[x]$ or v is the other endpoint of P,*

(V4) *the type of P can be determined in time $O(D)$,*

(V5) *the type of P depends only on $\{(\pi(xz), \pi(z)) \mid z \in N(x)\}$.*

Proof. Let $y_1 = y$. We apply Lemma 11. Let $F = (x, y_1, \dots, y_k)$ be the returned minimal active fan. Note that $k \le d(x) \le D$.

Since F is active, it satisfies (A1) or (A2). In the former case, we have a color $c \in \bar{\pi}(y_k) \cap \bar{\pi}(x)$. Then we *rotate* F using Algorithm 1, which clearly runs in time $O(k)$ since the condition in line 3 is checked in constant time using M_{y_1}. By property (F3), this results in the desired partial edge-coloring. Note that $(V2)$ is satisfied, and $(V1)$, $(V3)$-$(V5)$ are trivial since $P = \emptyset$.

Now assume F does not satisfy (A1) but it does satisfy (A2), i.e., by Lemma 11 we have a color c that is free both in y_k and y_i for some $i = 1, \ldots, k-1$. Such y_i can be found simply by testing if $c \in \pi(y_j)$ for every $j = 1, \ldots, k-1$: this takes time $O(k) = O(D)$.

Let c' be any color free at x (it exists since xy is uncolored, and if c' is not given, it can be found in time $O(D)$). Find a maximal (c, c')-alternating path P starting at x. This can be easily done in time $O(|P|)$ using the associative arrays M_*. Then there is $j \in \{i, k\}$ such that the other endpoint of the path P is not y_j. We swap the colors c and c' on P in $O(|P|)$ time, obtaining another partial coloring with the same set of colored edges, but with $c \in \bar{\pi}(x) \cap \bar{\pi}(y_j)$. Then we rotate F_j using Algorithm 1. We see that this procedure satisfies $(V2)$, $(V3)$ and $(V5)$. For $(V4)$, we just perform the procedure without finding P and without recoloring. ◀

4 The main algorithm

In this section we present our main result.

We begin with two lemmas which will be used to control the length of alternating paths from Theorem 12. The first one is due to Sinnamon [24] and uses randomization.

▶ **Lemma 13** (Lemma 10 in Sinnamon [24]). *Let G be a simple graph and with a partial edge-coloring π which colors all but ℓ edges of G. Let e be a random uncolored edge, let x be an arbitrary endpoint of e and let c_0 be a random color of $\pi(x)$. Let c_1 be an arbitrary color different than c_0. Consider the unique maximal (c_0, c_1)-alternating path P_r starting at x. Then, $\mathrm{E}(|P_r|) = O(|E(G)|\Delta(G)/\ell)$.*

Note that if we iterate Theorem 12 for $\ell = \Omega(|E|)$ times, then by Lemma 13 the expected length per edge is low, namely $O(\Delta(G)H_\ell) = O(\Delta(G)\log n)$. Below we state a lemma of Chrobak and Nishizeki, which achieves a similar goal deterministically. Since their proof does not analyze the dependency on $\Delta(G)$, we reproduce it here, but involving $\Delta(G)$ in the analysis.

▶ **Lemma 14** (Chrobak and Nishizeki [10]). *Let G be a simple graph and let $E_w \subseteq E(G)$ be a subset of weak edges. Denote $E_0 = E(G) \setminus E_w$. There is a deterministic algorithm which, given G, E_w and a partial Δ-edge-coloring π which colors E_0, finds a new partial Δ-edge-coloring π' of G which colors E_1 such that $E_0 \subseteq E_1$ and $|E_1 \setminus E_0| \geq |E_w|/(9\Delta(G)^5)$. The algorithm runs in time $O(|E(G)| + |E_w|\Delta(G))$.*

Proof. First, for every edge $xy \in E_w$ we determine the type $\{a, b\}$ of a maximal alternating path P_{xy} that would be used if xy was colored using Theorem 12; let's call it the type of xy. For the special case $P_{xy} = \emptyset$, the type of xy is also defined as $\emptyset$. By Theorem 12, property $(V4)$, this takes $O(|E_w|\Delta(G))$ time.

