Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2022-06. Published by the Egerváry Research Group, Pázmány Péter stny. 1/C, H–1117, Budapest, Hungary. Web site: egres.elte.hu. ISSN 1587–4451.

On vertex-coloring $\{a, b\}$-edge-weightings of graphs

Péter Madarasi and Máté Simon

May 2022

On vertex-coloring $\{a, b\}$-edge-weightings of graphs

Péter Madarasi⋆ and Máté Simon⋆⋆

Abstract

For a given graph $G = (V, E)$, an $\{a, b\}$-edge-weighting is an assignment $w : E \to \{a, b\}$, which we call proper if the induced labeling $z : V \to \mathbb{Z}$ is a proper vertex coloring of G, where a, b are distinct integers and $z(v) = \sum_{e \in \Delta(v)} w(e)$.

Dudek and Wajc [1] proved that deciding whether a given graph G has a proper $\{1, 2\}$-edge-weighting is NP-complete. Strengthening their result, we show that the problem is NP-complete for any distinct integers a and b.

Thomassen, Wu and Zhang [2] gave a polynomial-time algorithm to decide whether a given bipartite graph has a proper $\{1, 2\}$-edge-weighting. We consider a natural generalization of this problem when a partial edge-weighting is to be extended, which is shown to be NP-complete for any distinct integers a, b. We also prove that the problem is solvable in polynomial time for trees.

Keywords: 1-2-3 conjecture, $\{a, b\}$-edge-weighting, NP-completeness, Irregular graphs, Graph coloring

1 Introduction

Throughout this paper, $G = (V, E)$ denotes a simple, finite, undirected graph. A $\{1, \ldots, k\}$-*edge-weighting* is an assignment w which assigns numbers from $\{1, \ldots, k\}$ to the edges of G. We say that an edge-weighting is *proper* or *feasible* if the induced vertex coloring $z : V \to \mathbb{Z}$, where $z(v) = \sum_{e \in \Delta(v)} w(e)$, is a proper coloring, that is $z(u) \neq z(v)$ holds for every edge $uv \in E$. If G has a proper $\{1, 2, 3\}$-edge-weighting, then we say that G has the 1-2-3 *property*, which can be similarly defined for any other weight set as well.

⋆Department of Operations Research, ELTE Eötvös Loránd University, and the MTA-ELTE Egerváry Research Group on Combinatorial Optimization, Eötvös Loránd Research Network (ELKH), Pázmány Péter sétány 1/C, 1117 Budapest, Hungary. E-mail: `madarasip@staff.elte.hu`

⋆⋆Department of Operations Research, ELTE Eötvös Loránd University, and the MTA-ELTE Momentum Matroid Optimization Research Group, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary. E-mail: `matesimon@student.elte.hu`

Karoński, Łuczak and Thomason formulated the so-called 1-2-3 conjecture in 2004 [3], which states that every simple graph without isolated edges has the 1-2-3 property. This conjecture fostered several new interesting questions. The focus of the present paper is on one of these questions, the existence of $\{a, b\}$-edge-weightings.

In 2011, Dudek and Wajc [1] proved that deciding whether a given graph has the 1-2 property is NP-complete. To the best of our knowledge, the more general problem when we ask if a feasible $\{a, b\}$-edge-weighting exists, remained open thus far. As an extension of the results of Dudek and Wajc, Section 2 proves that their statement also holds for arbitrary a and b.

Furthermore, in 2016 Thomassen, Wu and Zhang [2] proved that deciding whether a given bipartite graph has the 1-2 property is possible in polynomial-time. More precisely, they proved that a bipartite graph has the 1-2 property if and only if it is not a so-called odd multi-cactus. In fact, their approach also extends to $\{a, b\}$-edge-weightings provided that $a < b$, a is odd and b is even. Recently, Lyngise showed that exactly the odd multi-cacti have no proper edge-weightings for 2-connected bipartite graphs when a is odd and $b = a + 2$ [4], and also for bridgeless bipartite graphs when $a = 0$ and $b = 1$ [5]. Based on these positive results, we investigate whether a partial edge-weighting is extendable such that the resultant edge-weighting is proper. In Section 3, we prove that this more general version of the problem is NP-complete even for bipartite graphs, however, it is polynomial-time solvable for trees. The latter statement will be proven by giving a dynamic programming algorithm which runs in polynomial-time. As a special case, this implies an alternative polynomial-time algorithm to decide whether a tree has the 0-1 property, which was first solved in [5].

The next section gives a brief overview of some of the problems and results related to the 1-2-3 conjecture.

1.1 Motivation and previous results

The question of the existence of $\{a, b\}$-edge-weightings was inspired by the 1-2-3 conjecture, which itself comes from the study of graph "irregularity". By simple graph theoretic observations, one can easily show that there exists no "opposite" of a simple regular graph, that is, a simple graph with all-different degrees. Chartrand et al. [6], tried to measure how irregular a graph is. In particular, they investigated the smallest value k such that by replacing each edge with at most k parallel edges, the resulting G' multigraph becomes irregular (that is each vertex has a different degree). The minimum value of k is called the irregularity strength of G, for further information on this topic see [7], [8] and [9]. Another possible approach is when we do not require that all vertices in the resulting multigraph have different degrees, but only that the degrees of the adjacent nodes are different. Notice that instead of edge multiplication, we can look for a proper $\{1, \dots, k\}$-edge-weighting. Exchanging the weight set $\{1, \dots, k\}$ to $\{a, b\}$, we obtain the $\{a, b\}$-edge-weighting problem.

