

Determining the circular flow number of a cubic graph

Robert Lukot'ka *

Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava, Slovakia

lukotka@dcs.fmph.uniba.sk

Submitted: May 28, 2020; Accepted: Feb 19, 2021; Published: Mar 12, 2021

© The author. Released under the CC BY license (International 4.0).

Abstract

A *circular nowhere-zero r -flow* on a bridgeless graph G is an orientation of the edges and an assignment of real values from $[1, r - 1]$ to the edges in such a way that the sum of incoming values equals the sum of outgoing values for every vertex. The *circular flow number*, $\phi_c(G)$, of G is the infimum over all values r such that G admits a nowhere-zero r -flow. A flow has its *underlying orientation*. If we subtract the number of incoming and the number of outgoing edges for each vertex, we get a mapping $V(G) \rightarrow \mathbb{Z}$, which is its *underlying balanced valuation*. In this paper we describe efficient and practical polynomial algorithms to turn balanced valuations and orientations into circular nowhere zero r -flows they underlie with minimal r . Using this algorithm one can determine the circular flow number of a graph by enumerating balanced valuations. For cubic graphs we present an algorithm that determines $\phi_c(G)$ in case that $\phi_c(G) \leq 5$ in time $O(2^{0.6 \cdot |V(G)|})$. If $\phi_c(G) > 5$, then the algorithm determines that $\phi_c(G) > 5$ and thus the graph is a counterexample to Tutte's 5-flow conjecture. The key part is a procedure that generates all (not necessarily proper) 2-vertex-colourings without a monochromatic path on three vertices in $O(2^{0.6 \cdot |V(G)|})$ time. We also prove that there is at most $2^{0.6 \cdot |V(G)|}$ of them.

Mathematics Subject Classifications: 05C21, 05C30, 05C85

1 Introduction

Many problems related to nowhere-zero flows have a prominent place within graph theory, e.g. Four Colour Theorem, Tutte's flow conjectures, etc. In this paper we study a natural

*The author received partial support from APVV-19-0308 and VEGA 1/0813/18.

refinement of the flow number—the circular flow number. A *circular nowhere-zero (CNZ) r -flow* on a bridgeless graph G is an orientation of the edges of G and an assignment of real values from $[1, r - 1]$ to the edges in such a way that the sum of the incoming values equals the sum of the outgoing values for every vertex. The *circular flow number (CFN)*, denoted by $\phi_c(G)$, of G is the infimum over all values r such that G admits a CNZ r -flow. The CFN was introduced by Goddyn, Tarsi and Zhang as the dual concept to the circular chromatic number and as a natural refinement of the ordinary flow number [3]. It is known that the infimum from the definition is attained for finite graphs and that the flow number is the ceiling of the CFN [3]. The concept has various associations to different areas of graph theory. That is why, despite the fact that the CFN was introduced only in the 90's, several important results can be found in older literature where equivalent results were obtained using related concepts. E.g. Jaeger published his (now disproved [8]) circular flow conjecture [11] in the 80's. Jaeger's result that connects CNZ r -flows, orientations, and balanced valuations is even older [10].

Let G be a graph on n vertices and m edges. A flow has its orientation. If we take an orientation and we subtract the number of edges incoming and the number of edge outgoing to each vertex we get a mapping $V(G) \rightarrow \mathbb{Z}$, a balanced valuation. This relation of flows, orientations, and balanced valuations plays an important role in our paper. Although the arguments of Jaeger [10] and the arguments in a referenced paper of Hakimi [7] are not constructive, with some care they can be turned into polynomial algorithms. As the implementation in [4] does not use the result of [7]. Thus we feel necessary to provide these algorithms. We significantly modify the arguments of Jaeger and Hakimi to obtain efficient algorithms to change a balanced valuation into an orientation ($O(m^{1.5})$ time) and an orientation into a CNZ r -flow with r minimal ($O(m^{1.5} \cdot \log^2(m))$ time). The algorithms use the algorithm of Goldberg and Rao [5] to solve maximum flow instances (if graphs are dense, another version of Goldberg's and Rao's algorithms provide better running times, but we are mostly interested in sparse graphs). This implies that given a graph G with d_i vertices of degree i can determine its CFN in $O(\prod (i - 1)^{d_i})$ time.