Since there are $\binom{\Delta(G)}{2} + 1 \leq \Delta^2$ types, there is a type, call it t, with at least $|E_w|/\Delta(G)^2$ edges. Let E_w^t be a set of $\lceil |E_w|/\Delta(G)^2 \rceil$ edges of type t.

Consider two two different edges $e = xy, e' = x'y'$, both from $E_w^{a,b}$, that are x-weak and x'-weak, respectively. We say that e *interacts* with e' if coloring e using Theorem 12 changes the type of e' or recolors an edge of $P_{e'}$, or vice versa, i.e., coloring e' changes the type of e or recolors an edge of P_e.

▷ Claim 15. Edges e and e' do not interact when

(i) $N[x] \cap N[x'] = \emptyset$, and

(ii) P_e does not end in $N[x']$ and $P_{e'}$ does not end in $N[x]$.

Proof. Consider $z' \in N(x')$. Assume $\pi(x'z')$ changes after coloring e using Theorem 12. By $(V2)$, $x'z'$ is incident to x or $x'z' \in P_e$. The former is excluded by (i), so assume the latter. It means $t = \{a, b\}$ for some colors a, b. By (ii), path P_e does not end in x', so a and b are not free at x', a contradiction with $P_{e'}$ being a maximal (a, b)-path with an endpoint in x'. Hence $\pi(x'z')$ does not change. Similarly, $\pi(z')$ does not change, for otherwise by $(V3)$ we have $z' \in N[x]$, contradicting (i), or P_e ends in z', contradicting (ii). Thus, by $(V5)$, the type of e' does not change.

Assume an edge of $P_{e'}$ was recolored after coloring e using Theorem 12. Then $P_{e'} \neq \emptyset$, so $t = \{a, b\}$ for some colors a, b. By (ii), we have $P_{e'} \neq P_e$, so by $(V2)$ path $P_{e'}$ has an edge incident to x. Also by (ii), path $P_{e'}$ does not end at x, contradicting P_e being a maximal (a, b)-path with an endpoint in x. ◁

We proceed as follows. We form a graph $Q = (E_w^t, E(Q))$, where $E(Q)$ is the set of pairs $e, e' \in E_w^t$ that violate (i). By definition, $\Delta(Q) \leq \Delta(G)(\Delta(G) - 1)^2$ so a simple greedy algorithm finds a maximal independent set S in Q of size

$$|S| \geq \frac{|E_w^t|}{\Delta(Q) + 1} \geq \frac{|E_w^t|}{\Delta(G)^3} \geq \frac{|E_w|}{\Delta(G)^5}.$$

The greedy algorithm runs in time linear in the size of Q, which is $|E_w^t|\Delta(Q) = O(E_w\Delta(G))$.

Finally, when $t \neq \emptyset$, i.e., $t = \{a, b\}$, we form yet another graph $Q' = (S, E(Q'))$, where $E(Q')$ is the set of pairs $e, e' \in S$ that violate (ii). To form Q' we need to find all the (a, b)-alternating paths that correspond to edges in S, but it takes only time $O(|E(G))$ thanks to the fact that these paths are pairwise disjoint. We claim that $|E(Q')| \leq |V(Q')|$. Indeed, consider $xy \in S$ and the (a, b)-alternating path P_{xy} starting at x. Let w be the other endpoint of P_{xy}. Since edges of S satisfy (i), x is not in a closed neighborhood of an endpoint of an edge from S different from e. If w is in a closed neighborhood of an endpoint of an edge e_w from S, then again by (i) we know that e_w is unique. Hence, xy generates at most one edge in Q', and $|E(Q')| \leq |V(Q')|$. It follows that Q' has a set Z of at least $|V(Q')|/3$ vertices of degree at most 2, because otherwise $|E(Q')| = \frac{1}{2}\sum_{e \in V(Q')} d_{Q'}(e) > \frac{1}{2} \cdot 3 \cdot \frac{2}{3}|V(Q')| = |V(Q')|$, a contradiction. By repeatedly selecting a vertex of Z and discarding its at most two neighbors we get an independent set I in Q of size

$$|I| \geq |V(Q')|/9 = \frac{|E_w|}{9\Delta(G)^5}.$$

This takes linear time in the size of Q', which is $O(|S|) = O(E_w)$. If $t = \emptyset$ we just set $I = S$.