Early articles and results, such as in which the 1-2-3 conjecture was first introduced [3], focus on the relationship between $\chi(G)$ and $\chi_{\Sigma}(G)$, where $\chi_{\Sigma}(G)$ is the smallest integer k for which a proper $\{1,\dots,k\}$-edge-weighting exists in G. One of the first results from [3] states the following:

Claim 1.1. *Let $(\Gamma,+)$ be a finite abelian group of odd order, and G be a $|\Gamma|$-colorable graph without isolated edges. Then there exists an edge-weighting of G with the elements of Γ such that the resultant induced vertex coloring is proper.*

Further results in connection with the chromatic number [10]:

Claim 1.2. *If G is 2-connected and $\chi(G) \geq 3$, then $\chi_{\sum}(G) \leq \chi(G)$. Moreover, for every integer $k \geq 3$ and any graph G without isolated edges the following hold:*

1. *(Karoński, Łuczak, Thomason [3]) If G is k-colorable for odd k, then $\chi_{\sum}(G) \leq k$;*
2. *(Duan, Lu, Yu [11]) If G is k-colorable for $k \equiv 0 \pmod 4$, then $\chi_{\sum}(G) \leq k$;*
3. *(Lu, Yang, Yu [12]) If G is k-colorable, 2-connected and has minimum degree at least $k+1$ for $k \equiv 2 \pmod 4$, then $\chi_{\sum}(G) \leq k$.*

The first general upper bound for $\chi_{\sum}(G)$ was given by Addario-Berry, Dalal, McDiarmid, Reed and Thomason [13], who proved that $\chi_{\sum}(G) \leq 30$. Their method is based on the investigation of the so-called degree-constrained subgraph problem, which was further refined by Addario-Berry, Dalal and Reed [14], who managed to improve this upper bound to 16, then Wang and Yu [15] further improved it to 13. The best known upper bound is due to Kalkowski, Karoński and Pfender [16], who proved that $\chi_{\sum}(G) \leq 5$ holds. In other words, every graph without isolated edges has the 1-2-3-4-5 property.
Moreover, it is also known that the 1-2-3 conjecture holds if G is large and dense enough: there exists a constant n' such that every graph $G = (V,E)$ with at least n' nodes has the 1-2-3 property if the degree of every node is at least $0.099985|V|$ [17]. Furthermore, it is known that if G is a random graph (according to the Erdős-Rényi model), then it has the 1-2 property asymptotically almost surely, see [14]. If we restrict ourselves to regular graphs, then Jakob Przybylo achieved the most significant progress [18], namely, every regular graph has the 1-2-3-4 property, and the 1-2-3 conjecture holds if $d \geq 10^8$ and G is d-regular. On the other hand, Dudek and Wajc [1] showed that deciding whether a given graph has the 1-2 property is NP-complete. However, based on the result of Thomassen, Wu and Zhang [2], this problem can be solved in polynomial time for bipartite graphs.

One might also define other kinds of weightings. For example, in the node-weighting problem, we want to assign weights to the nodes (instead of the edges) and the labels of the nodes are defined as the sum of the weights of their neighbours. It was shown in [19] that deciding whether a graph G has a proper node-weighting from the set $\{1,\dots,k\}$ is NP-complete for any $k \geq 2$. This result holds even if we restrict ourselves to 3-colorable planar graphs and $k = 2$. Furthermore, it is also NP-complete for 3-regular graphs in case of $k = 2$ [20].

Other problems can be obtained by modifying the definition of the labels of the nodes. For example, one can take the product of the weights instead of their sum.

This way, we obtain the problems called *edge-weighting by product* and *node-weighting by product*. Let us briefly summarize some of the hardness results related to these problems. It is NP-complete to decide whether a given 3-regular planar graph has a proper edge-weighting by product from the set $\{1, 2\}$ [21]. It was shown in [21] that deciding the existence of $\{1, 2\}$-node-weighting by product is NP-complete for 3-colorable planar graphs. Moreover, if we omit the planarity and colorability conditions but the weights can be chosen from set $\{1, \ldots, k\}$ for some $k \geq 3$, then we still get an NP-complete problem.

2 $\{a, b\}$-edge-weightings in general graphs

In 2011, Dudek and Wajc [1] proved that deciding whether a given graph G has the 1-2 property is NP-complete. In this section, we extend this result and prove that the statement holds for arbitrary a and b. First, we consider the case when $a \neq -1$ and $b \neq 1$. Then, a fundamentally different reduction will be given to deal with this exceptional case.

The following claim shows that we can restrict ourselves to the case when a and b are integers and relative primes.

Claim 2.1. *Let a, b be a rational pair. Then for every $d \neq 0$, there is one-to-one correspondence between proper $\{a, b\}$-edge-weightings and proper $\{ad, bd\}$-edge-weightings.*

This simple claim holds, since multiplication by $d \neq 0$ on all of the edges does not change the feasibility of an edge-weighting. We say that a and b are *relevant* if they are integers, relative primes, at most one of them is negative, $a \neq b$ and $|b| \geq |a|$. By Claim 2.1, we can assume without loss of generality that a, b are relevant whenever we consider $\{a, b\}$-edge-weightings.

The main result of this paper is the following:

Theorem 2.2. *Let a and b be relevant numbers, and let G be an arbitrary simple graph. Then it is NP-complete to decide whether a proper $\{a, b\}$-edge-weighting exists.*

The proof of this theorem consists of two parts. First, Theorem 2.3 extends the proof of Dudek and Wajc to the case when $a \neq -1$ and $b \neq 1$. Second, the proof of Theorem 2.6 gives a different reduction for $a = -1$, $b = 1$.

Theorem 2.3. *Let a and b be relevant numbers such that $a \neq -1$ and $b \neq 1$ holds. Then it is NP-complete to decide whether a proper $\{a, b\}$-edge-weighting exists.*

Proof. The case $a = 0, b = 1$ is settled in [1], so we can assume that $b \neq 1$. The problem is in NP, since one can easily decide in polynomial time if a given $\{a, b\}$-edge-weighting is feasible. Similarly to the proof in [1], we give a reduction from the NP-complete 3-COLOR problem, in which we are given a graph $G' = (V', E')$ and we want to decide if its nodes can be colored with 3 colors such that the colors of any two adjacent vertices are different. In what follows, we construct a graph G which has a feasible $\{a, b\}$-edge-weighting if and only if the vertices of G' can be colored with three colors.

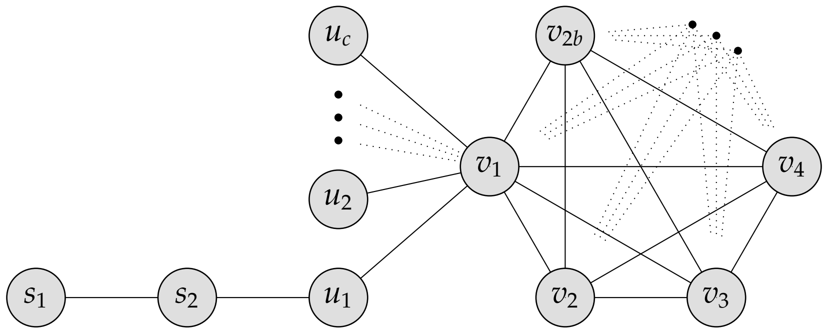

Figure 1: a-forcing gadget

For this purpose we need two gadgets. The first one is called a-forcing gadget, which is defined in the following way. Take a complete graph on nodes $v_1, \dots, v_{2b}$, and connect c new leaf nodes $u_1, \dots u_c$ to v_1, where $c = b - a$. Furthermore, add two new nodes s_1, s_2 with an edge between them, and connect s_2 to u_1. Figure 1 illustrates the construction.