For the main result of the paper we focus on cubic graphs. Cubic graphs play an important role in the study of flows as several important problems (Four Colour Theorem, Tutte's 5-flow Conjecture, etc.) can be naturally reduced to cubic graphs. The problem of determining the CFN is NP-hard even when in the class of cubic graph (CFN is > 4 is equivalent to 3-edge-colourability, which is NP complete [9]). A natural approach to determine circular flow number is to enumerate all balanced valuations of a graph and compute its circular flow number [4].

Assuming that the Tutte's 5-flow conjecture holds the CFN of a bridgeless graph is at most 5. If a balanced valuation of a cubic graph contains a path on three positive (or negative) vertices, its CFN is at least 5. This allows us to cut the balanced valuations we need to consider significantly [4]. Goedgebeur, Mattiolo, and Mazzuoccolo used this approach to devise and implement a practical algorithm to determine the CFN of a cubic graph. However, in their paper they do not address the problem of generating the “interesting” balanced valuations and they give no bounds on the time complexity of their algorithm. Besides the algorithm that generates an optimal CNZ flow from a balanced

valuation we give an algorithm to generate all (not necessarily proper) 2-vertex-colourings of cubic graph without three consecutive vertices of the same colour. The algorithm runs in $O(2^{0.6n})$. We also prove that the number of such colourings is at most $2^{0.6n}$ (we even sketch how to improve the constant 0.6 slightly). Together this yields an algorithm that either determines the CFN of a cubic graph, or it says that the CFN of the graph is > 5 and has running time $O(2^{0.6n})$ (To deal with the case that $\text{CFN} = 5$ we use a standard approach based on path decompositions). The algorithms are fairly practical to implement.

The paper is structured as follows. In Section 2 we discuss the relation between balanced valuations, orientations, and CNZ r -flows. We describe algorithms that take a balanced valuation/orientation/CNZ flow and return a corresponding object of a different type. In Section 3 we discuss how this correspondence can be used to determine the CFN. Section 4 is devoted to the key part of $O(2^{0.6n})$ algorithm. We show how to generate all relevant balanced valuations efficiently. This process is what determines the complexity of the algorithm. Finally, in Section 5 we discuss several marginal improvements of the algorithms (this is why we can drop the polynomial factors from the resulting time complexities) and the applicability of the algorithm to the dual concept of circular colourings.

2 Balanced valuations, orientations and flows

Let G be a graph. Consider a mapping $b : V(G) \rightarrow \mathbb{R}$, let $S \subseteq V(G)$, and let O be an orientation of G . We define $b(S) = \sum_{v \in S} b(v)$. By $\partial(S)$ we denote the number of edges of G with exactly one end in S . By $\partial_O^+(S)$ ($\partial_O^-(S)$) we denote the number of edges with exactly one end in S oriented towards (away from) a vertex of S . If $S = \{v\}$, we write just $\partial(v)$, $\partial_O^+(v)$ and $\partial_O^-(v)$. A *balanced valuation*¹ is a mapping $b : V(G) \rightarrow \mathbb{Z}$ such that

- $b(S) \leq \partial(S)$ for each $S \subseteq V(G)$ and
- $b(v) \equiv \partial(v) \pmod{2}$ for each $v \in V(G)$.

There is an easy correspondence between CNZ r -flows, orientations, and balanced valuations. Given a CNZ r -flow ϕ we have its *underlying* orientation O . From O we can produce a balanced valuation b by setting $b(v) = \partial_O^+(v) - \partial_O^-(v)$. We say that b is *underlying* both ϕ and O . Note that $b(S) = \partial_O^+(S) - \partial_O^-(S)$ for each $S \subseteq V(G)$. Thus b is, indeed, a balanced valuation.

The fact that ϕ is a CNZ r -flow restricts both O and b . As the maximal flow value is $r - 1$ and the minimal one is 1, for each $S \subseteq V(G)$ we have

$$\partial_O^+(S)/\partial_O^-(S) \leq r - 1. \quad (1)$$

As $b(S) = \partial_O^+(S) - \partial_O^-(S)$ and $\partial(S) = \partial_O^+(S) + \partial_O^-(S)$ we have $2 \cdot \partial_O^+(S) = \partial(S) + b(S)$ and $2 \cdot \partial_O^-(S) = \partial(S) - b(S)$. Thus we have

$$(\partial(S) + b(S))/(\partial(S) - b(S)) \leq r - 1. \quad (2)$$

¹The notion of the balanced valuation is a bit different than in Jaeger's paper. The change is made mostly to avoid unnecessary fractions.