We conclude by applying Theorem 12 to color all edges of I. Since the corresponding alternating paths are disjoint, and they do not change because edges of I do not interact with each other, their total length is bounded by $|E(G)|$. By Theorem 12, it follows that the total time of this coloring is $O(|I|\Delta(G) + |E(G)|) = O(|E_w|\Delta(G) + |E(G)|)$. ◀

Now we need a graph partitioning technique due to Zhou, Nakano, Nishizeki [31]. Since we state the properties of this technique somewhat differently than them, below we provide a self-contained proof.

Let $G = (V, E)$ be a graph. For an integer c, a (Δ, c)-*partition* of G is any partition $E = E_1 \cup E_2 \cup \cdots E_k$ such that for graphs $G_i = G[E_i]$, $i = 1, \ldots, k$ we have

(P1) $\Delta(G) = \sum_{i=1}^{k} \Delta(G_i)$, and

(P2) $\Delta(G_i) = c$ for $i = 1, \ldots, k - 1$, while $c \leq \Delta(G_k) < 2c$.

Recall that the *degeneracy* of G is the smallest integer k such that every subgraph of G has a vertex of degree at most k.

▶ **Lemma 16** (implicit in Zhou, Nakano, Nishizeki [31]). *For every graph G of degeneracy k, for every integer $c \geq k$, a (Δ, c)-partition of G can be found in linear time.*

Proof. Set $s = \lfloor \frac{\Delta(G)}{c} \rfloor$. It follows that $sc \leq \Delta(G) < (s+1)c$.

Let $n = |V(G)|$ and let $v_1, \ldots, v_n$ be the degeneracy ordering of $V(G)$, i.e., for every $i = 1, \ldots, n$, there are at most k edges between v_i and vertices $v_1, \ldots, v_{i-1}$. It is well known that degeneracy ordering can be found in linear time (by repeatedly removing vertices of the smallest degree and reversing the order).

Sets $E_1, \ldots, E_s$ are constructed using the following procedure. Begin with $E_i = \emptyset$ for $i = 1, \ldots, s$. Next process vertices v_i, $i = 1, \ldots, n$ one by one, as follows. For every vertex v_i we assign the edges incident to v_i and vertices $\{v_j \mid j > i\}$ to sets $E_1, \ldots, E_s$ in a way we describe later. This means that when we process v_i, its incident edges to vertices $\{v_j \mid j < i\}$ are already assigned. However, since we use the degeneracy ordering, the total number of these latter edges is at most k, so in particular for every $\ell = 1, \ldots, s$ the set E_ℓ contains at most k edges incident to vertices $\{v_j \mid j < i\}$. Then, for every $\ell = 1, \ldots, s-1$ we proceed as follows. Let r be the number of the yet not assigned edges from v_i to $\{v_j \mid j > i\}$. Assign $\min\{c - d_{G[E_\ell]}(v_i), r\}$ edges to $G[E_\ell]$. If all edges incident to v_i are assigned, proceed to v_{i+1}. If after adding edges to all graphs E_ℓ, $\ell = 1, \ldots, s-1$ there are still unassigned edges left, assign all of them to E_s.

Note that after processing any v_i, we have $d_{G[E_\ell]}(v_i) \leq c$ for every $\ell = 1, \ldots, s-1$. Also, $d_{G[E_s]}(v_i) < 2c$, because otherwise $d_G(v_i) = \sum_{\ell=1}^{s} d_{G[E_\ell]}(v_i) \geq (s+1)c > \Delta(G)$, a contradiction. These degrees do not change when we process vertices $\{v_j \mid j > i\}$. Hence, $\Delta(G[E_\ell]) \leq c$ for every $\ell = 1, \ldots, s-1$ and $\Delta(G[E_s]) < 2c$.