Claim 2.4. *In any feasible $\{a, b\}$-edge-weighting of the a-forcing gadget, the weight of edge s_1s_2 is a. This also holds when the gadget is glued to any graph along node s_1.*

Proof. As the degrees of nodes $v_2, \dots, v_{2b}$ are all $2b - 1$, their possible labels in any feasible $\{a, b\}$-edge-weighting are of the form $xa + (2b - 1 - x)b$ for some $x \in \{0, \dots, 2b - 1\}$. In addition, the labels of any two of these nodes must be distinct as they induce a complete graph. Furthermore, labels $(2b - 1)a$ and $(2b - 1)b$ may not appear simultaneously, because these correspond to the cases $x = 0$ and $x = 2b - 1$, respectively, which would in turn mean that the incident edges of a node are all weighted a, while the incident edges of another node are all weighted b, which is clearly not possible. In fact, the number of nodes at hand is $2b - 1$ and the number of possible labels are $2b$, therefore exactly one of labels $(2b - 1)a$ and $(2b - 1)b$ must appear.

First, consider the case when $(2b - 1)a$ appears as one of the labels of nodes $v_2, \dots, v_{2b}$ and $(2b - 1)b$ does not. By symmetry, one can prescribe that the label of v_i is

$$z(v_i) = (i - 1)a + (2b - i)b$$

for $i = 2, \dots, 2b$. We prove by induction on b that there is a unique $\{a, b\}$-edge-weighting which generates these very labels, and the sum of the weights of the edges induced by v_1 and some node $v_2, \dots, v_{2b}$ is $ba + (b - 1)b$. For $b = 1$, the claim holds because the label of v_2 is prescribed to be a, hence the only edge incident to it must have weight a. For the inductive step, assume that $b \geq 2$ and the statement holds for $b - 1$. The labels of v_2 and v_{2b} are prescribed to be $a + (2b - 2)b$ and $(2b - 1)a$, respectively. Therefore the weights of all edges incident to v_{2b} must be a, and hence the weights of all edges incident to v_2 must be b except for edge $v_{2b}v_2$, which is already set to a. Observe that removing v_2 and v_{2b}, one can inductively apply the statement for $v_3, \dots, v_{2b-1}$ where the label of v_i is prescribed to be $(i - 2)a + (2b - i - 1)b$.

By induction, we get that there exists a unique $\{a, b\}$-edge-weighting in this smaller instance, and also that the sum of the weights of the edges induced by v_1 and some node $v_3, \ldots, v_{2b-1}$ is $(b-1)a + (b-1-1)b$. Putting back nodes v_2 and v_{2b}, one gets that there exists a unique edge-weighting with the prescribed labels, because the weights of the edges incident to v_2 and v_{2b} are uniquely determined by the label-prescription, and the uniqueness of the weighting of the rest of the edges follows by the inductive step. Furthermore, the sum of the weights of the edges induced by v_1 and some node $v_2, \ldots, v_{2b}$ is $(b-1)a+(b-1-1)b+a+b = ba+(b-1)b$, which was to be proven. This immediately implies that the only valid setting of the weight of edge u_jv_1 is b for all $j = 1, \ldots, c$, since if the weight of exactly j such edges are a, then the label of v_1 would be $ba+(b-1)b+ja+(b-a-j)b = ja+(2b-1-j)b$, which does not conflict with any of the labels of $v_2, \ldots, v_{2b}$ only if $j = 0$. Meaning that the weight of edge u_jv_1 must be b for all $j = 1, \ldots, c$.

In the second case, $(2b-1)b$ is one of the labels of nodes $v_2, \ldots, v_{2b}$ and $(2b-1)a$ is not. Along the lines of the first case, one can prove by induction on b that the sum of the weights of the edges induced by v_1 and some node $v_2, \ldots, v_{2b}$ is $bb + (b-1)a$. But just like in the previous case, this means that the weight of edge u_jv_1 must be b for all $j = 1, \ldots, c$, otherwise there would exist a vertex $u \in v_2, \ldots, v_{2b}$ whose label is the same as the label of v_1.

In both cases, the weight of edge u_jv_1 is b for all $j = 1, \ldots, c$. Hence the weight of s_1s_2 must be a, and that of s_2u_1 can be always chosen appropriately. Furthermore, we did not utilize that s_1 is a leaf node, hence the second part of the claim follows as well, which completes the proof. □

Notice that by removing nodes s_1 and s_2 along with the two incident edges, we obtain the so-called b-forcing gadget, in which the weight of edge v_1u_1 must be b in any feasible $\{a, b\}$-edge-weighting.

We need one more gadget for the reduction, which is called k-excluding gadget. The k-excluding gadget will have a root node, which will be unified with a vertex of the original graph, and it achieves that the label of the root node cannot be k. For a, b and $k = xa + yb$ $(x, y \in \mathbb{N})$, the construction of the k-excluding gadget is as follows. Start with the root node r and two more nodes u, v. Add edges ru, uv and rv, so that we get a triangle. Add $(x-1)$ copies of the a-forcing gadgets and $(y-1)$ copies of the b-forcing gadgets, and unify all the s_1 nodes of the a-forcing gadgets and all the u_1 nodes of the b-forcing gadgets with node u. Repeat the same procedure for node v instead of u. Notice that the label of node r must be different from k, because regardless of how we weight edge uv, one of ru and rv must have weight a and the other one must have weight b. In addition, one can also show that the gadget forbids no other labels at node r — which will be useful when we glue the gadget to other graphs.

Now we have all the tools required to define a graph G which has a proper $\{a, b\}$-edge-weighting if and only if the nodes of G' can be colored with 3 colors. Let $G' = (V', E')$ and $n = |V'|$. Without loss of generality, suppose that $n \geq 3$. Our graph G will be obtained by extending the original graph G' with additional edges and vertices as follows.