We define the minimal r such that (1) holds as the *CFN* of O and the minimal r such that (2) holds as the *CFN* of b . Now we are ready to state Jaeger's result in our terms.

Theorem 1. [10] *Let G be a graph. Then the following statements are equivalent:*

- G admits a CNZ r -flow
- G admits an orientation with *CFN* at most r .
- G admits a balanced valuation with *CFN* at most r .

Turning a CNZ r -flow into an orientation with *CFN* at most r is easy and so is turning an orientation into a balanced valuation. Thus we focus on the opposite direction.

Theorem 2. *Let G be a graph and let b be a mapping from $V(G)$ to \mathbb{Z} . There exists an algorithm with running time $O(|E(G)|^{3/2})$ that takes G and b as the input and as the output, if b is a balanced valuation, it computes an orientation O of G with the same *CFN* as b ; otherwise it finds a vertex-set violating the definition of balanced valuation.*

Proof. It is easy to check if the parity condition holds for b . Thus assume that it does. Consider an orientation O such that b is the underlying balanced valuation of O . Note that in this case, the equations (1) and (2) are equivalent. Thus b and O necessarily have the same *CFN*. Thus it suffices to calculate any orientation which b underlies.

Let O_A be an arbitrary orientation of G and let b_A be its underlying balanced valuation. We construct an auxiliary graph G' and its orientation O' from G and O_A as follows. We start modifying G by adding two auxiliary vertices w_+ and w_- . We set O' to be such that each edge has the opposite orientation as in O_A . For each vertex $v \in V(G)$ we do the following. If $b_A(v) - b(v) > 0$, then we add $(b_A(v) - b(v))/2$ edges oriented from w_+ to v (as both b_A and b satisfy the parity condition, $b_A(v) - b(v)$ is even). If $b_A(v) - b(v) < 0$, then we add $(b(v) - b_A(v))/2$ edges oriented from v to w_- . If $b_A(v) - b(v) = 0$ we do not add any new edges. Note that the outdegree of w_+ equals the indegree of w_- . Also in this way the number of edges from w_+ to v minus the number of edges from v to w_- equals $(b_A(v) - b(v))/2$.

We solve the maximum flow problem on G' , O' with source w_+ and sink w_- with all capacities set to 1. As the capacities are integers so is some optimal flow. If the maximum flow transfers value $\partial_{O'}^-(w_+)$ from source to sink², then all edges incident with w_+ and w_- are saturated. Then we can get O from O_A by swapping the orientation of edges of G that were used in the integral optimal flow of G' , O' . To show that O defined in this way has b as its underlying balanced valuation consider the decomposition of the edges in the optimal flow into disjoint paths. Let $w_+v_1 \dots v_k w_-$ be one such a path. Reverting the orientation of $v_1v_2, \dots, v_{k-1}v_k$ in G , O_A affects the value of $\partial_{O_A}^+(v) - \partial_{O_A}^-(v)$ only for $v = v_1$ and $v = v_k$ and the value is changed by -2 and $+2$, respectively. Thus by doing this for all such paths the value is increased by $b(v) - b_A(v)$. Thus

$$\partial_O^+(v) - \partial_O^-(v) = b_A(v) + (b(v) - b_A(v)) = b(v),$$

²we do not specify the graph in our notation, if the orientation is O' , the graph is G' , otherwise the graph is G

as desired, for each $v \in V(G)$. Note that as b is underlying to some orientation, it is a balanced valuation.

On the other hand, if there is no flow in G' , O' that transfers value $\partial_{O'}^-(w_+)$, then there exists a cut that separates w_+ and w_- and this cut contains less than $\partial_{O'}^-(w_+)$ edges oriented from the part containing w_+ to the part containing w_- . Let S' be the part of the cut containing w_+ and let $S = S' - \{w_+\}$. Let V^+ (V^-) be the set of those vertices $v \in V(G)$ that have $b_A(v) - b(v)$ positive (negative). As the optimal flow does not saturate edges incident with w_+ we have $\partial_{O'}^-(w_+) > \partial_{O'}^-(S')$. Due to the construction of G' ,

$$\begin{aligned}\partial_{O'}^-(w_+) &= \sum_{v \in V^+} \frac{b_A(v) - b(v)}{2} \\ \partial_{O'}^-(S') &= \partial_{O_A}^+(S) + \sum_{v \in S \cap V^-} -\frac{b_A(v) - b(v)}{2} + \sum_{v \in V^+ - S} \frac{b_A(v) - b(v)}{2}\end{aligned}$$