Now consider v_i such that $d_G(v_i) = \Delta(G)$. We claim that $d_{G[E_\ell]}(v_i) = c$ for every $\ell = 1, \ldots, s-1$. Indeed, otherwise at most $c(s-1) - 1$ edges incident to v_i were assigned to all E_ℓ, $\ell = 1, \ldots, s-1$ and at most k edges incident to v_i were assigned to E_s, and these are all edges incident to v_i so $\Delta(G) = d_G(v_i) \leq c(s-1) - 1 + k \leq cs - 1$, a contradiction with the choice of s. Hence indeed $d_{G[E_\ell]}(v_i) = c$ for every $\ell = 1, \ldots, s-1$. This, together with $sc \leq d_G(v_i) < (s+1)c$ implies that $c \leq d_{G[E_s]}(v_i) < 2c$. By the previous paragraph we get that $\Delta(G[E_\ell]) = c$ for every $\ell = 1, \ldots, s-1$ and $c \leq \Delta(G[E_s]) < 2c$, as required. ◀

Now we proceed with a lemma that is a basic building block of our main result.

▶ **Lemma 17.** *Every graph G with n vertices and m edges such that $\Delta(G) \geq 2\,\mathrm{mad}(G)$ can be $\Delta(G)$-edge-colored*

1. *by a randomized algorithm running in time $O(m\Delta(G)\,\mathrm{mad}(G)^2 \log n)$ in the expectation and with high probability,*
2. *by a deterministic algorithm in time $O(m\Delta(G)^5(\mathrm{mad}(G)^2 + \Delta(G)) \log n)$.*

Proof. The algorithm is recursive: it invokes itself recursively for a subgraph of the given graph. Consider a single recursive call and let $H \subseteq G$ be the graph passed to the current call. We have $D = \Delta(G)$ colors. We begin by removing all isolated vertices from H. If we end up with the empty graph, we return the empty coloring. Otherwise, we find the set E_w of weak edges of H. This can be easily done in linear time, by first computing, for each vertex v its degree $d(v)$ and Δ-degree $d^\Delta(v)$; then checking if an edge is weak directly from definition takes $O(1)$ time. By Theorem 3, $|E_w| \geq |E(H)|/(2\,\mathrm{mad}(G)^2)$. In what follows we assume $|E_w| = \lceil |E(H)|/(2\,\mathrm{mad}(G)^2) \rceil$ by skipping some edges if needed. Next, we form a graph $H' = H - E_w$ in time linear in the size of H. Then, if $E(H') \neq \emptyset$, we color H' recursively. Thus we obtain a partial coloring π of H.

Now we proceed differently depending on whether we aim at randomized or deterministic algorithm. In the randomized case, we color edges of E_w in random order, using Theorem 12 (the color c' mentioned in the theorem statement is chosen randomly: it takes time $O(D)$

to generate the list of free colors and next we sample from it in constant time). In the deterministic case, we apply Lemma 14 repeatedly until everything is colored (in every iteration E_w in Lemma 14 is the set of yet uncolored edges). This completes the description of the algorithm.

Now we proceed to the running time analysis. Let us analyze the time complexity of the single level of the recursion.

Begin with the randomized case. By Theorem 12 and Lemma 13, when $\ell = 1, \ldots, |E_w|$ edges are still uncolored, coloring a random uncolored edge xy of E_w takes expected time $O\left(\Delta(G) + |E(H)|\Delta(G)/\ell\right) = O(|E(H)|\Delta(G)/\ell)$. Hence, coloring all the ℓ edges is done in expected time $O(|E(H)|H_{|E_w|}\Delta(G))$, where $H_{|E_w|}$ is the $|E_w|$-th harmonic number. Since $H_{|E_w|} < \ln|E_w| + 1 = O(\log n)$ we get the bound $O(|E(H)|\Delta(G)\log n)$ and this is also a bound on the expected running time for a single level of the recursion, since E_w and H' can be found in time $O(|E(H)|)$.