To define G, modify G' in the following way. For every $v \in V'$

1. Add two new vertices s_v and t_v, then connect them to v.
2. Let U_v be a new vertex set of size $n - 1 - d_{G'}(v)$ and connect every element of U_v to v.
3. Add $n - 1$ new k-excluding gadgets for every $k = xa + yb$, where $x + y = 3n - 1$ and y runs from $n + 2$ to $2n$. Unify the roots of these k-excluding gadgets with v.

Notice that the construction of G can be done in polynomial-time, since a and b are constants. Next, we formulate an important lemma concerning the possible labels of those nodes in G which were inherited from the original graph G'.

Lemma 2.5. *Let $G = (V, E)$ be the graph we obtained above. Then in every feasible $\{a, b\}$-edge-weighting of G, the following holds for every $v \in V \cap V'$:*

$$z(v) \in \{2na + (n - 1)b, (2n - 1)a + nb, (2n - 2)a + (n + 1)b\}.$$

Proof. Let $v \in V \cap V'$ an arbitrary node. Then $w(s_v v) + w(vt_v) \in \{2a, a + b, 2b\}$, where s_v and t_v are the nodes defined above in 1. By the construction of G, there are $n - 1$ edges incident to v which go to $U_v \cup (V \cap V')$. Furthermore, each of the $n - 1$ copies of the k-excluding gadget connected to v adds $a + b$ to the label of v. Since there are no other edges incident to v, it follows that

$$\{2a, a + b, 2b\} + \{xa + yb \mid x + y = n - 1,\ y = 0, \dots, n - 1\} + (a + b)(n - 1) = \tag{1}$$
$$= \{xa + yb \mid x + y = 3n - 1,\ y = n - 1, \dots, 2n\},$$

where by the sum of two or more sets we now mean the set whose elements can be obtained by taking one element from each set and then adding them together. Observe that $z(v) \notin \{xa + yb \mid x + y = 3n - 1,\ y = n + 2, \dots, 2n\}$ due to the k-excluding gadgets incident to v, according to 3. From this, we get that

$$z(v) \in \{2na + (n - 1)b, (2n - 1)a + nb, (2n - 2)a + (n + 1)b\},$$

which was to be proven. □

It remains to show that the nodes of G' can be colored with 3 colors if and only if there exists a feasible $\{a, b\}$-edge-weighting in G.

First suppose that the nodes of G' can be colored with 3 colors. Without loss of generality let the colors be $2na+(n-1)b$, $(2n-1)a+nb$, and $(2n-2)a+(n+1)b$. We will show that the edges of G can be weighted so that the induced labels obtained by this weighting match the original colors of the vertices, thus meaning that there exists a proper edge-weighting. We weight the edges as follows: For every $e \in E \cap E'$, let $w(e) = a$. For every edge which is incident to the vertices in U_v, let $w(e) = a$. Moreover, for every $v \in V \cap V'$,

- if $\chi(v) = 2na + (n - 1)b$, then let $w(vs_v) = w(vt_v) = a$;

- if $\chi(v) = (2n-1)a + nb$, then let $w(vs_v) = a$ and $w(vt_v) = b$;
- otherwise, $\chi(v) = (2n-2)a + (n+1)b$, when we set $w(vs_v) = w(vt_v) = b$,

where $\chi(v)$ is the color of v.

This (partial) edge-weighting can be extended to the rest of the edges, since we have already seen that the k-excluding gadget forbids only label k at its root node. Moreover, the weight of edge s_2u_1 in every a-forcing gadget can be chosen properly, and similarly, the edges of the b-forcing gadgets can be weighted in two different ways — one of which will be always feasible. We now prove that the edge-weighting obtained as defined above is indeed proper. For any $v \in V' \cap V$ there are $(n-1)$ edges coming from U_v or inside of G' with weight a. From every k-excluding gadget, two edges come into v: one with weight a, the other one with weight b, and there are $(n-1)$ of them in total. Thus, adjusting the weights of the edges coming from s_v and t_v, we can achieve that $z(v) = \chi(v)$. We have seen above that the weights of the rest of the edges of the gadgets can be chosen properly as well. Lastly, every other node has degree one, while the degree of their neighbours are strictly greater than one. So $z(u) \neq z(v)$ for any edge uv in G, that is, w is a proper $\{a, b\}$-edge-weighting.

Second, if G cannot be colored with 3 colors, then there is no feasible $\{a, b\}$-edge-weighting of G by Lemma 2.5. □

Next, we settle the case of $a = -1, b = 1$.

Theorem 2.6. *Let $G = (V, E)$ be a simple graph. It is NP-complete to decide if G has a $\{-1, 1\}$-edge-weighting.*

Proof. Clearly, the problem is in NP, since one can easily decide in polynomial time if a given edge-weighting is feasible. To prove the hardness of the problem, we give a reduction from the NP-complete NAE-3SAT3 problem. Here we are given a conjunctive normal form in which each clause is of size 3 and each variable appears exactly 3 times. The goal is to find an assignment of the variables such that every clause has at least one true and at least one false literal. The hardness of this problem immediately follows from the NP-completeness of the Monotone NAE-3SAT problem in which every variable appears in exactly 3 clauses, every literal is positive, and the size of each clause is either 2 or 3 [22]. We construct a graph G which has a proper $\{-1, 1\}$-edge-weighting if and only if the NAE-3SAT3 instance has a feasible solution. To this end, we need two gadgets.

The first gadget is the so-called $\{-3, 3\}$-excluding graph, which is shown in Figure 2. We claim that, in any feasible $\{-1, 1\}$-edge-weighting, exactly one of v_2 and v_9 has label 4, the other one -4. Edge v_7v_8 ensures that the labels of edges v_2v_7 and v_8v_9 are different, that is, $w(v_2v_7) = -w(v_8v_9)$. By symmetry, we can assume that $w(v_2v_7) = 1$. Observe that $w(v_1v_2) = 1$ — otherwise $w(v_1v_2) + w(v_2v_7) = 0$ would imply that the C_5 incident to v_2 has a feasible edge-weighting, which is not possible. Simple enumeration of cases shows that $w(v_2v_3) = w(v_2v_6) = 1$:

- If $w(v_2v_3) = w(v_2v_6) = 1$, then setting $w(v_3v_4) = 1$, $w(v_4v_5) = -1$ and $w(v_5v_6) = -1$ one obtains a feasible edge-weighting.