Thus we have

$$\begin{aligned}\sum_{v \in S \cap V^+} \frac{b_A(v) - b(v)}{2} &> \partial_{O_A}^+(S) + \sum_{v \in S \cap V^-} -\frac{b_A(v) - b(v)}{2} \\ \sum_{v \in S} \frac{b_A(v) - b(v)}{2} &> \partial_{O_A}^+(S) \\ b_A(S) - b(S) &> 2\partial_{O_A}^+(S) \\ b_A(S) - b(S) &> \partial(S) + b_A(S) \\ -b(S) &> \partial(S) \\ b(V(G) - S) &> \partial(V(G) - S)\end{aligned}$$

Thus b is not a balanced valuation.

The time complexity of the algorithm is dominated by the construction of the maximum flow. As the capacities are 1 this can be done in $O(|E(G)|^{1.5})$ time [6]. \square

It is easy to find out if an orientation admits a flow, it just suffices to check if the graph is strongly connected (there exists a directed path between each pair of vertices). Thus we may assume that the orientation makes a strongly connected oriented graph. The problem of finding the flow number of an orientation can be easily solved by solving the following linear program.

$$\begin{aligned}\text{minimize} \quad & c \\ \text{subject to} \quad & 1 \leq x_e \leq c - 1 && \text{for each } e \in E(G) \\ & \sum_{e \text{ is inc. to } v} x_e - \sum_{e \text{ is outg. from } v} x_e = 0 && \text{for each } v \in V(G)\end{aligned}$$

Note that this linear program can be solved in strongly polynomial time [14]. We propose a different approach. We start with an auxiliary algorithm that tries to find a CNZ p/q -flow for given p and q .

Theorem 3. *Let G be a graph and let O be an orientation of G . Let p and q be two positive integers, $p \geq 2q$. Then there is an algorithm that either calculates a CNZ p/q -flow ϕ such that O is underlying ϕ or it decides that the CFN of O is more than p/q . The algorithm runs in $O(|E(G)|^{1.5} \cdot \log(|E(G)|))$ time.*

Proof. We construct an auxiliary graph G' with an orientation O' and a capacity labelling $c : E(G') \rightarrow \mathbb{N}$ as follows. We start with G and add two new vertices w_+ and w_- to the graph. For each $e \in E(G)$ we do the following. Assume e is oriented from u to v . We set $c(e) = p - 2q$. We add an edge e_+ from w_+ to v and an edge e_- from u to w_- , both with capacity q . We find a maximum flow on G' , O' , c with source w_+ and sink w_- . Let ϕ' be the optimal flow. If the value of the flow is $|E(G)| \cdot q$, then all edges incident with w_+ are saturated. We obtain the flow ϕ on G , O by setting $\phi(e) = \phi'(e)/q + 1 = (\phi'(e) + \phi'(e_+))/q$ for every $e \in E(G)$. This is a CNZ p/q -flow as each value is between 1 and $p/q - 1$.

On the other hand, if there is no such flow there exists a cut separating w_+ and w_- with capacity less than $|E(G)| \cdot q$. Let S' be the part of the cut containing w_+ and let $S = S' - \{w_+\}$. Let a be the number of edges between w_+ and a vertex not in S . Clearly, all the other edges incident to w_+ go to S thus the number of incoming edges to S is at least $\partial(w_+) - a = |E(G)| - a$ and their capacity is at least $(|E(G)| - a) \cdot q$. On the other hand, the capacity on outgoing edges from S' is less than $|E(G)| \cdot q$, thus the capacity on edges outgoing from S in G' is less than $(|E(G)| - a) \cdot q$. We conclude that the capacity of edges between w_+ and S is less than the capacity of edges outgoing from S . We will compute and compare these amounts in the following paragraphs.

Now consider the edges from G and examine the boundary of S in G' . If an edge e of G has both endpoints in S , then in G' there is an edge e^+ from w_+ to S and an edge e^- from S to w_- . Both have capacity q . If e starts in S but ends outside of S then e is outgoing from S' with capacity $p - 2q$, moreover, e^- is outgoing from S' with capacity q . Finally, if e starts outside of S and ends in S , then e^+ in G' goes from w_+ to S with capacity q . There is also an edge e in G' that incoming to S but it does not come from w_+ and thus is irrelevant in our computations.