For the deterministic case, we see that after each invocation of Lemma 14, the number of uncolored edges decreases at least by the factor $r_0 = 1 - \frac{1}{9\Delta(G)^5}$. The number of invocations is then at most $\log_{1/r_0}|E_w| = O(\Delta(G)^5 \log|E_w|)$ by standard properties of logarithm. By Lemma 14, it follows that in the deterministic case a single level of the recursion takes time $O((|E(H)| + |E_w|\Delta(G))\Delta(G)^5 \log n) = O((1 + \Delta(G)/\operatorname{mad}(G)^2)|E(H)|\Delta(G)^5 \log n)$.

Now we bound the time of all the levels of the recursion. Note that if $\Delta(H) < \Delta(G) = D$, then every edge of H is weak and there are no more recursive calls. Otherwise, we have $D = \Delta(H)$ and $\Delta(H) = \Delta(G) \geq 2\operatorname{mad}(G) \geq 2\operatorname{ad}(H)$, so by Theorem 3 we have $|E(H')| \leq r|E(H)|$ for

$$r = 1 - \frac{1}{2\operatorname{ad}(H)^2} \leq 1 - \frac{1}{2\operatorname{mad}(G)^2}.$$

Hence, in the randomized case, all the L levels of the recursion take expected time

$$O\left(\sum_{i=0}^{L-1} r^i |E(G)|\Delta(G)\log n\right).$$

Since $\sum_{i=0}^{L-1} r^i < \sum_{i=0}^{\infty} r^i = 2\operatorname{mad}(G)^2$, the total time complexity is $O(m\Delta(G)\log n \operatorname{mad}(G)^2)$ in expectation, as required. The same kind of calculation for the deterministic case gives the running time bound of $O(m(\operatorname{mad}(G)^2 + \Delta(G))\Delta(G)^5 \log n)$.

Obtaining the high probability guarantee for the running time in the randomized case is fairly standard, we omit it here (see the full version [22]). ◀

Now we are ready to prove our main result.

▶ **Theorem 1.** *Every graph G with n vertices and m edges such that $\Delta(G) \geq 2\operatorname{mad}(G)$ can be $\Delta(G)$-edge-colored*

1. *by a randomized algorithm running in time $O(m\operatorname{mad}(G)^3 \log n)$ in the expectation and with high probability,*
2. *by a deterministic algorithm in time $O(m\operatorname{mad}(G)^7 \log n)$.*

Proof. Let $c = \lceil 2\operatorname{mad}(G)\rceil$. We begin with finding a (Δ, c)-partition $E_1, \ldots, E_s$ of G using Lemma 16. Note that by definition, degeneracy is at most $\operatorname{mad}(G)$ so the assumptions of the lemma are satisfied.

Thanks to property (P2) of (Δ, c)-partition, for every $i = 1, \ldots, s$,

$$2\operatorname{mad}(G[E_i]) \leq 2\operatorname{mad}(G) \leq \Delta(G[E_i]) < 2\lceil 2\operatorname{mad}(G)\rceil < 4\operatorname{mad}(G) + 2.$$

Hence we can use Lemma 17 to $\Delta(G[E_i])$-edge-color each graph $G[E_i]$, $i = 1, \ldots, s$ separately, in expected time $O(|E_i| \log n \operatorname{mad}(G)^3)$ or deterministic time $O(|E_i| \operatorname{mad}(G)^7 \log n)$. In the randomized case, by linearity of expectation, we get $O(m \log n \operatorname{mad}(G)^3)$ expected time in total (and the high probability bound also follows since the probability of a bad event increases only at most $s \le \Delta \le n$ times). In the deterministic case, after summing we get $O(m \operatorname{mad}(G)^7 \log n)$. Since by the property (P1) we can use disjoint sets of colors, this results in the desired coloring of G. ◀

5 Conclusion and further research

We showed that every graph G of bounded maximum average degree can be edge-colored using the optimal number of Δ colors in quasilinear time, whenever $\Delta \ge 2 \operatorname{mad}(G)$. We presented two algorithms: a randomized one in time $O(m \operatorname{mad}(G)^3 \log n)$ and a deterministic one in time $O(m \operatorname{mad}(G)^7 \log n)$. As a corollary we get also that every graph G of bounded maximum average degree can be $(\Delta + 1)$-edge-colored in quasilinear *deterministic* time, thus derandomizing the result of Bhattacharya et al [5]. We conclude with some open problems as follows.