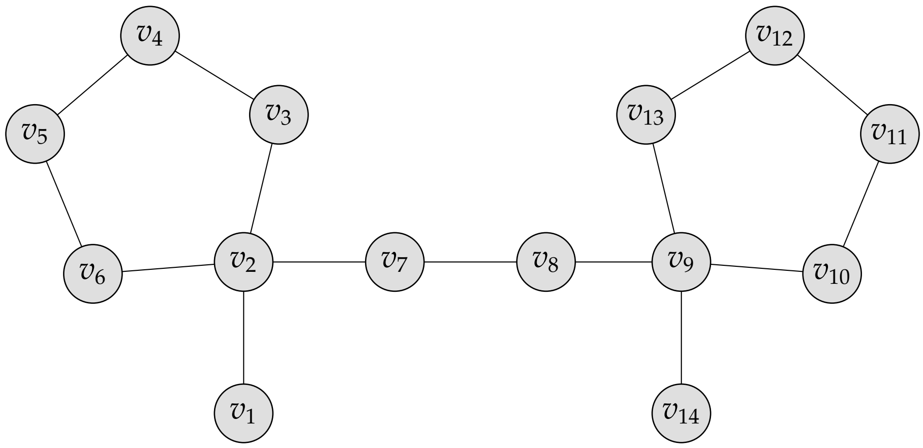

Figure 2: $\{-3, 3\}$-excluding gadget

- If $w(v_2v_3) \neq w(v_2v_6)$, then we can assume that $w(v_2v_3) = 1$ and $w(v_2v_6) = -1$. As $w(v_1v_2) = 1$ and $w(v_2v_7) = 1$, it follows that $w(v_3v_4) = -1$ and $w(v_4v_5) = -1$, but then neither $w(v_5v_6) = 1$ nor $w(v_5v_6) = -1$ is feasible.
- If $w(v_2v_3) = w(v_2v_6) = -1$, then $w(v_3v_4) = -1, w(v_4v_5) = 1$ follows and hence neither $w(v_5v_6) = 1$ nor $w(v_5v_6) = -1$ is feasible.

Therefore, $w(v_2v_3) = w(v_2v_6) = 1$ follows. This means that $z(v_2) = 4$ and, by symmetry, $z(v_9) = -4$. Hence, one of v_2 and v_9 has label 4, and the other one has label -4, as we claimed. Moreover, there exists a feasible $\{-1, 1\}$-edge-weighting of the $\{-3, 3\}$-excluding gadget such that the label of v_2 is 4 and that of v_9 is -4. The opposite of this weighting is also feasible and the label of v_2 is -4 and that of v_9 is 4.

The second gadget, the so-called 6-equal graph, is shown in Figure 3. We claim that the weights of the six leaf edges must be either all -1 or all 1 in any $\{-1, 1\}$-edge-weighting. First, observe that the possible labels of nodes $v_2, \ldots, v_7$ are distinct values from the set $\{-6, -4, -2, 0, 2, 4, 6\}$. Furthermore, -6 and 6 may not appear simultaneously, because we cannot have two nodes whose incident edges are either all 1 or all -1. This also means that all labels $\{-4, -2, 0, 2, 4\}$ must appear. Assume that label 6 appears among the labels and -6 does not. By symmetry, we can also assume that the labels of $v_2, \ldots, v_7$ are $-4, -2, 0, 2, 4, 6$, respectively. This means that exactly i edges among the six edges incident to v_i have weight 1 for $i = 2, \ldots, 7$. This immediately implies that the sum of the weights of the edges between v_1 and $v_2, \ldots, v_7$ are zero, and hence the sum of the weights of the leaf edges must be -6 — otherwise, it would be $-4, -2, 0, 2, 4$ or 6, any of which would conflict with the label of $v_2, \ldots, v_6$ or v_7. Note that there exists a feasible $\{-1, 1\}$-edge-weighting of the 6-excluding gadget such that the weights of the leaf edges are all -1, furthermore, the opposite of this weighting is also feasible and all the leaf edges have weight 1.

Now, we describe the construction of the graph which has a proper $\{-1, 1\}$-edge-weighting if and only if the instance of NAE-3SAT3 is solvable. Let $x_1, \ldots, x_n$ denote the variables and let $C_1, \ldots, C_k$ denote the clauses. For each variable x_i, let us introduce a copy of the 6-equal gadget, and for each clause C_j, a copy of the

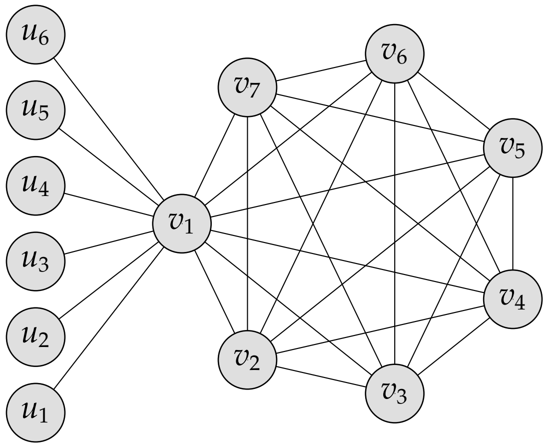

Figure 3: 6-equal gadget

$\{-3,3\}$-excluding gadget. For each variable x_i, unify one of the leaf nodes of the $\{-3,3\}$-excluding gadget of x_i and one of the leaf nodes of all the 6-equal gadgets associated with the clauses including x_i or $\neg x_i$. Repeat the same procedure for the other leaf node of the $\{-3,3\}$-excluding gadget of x_i. Finally, if x_i appears negated in C_j, then subdivide with two-two new nodes the two edges between the gadget associated with x_i and the gadget associated with C_j, which are incident with the K_7 graphs. Figure 4 gives an example for this construction.