Let a' be the number of edges in G that are inside S . The capacity of edges of G' between w_+ and a vertex of S is $a'q + \partial_O^+(S) \cdot q$. The capacity of edges outgoing S' is $a'q + \partial_O^-(S) \cdot (p - 2q + q)$. We have

$$\begin{aligned} a'q + \partial_O^+(S) \cdot q &> a'q + \partial_O^-(S) \cdot (p - 2q + q) \\ \partial_O^+(S)/\partial_O^-(S) &> p/q - 1. \end{aligned}$$

Thus the CFN of O is more than p/q .

The running time of the algorithm is dominated by the search of maximum flow. As capacities are integral we can use the algorithm of Goldberg and Gao [5]. Note that the CFN of G , O can be expressed as p^*/q^* with $q^* < |E(G)|$. Thus if $q \geq |E(G)|$ then instead of running the algorithm with p and q we can run the algorithm with p' and q' such that $p'/q' \leq p/q$, $q < |E(G)|$, and there is no rational number p^*/q^* with $q^* < |E(G)|$ with $p'/q' < p^*/q^* < p/q$. \square

Theorem 4. *Let G be a graph and let O be an orientation of G such that G with O is strongly connected. Then there is an algorithm that calculates the CFN r of O and a NZC r -flow ϕ such that O is the underlying orientation of ϕ and runs in time $O(|E(G)|^{1.5} \cdot \log^2(|E(G)|))$.*

Proof. Theorem 1 together with the definition of the flow number of an orientation imply that $r = p/q$, where $p, q \in \mathbb{N}$ and $q < |E(G)|$ ([12, Theorem 2.2] can be used to improve the constants). Similarly $r < |E(G)|$. Then we set $q = |E(G)|^2$ and using Theorem 3 and binary search we find p such that G, O has CFN at most p/q , and strictly more than $(p-1)/q$. As difference between two non equal fractions with denominator less than $|E(G)|$ is more than $1/|E(G)|^2$ there is only one rational number with denominator less than $|E(G)|$ in $((p-1)/q, p/q]$. This is the CFN of O . We find the flow by running the algorithm from Theorem 3 again. The algorithm requires $O(\log(E(G)))$ calls to the algorithm from Theorem 3, thus its running time is $O(|E(G)|^{1.5} \cdot \log^2(|E(G)|))$. \square

We believe that the running time can be improved to $O(|E(G)|^{1.5} \cdot \log(|E(G)|))$ by modifying the Goldberg-Gao algorithm.

3 Computing the circular flow number of a graph

Given the algorithms from the previous two theorems to calculate the CFN of a graph one just needs to calculate the minimal CFN over its balanced valuations. One needs to consider only balanced valuations b with $b(v) < |\partial(\{v\})|$ for each vertex v . Otherwise all edges incident with v would be oriented either towards or away from v and this orientation would not be underlying to any CNZ flow. This approach requires $O(\prod_{v \in V(G)} (\partial(v) - 1))$ calls to the algorithm from Theorem 4. Note that it is significantly better to consider balanced valuations instead of orientations. E.g. if G is 9-regular, it has $\approx 22.63^{|V(G)|}$ orientations and one needs to consider at most $8^{|V(G)|}$ mappings as a balanced valuation.

For cubic graphs we just need to consider balanced valuations with values in $\{-1, 1\}$. A balanced valuation b of a cubic graph G is a k -bisection if

- vertices v with $b(v) = 1$ induce no component containing more than k vertices and
- vertices v with $b(v) = -1$ induce no component containing more than k vertices.

Note that if b is not a 3-bisection, its CFN is at least 6. As each bridgeless graph admits a nowhere-zero 6-flow [13], we do not need to consider balanced valuations that are not 3-bisections. Considering this fact we can get some improvement over the trivial $2^{|V(G)|}$ generation algorithm, however none of the methods we are aware of provides a significant speedup. Instead of that we will focus on graphs with CFN less than 5.