1. Is it possible to relax our assumption to $\Delta \ge c \operatorname{mad}(G)$ for $c < 2$? As shown in Theorem 7 this would involve something more than the VAL in the coloring procedure. Since the graphs constructed in Theorem 7 have arbitrarily large minimum degree, it is also not sufficient to find reducible configurations with vertices of bounded degree, as it was the case in the work of Cole and Kowalik [12].
2. Is it possible to design a data structure for *dynamic* Δ-edge-coloring graphs of bounded maximum average degree, assuming $\Delta \ge c \operatorname{mad}(G)$ for a constant c, with updates in sublinear time? Or in polylogarithmic time? Note that the dynamic edge-coloring is an active area of research recently, see e.g. [15, 8, 6, 9].
3. Is it possible to implement a version of Theorem 1 in the LOCAL model using only $\operatorname{poly}(\Delta, \log n)$ rounds? For some recent related results see e.g. [9, 2].

References

1 Sepehr Assadi. Faster Vizing and near-Vizing edge coloring algorithms. *CoRR*, abs/2405.13371, 2024. doi:10.48550/arXiv.2405.13371.

2 Anton Bernshteyn. A fast distributed algorithm for $(\Delta + 1)$-edge-coloring. *J. Comb. Theory, Ser. B*, 152:319–352, 2022. doi:10.1016/J.JCTB.2021.10.004.

3 Anton Bernshteyn and Abhishek Dhawan. Fast algorithms for Vizing's theorem on bounded degree graphs. *CoRR*, abs/2303.05408, 2023. doi:10.48550/arXiv.2303.05408.

4 Sayan Bhattacharya, Din Carmon, Martín Costa, Shay Solomon, and Tianyi Zhang. Faster $(\Delta + 1)$-edge coloring: Breaking the $m\sqrt{n}$ time barrier. *CoRR*, abs/2405.15449, 2024. doi:10.48550/arXiv.2405.15449.

5 Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Density-sensitive algorithms for $(\Delta{+}1)$-edge coloring. *CoRR*, abs/2307.02415, 2023. To appear in ESA'24. arXiv:2307.02415, doi:10.48550/arXiv.2307.02415.

6 Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Arboricity-dependent algorithms for edge coloring. In Hans L. Bodlaender, editor, *19th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2024, June 12-14, 2024, Helsinki, Finland*, volume 294 of *LIPIcs*, pages 12:1–12:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.SWAT.2024.12.

7 Yan Cao, Guantao Chen, Suyun Jiang, Huiqing Liu, and Fuliang Lu. Average degrees of edge-chromatic critical graphs. *Discrete Mathematics*, 342(6):1613–1623, 2019. doi:10.1016/j.disc.2019.02.014.

8 Aleksander B. G. Christiansen, Eva Rotenberg, and Juliette Vlieghe. Sparsity-parameterised dynamic edge colouring. In Hans L. Bodlaender, editor, *19th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2024, June 12-14, 2024, Helsinki, Finland*, volume 294 of *LIPIcs*, pages 20:1–20:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.SWAT.2024.20.

9 Aleksander Bjørn Grodt Christiansen. The power of multi-step Vizing chains. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1013–1026. ACM, 2023. doi:10.1145/3564246.3585105.

10 Marek Chrobak and Takao Nishizeki. Improved edge-coloring algorithms for planar graphs. *J. Algorithms*, 11(1):102–116, 1990. doi:10.1016/0196-6774(90)90032-A.