To complete the proof, we show that this graph has a feasible $\{-1,1\}$-edge-weighting if and only if the NAE-3SAT3 instance is solvable. On the first hand, assume that the NAE-3SAT3 instance has a feasible solution. For each variable x_i, set the weights of all 6 leaf edges of the 6-equal gadget to 1 if x_i is assigned true, and -1 if it is assigned false. The rest of the edges of the 6-equal gadgets can be weighted feasibly, since the leaf nodes of each gadget are either all 1 or all -1. For each subdivided edge, let the weight of the subdivision incident to a leaf node of the $\{-3,3\}$-excluding gadget be the opposite of the weight of the subdivision incident to the 6-equal gadget. Since each clause C_j must contain at least one true and at least one false literal, the weight of the leaf nodes of the $\{-3,3\}$-excluding gadget associated with C_j may only be $-2, 0$ or 2. Therefore, the current edge-weighting can be extended to the rest of the edges of the $\{-3,3\}$-excluding gadgets. Clearly, we can set the weight of the middle subdivision of the subdivided edges such that no conflict arises (since at most one of -1 and 1 is excluded by the label of the leaf node of the $\{-3,3\}$-excluding gadget). This way one obtains a feasible $\{-1,1\}$-edge-weighting, which was to be shown.

On the other hand, let us given a feasible $\{-1,1\}$-edge-weighting of the graph. For each variable x_i, set x_i to true if the leaf edges of the 6-equal gadget associated with it are all 1, otherwise, they are all -1 by the construction of the 6-equal gadget, and let x_i be false. This way a feasible solution to the NAE-3SAT3 problem is obtained. All we need to verify is that each clause C_j contains at least one true and at least one false literal. Consider the two leaf nodes u and v of the $\{-3,3\}$-excluding gadget associated with C_j. By the construction of the gadget, the sum of

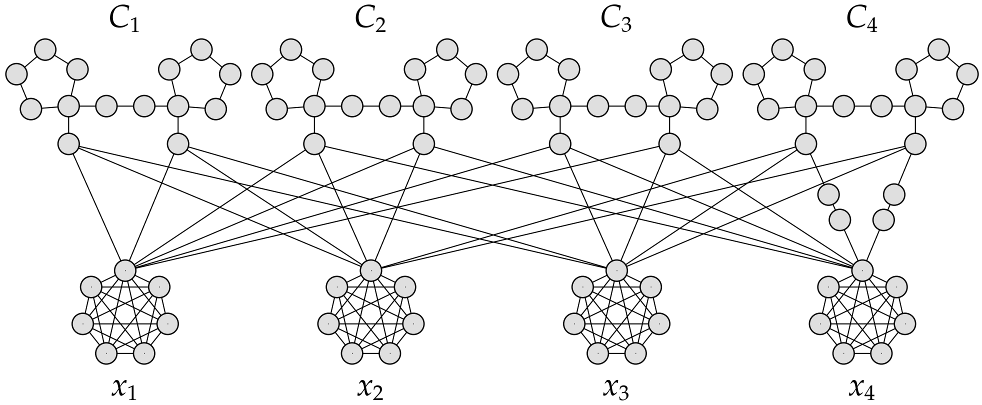

Figure 4: Illustration of the construction in the proof of Theorem 2.6 for the NAE-3SAT3 instance $C_1 \wedge C_2 \wedge C_3 \wedge C_4$, where $C_1 = (x_1 \vee x_2 \vee x_3), C_2 = (x_1 \vee x_2 \vee x_4), C_3 = (x_1 \vee x_3 \vee x_4)$ and $C_4 = (x_2 \vee x_3 \vee \neg x_4)$.

the weights of the edges incident to u is neither -4 nor 4, and the same holds for v. This means that at least one of the incident edges has weight 1 and at least one has weight -1. Consider an incident edge, and let x_i be the variable associated with the corresponding 6-equal gadget. If this edge was not subdivided, then its weight is 1 if and only if x_i was set to true. Whereas, if the edge was subdivided, then its weight is 1 if and only if x_i was set to false. As the edge is subdivided if and only if x_i is negated in C_j, no clause exists with all true or all false literals. This completes the proof of the theorem. □

3 Extending partial edge-weightings

Thomassen, Wu and Zhang [2] proved in 2016 that deciding whether a given bipartite graph has the 1-2 property is possible in polynomial time, while the same problem for arbitrary graphs is NP-complete [1]. Motivated by the former statement, this section investigates whether a partial $\{a, b\}$-edge-weighting can be extended on bipartite graphs, where by a partial $\{a, b\}$-edge-weighting we mean that on a subset of the edges we fix the labels in advance. We will prove that this problem is NP-complete even for bipartite graphs, but polynomial-time solvable on trees for any rational a, b. First of all, let us outline the basic problem, which has not been addressed in the literature yet, as far as we know.

Problem 3.1. Given a graph G with some of its edges already labeled from set $\{a, b\}$, where a, b are two rational numbers. The question is if we can assign weights from $\{a, b\}$ to the uninitialized edges such that the induced coloring is proper.

Theorem 3.2. *Problem 3.1 is NP-complete for bipartite graphs.*

Proof. The polynomial reduction will be given from the NP-complete degree-prescribed subgraph problem [23]. In this problem, the goal is to find a subgraph

$H = (V, E')$ of a given graph $G = (V, E)$ such that the degree of every vertex v in H is from a predefined degree set $F_v \subseteq \{0, \ldots, d_G(v)\}$, that is, $d_H(v) \in F_v$ for every v in V. Let $v_1, \ldots, v_n$ denote the nodes in V.

Without loss of generality, we can assume that G is bipartite. Otherwise, one can replace each edge with a path of length two. We keep the original F_{v_i} sets for $v_i \in V$, and define F_u as $\{0, 2\}$ on each newly created vertex u. This newly created degree-prescribed subgraph problem is defined on a bipartite graph, and it is equivalent to the original one.

Given an instance of the degree-prescribed subgraph problem G, F, we construct a graph G' and a partial $\{a, b\}$-edge-weighting which is extendable to a proper edge-weighting if and only if the degree-prescribed subgraph problem is solvable. Suppose that a and b are relevant. First, observe that the problem is in NP, since if an oracle gives an extended $\{a, b\}$-edge-weighting, then it can be decided in polynomial time whether the weighting is feasible.