Tutte's 5-flow conjecture asserts that $\phi_c(G) \leq 5$. A balanced valuation with CFN less than 5 is a 2-bisection. Thus if $\phi_c(G) < 5$, then one has to consider only 2-bisections. In the next section we show how to generate all of them in $O(2^{0.6 \cdot |V(G)|})$ time. To check if $\phi_c(G) \leq 5$ one only needs to consider integral flows [3]. This can be done using dynamic programming over path decompositions provided by the algorithm of Fomin and Høie [2]

(this runs in $O(2^{0.34 \cdot |V(G)|})$). Thus we can also recognise graphs with $\phi_c \leq 5$ and determine its CFN in $O(2^{0.6 \cdot |V(G)|})$.

4 Enumerating 2-bisections

In this section we show how to enumerate all 2-bisections of a cubic graph G in $O(2^{0.6 \cdot |V(G)|})$. This is also a bound on the number of 2-bisections. In contrast to the previous sections, we assume here that G is simple. Note that this does not make problems in our CNZ flow algorithm as for cubic graphs with a 2-circuit one can simply contract the edges incident to one of its vertices. This produces cubic graph with the same CFN (one can easily extend a CNZ r -flow, $r \geq 3$, of the contracted graph to the original graph and cubic graphs do not admit CNZ r -flows with $r < 3$). We also assume that $G \neq K_4$.

This is our algorithm:

1. Find a proper 3-vertex-colouring of G .
2. Extend one colour class into a maximal (with respect to inclusion) independent set. The graph induced by the remaining vertices is an union of paths (possibly containing only one vertex) and even circuits.
3. Recursively generate all possible (not necessarily proper) 2-vertex-colourings of the paths and cycles such that there are no three consecutive edges of the same colour.
4. For each such colouring we do the following.
 - (a) We extend the colouring of the paths and circuits to the independent set. As all three neighbouring vertices are already coloured, two of them has to have the same colour and thus we have to colour the vertex with the other colour. We check if this colour does not introduce a monochromatic connected subgraph on three vertices.
 - (b) We check if the colouring is a bipartition (we keep track of the number of vertices of each colour during the generation process so that this check is done in constant time).

An easy recursive enumeration shows, that the number of improper 2-colourings avoiding three consecutive vertices of the same colour on a path with i vertices is $2F_{i+1}$ (F_i denotes the i -th Fibonacci number). The number of 2-colourings of cycles on i vertices is at most $2F_{i+1}$ (for the exact value see Corollary in [1]). For cycles of length 4 we use the exact value 6.

Consider the complement of the independent set from Step 2 of the algorithm. Let p_i be the number of paths on i vertices and let c_i be the number of cycles on i vertices. Then the number of colourings c is at most

$$\prod_{i=1}^{\infty} (2F_{i+1})^{p_i} \cdot 6^{c_4} \cdot \prod_{i=3}^{\infty} (2F_{2i+1})^{c_{2i}}.$$

The number of vertices of such a graph is

$$n = \sum_{i=1}^{\infty} (ip_i + \frac{i+2}{3} \cdot p_i) + \sum_{i=2}^{\infty} (2ic_{2i} + \frac{2i}{3} \cdot c_{2i}) = \sum_{i=1}^{\infty} \frac{4i+2}{3} \cdot p_i + \sum_{i=2}^{\infty} \frac{8i}{3} \cdot c_{2i}.$$

After some calculation

$$\begin{aligned} \lg(c) &= \lg(6)c_4 + \sum_{i=1}^{\infty} \lg(2F_{i+1})p_i + \sum_{i=3}^{\infty} \lg(2F_{2i+1})c_{2i} \leq \\ &\leq 0.6\frac{16}{3}c_4 + \sum_{i=1}^{\infty} 0.6\frac{4i+2}{3}p_i + \sum_{i=3}^{\infty} 0.6\frac{8i}{3}c_{2i} \leq 0.6n. \end{aligned} \quad (3)$$

5 Concluding remarks

First, we sketch how the running time of our algorithms to calculate the CFN can be marginally improved. We do not provide any involved analysis, we believe that it is obvious that the ideas we present provide a minor improvement of the exponents in the running time. This justifies dropping the polynomial terms in the time complexities given in the introduction. We decided not to pursue these improvements, first, because the improvement is barely worth the increased complexity, and second, there is a good chance that there exists a completely different more efficient approach to generate 2-bisections we are not aware of.