11 Marek Chrobak and Moti Yung. Fast algorithms for edge-coloring planar graphs. *J. Algorithms*, 10(1):35–51, 1989. doi:10.1016/0196-6774(89)90022-9.

12 Richard Cole and Lukasz Kowalik. New linear-time algorithms for edge-coloring planar graphs. *Algorithmica*, 50(3):351–368, 2008. doi:10.1007/S00453-007-9044-3.

13 Richard Cole, Kirstin Ost, and Stefan Schirra. Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Comb.*, 21(1):5–12, 2001. doi:10.1007/S004930170002.

14 Daniel W. Cranston and Douglas B. West. An introduction to the discharging method via graph coloring. *Discrete Mathematics*, 340(4):766–793, 2017. doi:10.1016/j.disc.2016.11.022.

15 Ran Duan, Haoqing He, and Tianyi Zhang. Dynamic edge coloring with improved approximation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1937–1945. SIAM, 2019. doi:10.1137/1.9781611975482.117.

16 Michael Elkin and Ariel Khuzman. Deterministic simple $(1+\varepsilon)\Delta$-edge-coloring in near-linear time, 2024. arXiv:2401.10538.

17 Stanley Fiorini. Some remarks on a paper by Vizing on critical graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77(3):475–483, 1975. doi:10.1017/S030500410005129X.

18 Harold N. Gabow, Takao Nishizeki, Oded Kariv, Daniel Leven, and Osamu Terada. Algorithms for edge-coloring graphs. Technical report, Tel Aviv University, 1985. URL: https://www.ecei.tohoku.ac.jp/alg/nishizeki/sub/e/Edge-Coloring.pdf.

19 Dawit Haile. Bounds on the size of critical edge-chromatic graphs. *Ars Comb.*, 53:85–96, 1999.

20 Ian Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981. doi:10.1137/0210055.

21 Lukasz Kowalik. Approximation scheme for lowest outdegree orientation and graph density measures. In Tetsuo Asano, editor, *Algorithms and Computation, 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006, Proceedings*, volume 4288 of *Lecture Notes in Computer Science*, pages 557–566. Springer, 2006. doi:10.1007/11940128_56.

22 Lukasz Kowalik. Edge-coloring sparse graphs with Δ colors in quasilinear time. *CoRR*, abs/2401.13839, 2024. doi:10.48550/arXiv.2401.13839.

23 Daniel P. Sanders and Yue Zhao. On the size of edge chromatic critical graphs. *J. Comb. Theory, Ser. B*, 86(2):408–412, 2002. doi:10.1006/JCTB.2002.2135.

24 Corwin Sinnamon. Fast and simple edge-coloring algorithms. *CoRR*, abs/1907.03201, 2019. arXiv:1907.03201.

25 Michael Stiebitz, Diego Scheide, Bjarne Toft, and Lene M. Favrholdt. *Graph edge coloring: Vizing's theorem and Goldberg's conjecture.* Wiley Hoboken, NJ, Hoboken, NJ, 2012.

26 Vadim G. Vizing. On the estimate of the chromatic class of a p-graph. *Diskret. Analiz*, 3:25–30, 1964.

27 Vadim G. Vizing. Critical graphs with a given chromatic number. *Diskret. Analiz*, 5:9–17, 1965.

28 Vadim G. Vizing. Some unsolved problems in graph theory (in Russian). *Uspeki Mat. Nauk.*, 23:117–134, 1968. English translation in [*Russian Mathematical Surveys* **23** (1968), 125-141].

29 Douglas R. Woodall. The average degree of an edge-chromatic critical graph II. *Journal of Graph Theory*, 56(3):194–218, 2007. `doi:10.1002/jgt.20259`.

30 Douglas R. Woodall. Erratum: The average degree of an edge-chromatic critical graph II. *Journal of Graph Theory*, 92(4):488–490, 2019. `doi:10.1002/jgt.22501`.

31 Xiao Zhou, Shin ichi Nakano, and Takao Nishizeki. Edge-coloring partial k-trees. *Journal of Algorithms*, 21(3):598–617, 1996. `doi:10.1006/jagm.1996.0061`.