We begin the construction of G' with a copy of G, and for every $v_i \in V$, we modify G' as follows. Add iM new leaf nodes connected only to v_i, where $M = 2n$. Let D_i denote the set of these newly added vertices. Initialize the labels of the edges incident to D_i with b. For all $x \in \{0, 1 \ldots, d_G(v_i)\}$, if $x \notin F_{v_i}$, then choose one of the previously-added leaf nodes u, which has not yet been chosen (such a leaf node u always exists, because $|F_{v_i}| \leq M$), and add $(d_G(v_i) + iM - 1)$ new leaf nodes only connected to u. Let the labels of $(iM + x - 1)$ of these edges be b, and let the labels of the remaining $(d_G(v_i) - x)$ edges be a. Clearly, one can construct G' in polynomial time. In the rest of this proof, we show that the partial edge-weighting can be extended in G' if and only if the degree-prescribed subgraph problem can be solved for G and F.

First, assume that the degree-prescribed subgraph problem is solvable for G and F, and let H denote a feasible solution. To extend the partial $\{a, b\}$-edge-weighting of G' to a proper weighting, set the labels of the uninitialized edges of G' — which are the edges of G by the construction of G' — in the following fashion: for each such edge e, if e is contained in H, then let $w(e)$ be b, otherwise a. We show that this $\{a, b\}$-edge-weighting is proper. In G', there cannot be a collision between two vertices which are both in the original graph, because M was defined as $2n$, and hence the induced color of any two nodes of the original graph are different. To show this, it suffices to prove that there cannot be a collision between any two consecutive vertices v_i, v_{i+1}. The largest possible label of v_i is $U_i = (2in + n - 1)b$, since the contribution of the newly added edges is at most $2nib$, while that of the edges of the original graph is at most $(n - 1)b$. On the other hand, the smallest possible label of v_{i+1} is 1) $L_{i+1} = 2(i + 1)nb + a$ if a is non-negative or 2) $L_{i+1} = (2in + n + 1)b$ if a is negative — which can be attained when $|a| = |b|$. In both cases, we can see that L_{i+1} is strictly greater than U_i, that is $z_{v_{i+1}} > z_{v_i}$ in any proper $\{a, b\}$-edge-weighting, since $L_{i+1} \leq z_{v_{i+1}}$ and $U_i \geq z_{v_i}$.

Observe that there is no collision between v_i and the nodes in D_i, because the label of v_i is $bd_H(v_i) + a(d_G(v_i) - d_H(v_i)) + biM$ and the label of the nodes in D_i are $\{by + a(d_G(v_i) - y) + biM : y \in \{0, \ldots, d_G(v_i)\} \setminus F_{v_i}\}$ by the construction. This set does not contain the label of v_i, because $d_H(v_i) \in F_{v_i}$ and $d_{G'}(v_i) = d_{G'}(u_i)$ for all $u_i \in D_i$.

Second, assume that the partial edge-weighting of G' can be extended feasibly. Then let H be the subgraph of G consisting of the edges with weight b. The subgraph H obtained this way is a solution to the degree-prescribed subgraph problem, since as we have seen above, node v_i is not allowed to have exactly $x + iM$ edges with weight b if x is not in F_{v_i}, thus $d_H(v_i)$ must be in F_{v_i} for every i. □

3.1 Extendability on trees

Theorem 3.2 shows that Problem 3.1 is NP-complete on bipartite graphs. In this section, we investigate the same problem on trees, and we give a polynomial-time algorithm which, for a given tree and integer numbers a, b, either completes a given partial $\{a, b\}$-edge-weighting or concludes that no such weighting exists. As a special case, one obtains a new method to decide whether a tree has the 0-1 property, which was first shown to be polynomial-time solvable in [5].

Theorem 3.3. *Problem 3.1 can be solved in polynomial time on trees for any integers a, b.*

Proof. We give a dynamic programming algorithm which either extends the partial a, b weighting into a feasible one or concludes that it cannot be extended. Let us appoint one of the leaf nodes as the root of the tree and let T_v denote the subtree beneath v. For every edge uv, let $L_{uv} \subseteq \{a, b\}$ denote the set of the allowed labels at uv based on the partially initialized edge-weighting. We want to decide if T_v can be extended feasibly such that we fix the weight of uv and the sum of the weights on the edges incident to v, where u is closer to the root than v is. Formally, for every edge uv, we define a subproblem $f(uv)$ as the set of those $(k, l) \in \mathbb{Z} \times \{a, b\}$ pairs for which there exists a weighting of T_v such that $w(uv) = l \in L_{uv}$ and $z(v) = k - l$.

For a given edge uv, let e_i denote the edge between v and its children v_i' for $i = 1, \dots, d(v)-1$. Notice that $(k, l) \in f(uv)$ if and only if the following two conditions hold:

1. For every $i = 1, \dots, d(v) - 1$, there exists a weight $l_i \in L_{e_i}$ and label $k_i \in \mathbb{Z}\setminus\{k\}$ such that $(k_i, l_i) \in f(e_i)$, and
2. $\sum l_i = k - l$,

which gives a way to recursively compute $f(uv)$ in polynomial time, because these conditions can be checked efficiently as follows. There exist unique integers α and β for which

$$\begin{aligned} a \cdot \alpha + b \cdot \beta &= k - l \\ \alpha + \beta &= d(v) - 1 \end{aligned} \tag{2}$$

hold, because $(k, l) \in f(uv)$. That is, out of $e_1, \dots, e_{d(v)-1}$ exactly α have weight a, and β are weighted b. Let $L_{e_i}^k \subseteq L_{e_i}$ denote the possible weights of e_i if $z(v_i') \neq k$, and observe that $(k, l) \in f(uv)$ if and only if $L_{e_i}^k \neq \emptyset$, $|\{i : a \in L_{e_i}^k\}| \geq \alpha$ and $|\{i : b \in L_{e_i}^k\}| \geq \beta$ hold for every $i = 1, \dots, d(v) - 1$. For any e_i and k, $L_{e_i}^k$ can be easily computed by iterating through $f(e_i)$, therefore we obtain an algorithm to compute $f(uv)$ running in $O(n^2)$ steps, provided that the subproblems are computed in increasing order by the depth of the subtrees T_v.

For the base case of the recursion, if subtree T_v consist of a single node for uv, then $f(uv) = \{(l, l) : l \in L_{uv}\}$ by definition.