We continue by sketching how small improvements to the presented algorithms can be made:

Non-cubic case: Thanks to the Seymour's 6-flow theorem [13], we need to examine only balanced valuations that have $|b(v)| < 5/6 \cdot \partial(v)$ for each vertex. This improves the running time if many edges of G are incident with vertices of degree more than 6. On the other hand, if this is not the case, then we can merge many vertices of small degree into quadruples of consecutive vertices. Again, due to Seymour's theorem it is not possible that all four vertices v will have $b(v) = \partial(v) - 2$, and the same holds for $b(v) = -(\partial(v) - 2)$, reducing the number of mappings we need to consider.

Cubic case: The equality in the bound (3) is attained if all components in the complement of the independent set are paths on two vertices. Thus if there aren't many such components, the running time improves. On the other hand if almost all such components are paths on two vertices, then there are many vertices in the independent set neighbouring with at least two such paths. Note that if two vertices of first path belong to one partite and the two vertices of the second path belong to the other one we cannot colour the neighbouring vertex from the independent set. This, again, reduces the number of mappings we need to consider. A finer analysis of this argument can be obtained by discharging. We believe that using these (and several similar) arguments one can improve the running time to approximately $O(2^{0.56 \cdot |V(G)|})$.

A circular r -colouring is a vertex labelling $c : V(G) \rightarrow \mathbb{R}/r\mathbb{Z}$ such that for each two neighbouring vertices u and v , we have $c(u) - c(v) \in [1, r - 1]$. As circular colourings

are dual to circular flows we can use our algorithms to determine the circular colouring number of planar graphs. Our algorithm runs most efficiently on cubic graphs, hence the dual algorithm is most efficient on triangulations. If the graph is not planar we still can use orientations. Similarly as with flow number one can define the *circular colouring number* of an orientation O as $\min_{C \in \mathcal{C}(G)} |E(C)|/|E_O^+(C)|$, where $\mathcal{C}(G)$ denotes the set of oriented circuits in the unoriented graph G (we just take a circuit of G and we assign one of the two orientation of it) and $E_O^+(C)$ denotes the set of edges of C that have the same orientation in O . We will call $|E(C)|/|E_O^+(C)|$ as the *unbalancednesses* of C . Goddyn, Tarsi, and Zhang [3] proved that G admits a circular r -colouring if and only if G admits an orientation with circular colouring number at most r . Given an orientation one can determine its circular chromatic number using linear programming. An interesting question is, whether there exists notion for circular colourings equivalent to the notion of balanced valuation for flows that works also for non-planar graphs.

References

- [1] Z. Agur, A. S. Fraenkel, S. T. Klein: The number of fixed points of the majority rule. *Discrete Math.*, 70(3): 295–302, 1988.
- [2] F. V. Fomin, K. Høie: Pathwidth of cubic graphs and exact algorithms. *Inform. Process. Letters*, 97(5):191–196, 2006.
- [3] L. A. Goddyn, M. Tarsi, C.-Q. Zhang: On (k, d) -colorings and fractional nowhere-zero flows. *J. Graph Theory*, 28(3):155–161, 1998.
- [4] J. Goedgebeur, D. Mattiolo, G. Mazzuocolo: Computational results and new bounds for the circular flow number of snarks. *Discrete Math.*, 343(10):112026, 2020.
- [5] A. V. Goldberg, S. Rao: Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, 1998.
- [6] A. V. Goldberg, R. E. Tarjan: Efficient maximum flow algorithms. *Communications of the ACM*, 57(8):82–89, 2014.
- [7] S. L. Hakimi: On the degrees of the vertices of a directed graph. *J. Franklin Inst.*, 279(4):290–308, 1965.
- [8] M. Han, et al.: Counterexamples to Jaeger’s circular flow conjecture. *J. Combin. Theory Ser. B*, 131:1–11, 2018.
- [9] I. Holyer: The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981.
- [10] F. Jaeger: Balanced valuations and flows in multigraphs. *Proc. Amer. Math. Soc.*, 55(1):237–242, 1976.
- [11] F. Jaeger: On circular flows in graphs. Finite and infinite sets. *Colloquia Mathematica Societatis János Bolyai*, 37:391–402, 1984.
- [12] R. Lukotka, M. Škoviera: Real flow number and the cycle rank of a graph. *J. Graph Theory*, 59(1):11–16, 2008.

- [13] P. D. Seymour: Nowhere-zero 6-flows. *J. Combin. Theory Ser. B*, 30(2):130–135, 1981.
- [14] E. Tardos: A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34(2):250–256, 1986.