Once $f(uv)$ is computed for all $uv \in E$, there exists a feasible extension of the partial edge-weighting if and only if there exists $(k, l) \in f(e)$ such that $k \neq l$, where e is the leaf edge incident to the root. This means that the labels of the two endpoints of e are different in the weighting provided by the fact that $(k, l) \in f(e)$. Otherwise, if $k = l$ for all $(k, l) \in f(e)$, then $f(e)$ is either empty or the endpoints of e have the same label in each feasible weighting, which means that no feasible extension of the partial edge-weighting exists. Computing a subproblem $f(uv)$ takes $O(n^2)$ steps, hence the total running time of the algorithm is $O(n^3)$. □

Note that Theorem 3.3 easily extends to the minimum-cost version of the problem in which each weight-edge assignment has an associated cost, and the total cost of the $\{a, b\}$-edge-weighting is to be minimized.

4 Conclusion

This paper presented some progress in terms of the hardness of finding $\{a, b\}$-edge-weightings, and proposed the question of the extendability of a partial edge-weighting in bipartite graphs.

As a generalization of the result of Dudek and Wajc [1], we proved that it is NP-complete to decide whether a graph has a proper $\{a, b\}$-edge-weighting.

If we restrict ourselves to bipartite graphs, then Thomassen, Wu and Zhang [2] proved that it can be decided in polynomial-time if a given bipartite graph has the 1-2 property. More precisely, a bipartite graph has the 1-2 property if and only if it is not a so-called odd multi-cactus. Their approach also extends to $\{a, b\}$-edge-weightings provided that $a < b$, a is odd and b is even. Since then, significant progress has been made regarding the missing cases. Lyngise showed that exactly the odd multi-cacti have no proper edge-weightings for 2-connected bipartite graphs when a is odd and $b = a+2$ [4], and also for bridgeless bipartite graphs when $a = 0$ and $b = 1$ [5]. The general case, however, remains open. As a generalization of the $\{a, b\}$-edge-weighting problem on bipartite graphs, we asked whether a partial $\{a, b\}$-edge-weighting of a bipartite graph can be extended. This problem was shown to be NP-complete, and a polynomial-time algorithm was given for trees. As a special case, the latter result implies an alternative polynomial-time algorithm to decide whether a tree has the 0-1 property, which was first solved in [5].

5 Acknowledgement

The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021.

References

[1] Andrzej Dudek and David Wajc. On the complexity of vertex-coloring edge-weightings. *Discrete Mathematics and Theoretical Computer Science*, 13(3):45–50, 2011.

[2] Carsten Thomassen, Yezhou Wu, and Cun-Quan Zhang. The 3-flow conjecture, factors modulo k, and the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 121:308–325, 2016.

[3] Michał Karoński, Tomasz Łuczak, and Andrew Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory Series B*, 91(1):151–157, 2004.

[4] Julien Bensmail, Fionn Mc Inerney, and Kasper Lyngsie. On {a, b}-edge-weightings of bipartite graphs with odd a, b. *Discussiones Mathematicae Graph Theory*, 2019.

[5] Kasper Szabo Lyngsie. On neighbour sum-distinguishing $\{0, 1\}$-edge-weightings of bipartite graphs. *Discrete Mathematics & Theoretical Computer Science*, 20, 2018.

[6] Gary Chartrand, Michael S Jacobson, Jeno Lehel, Ortrud R Oellermann, Sergio Ruiz, and Farrokh Saba. Irregular networks. *Congr. Numer*, 64(197-210):250th, 1988.

[7] Ralph Faudree and Jeno Lehel. Bound on the irregularity strength of regular graphs. In *Colloq Math Soc Janos Bolyai*, volume 52, pages 247–256, 1987.

[8] Martin Aigner and Eberhard Triesch. Irregular assignments of trees and forests. *SIAM Journal on Discrete Mathematics*, 3(4):439–449, 1990.

[9] Till Nierhoff. A tight bound on the irregularity strength of graphs. *SIAM Journal on Discrete Mathematics*, 13(3):313–323, 2000.

[10] Ben Seamone. The 1-2-3 conjecture and related problems: a survey. *arXiv preprint arXiv:1211.5122*, 2012.

[11] Yinghua Duan, Hongliang Lu, and Qinglin Yu. L-factors and adjacent vertex-distinguishing edge-weighting. *East Asian Journal on Applied Mathematics*, 2(2):83–93, 2012.

[12] Hongliang Lu, Xu Yang, and Qinglin Yu. On vertex-coloring edge-weighting of graphs. *Frontiers of Mathematics in China*, 4(2):325–334, 2009.

[13] Louigi Addario-Berry, Ketan Dalal, Colin McDiarmid, Bruce A Reed, and Andrew Thomason. Vertex-colouring edge-weightings. *Combinatorica*, 27(1):1–12, 2007.

[14] Louigi Addario-Berry, Ketan Dalal, and Bruce A Reed. Degree constrained subgraphs. *Electron. Notes Discret. Math.*, 19:257–263, 2005.

[15] Tao Wang and Qinglin Yu. On vertex-coloring 13-edge-weighting. *Frontiers of Mathematics in China*, 3(4):581–587, 2008.

[16] Maciej Kalkowski, Michał Karoński, and Florian Pfender. Vertex-coloring edge-weightings: towards the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 100(3):347–349, 2010.

[17] Liang Zhong. The 1-2-3-conjecture holds for dense graphs. *Journal of Graph Theory*, 90(4):561–564, 2019.

[18] Jakub Przybyło. The 1–2–3 conjecture almost holds for regular graphs. *Journal of Combinatorial Theory, Series B*, 147:183–200, 2021.

[19] Arash Ahadi, Ali Dehghan, Mohammad-Reza Kazemi, and Esmaeil Mollaahmadi. Computation of lucky number of planar graphs is NP-hard. *Information processing letters*, 112(4):109–112, 2012.

[20] Ali Dehghan, Mohammad-Reza Sadeghi, and Arash Ahadi. The complexity of the sigma chromatic number of cubic graphs. *arXiv preprint arXiv:1403.6288*, 2014.

[21] Ali Dehghan, Mohammad-Reza Sadeghi, and Arash Ahadi. Algorithmic complexity of proper labeling problems. *Theoretical Computer Science*, 495:25–36, 2013.

[22] Ali Dehghan, Mohammad-Reza Sadeghi, and Arash Ahadi. On the complexity of deciding whether the regular number is at most two. *CoRR*, abs/1403.1182, 2014.

[23] László Lovász. The factorization of graphs. ii. *Acta Mathematica Hungarica*, 23(1-2):223–246, 1972.