



HAL
open science

An oriented version of the 1-2-3 Conjecture

Olivier Baudon, Julien Bensmail, Eric Sopena

► **To cite this version:**

Olivier Baudon, Julien Bensmail, Eric Sopena. An oriented version of the 1-2-3 Conjecture. 2013.
hal-00849065v2

HAL Id: hal-00849065

<https://hal.science/hal-00849065v2>

Preprint submitted on 20 Aug 2013 (v2), last revised 7 Oct 2013 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An oriented version of the 1-2-3 Conjecture

Olivier Baudon, Julien Bensmail, Éric Sopena

Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France
CNRS, LaBRI, UMR 5800, F-33400 Talence, France

August 12, 2013

Abstract

The well-known 1-2-3 Conjecture addressed by Karoński, Łuczak and Thomason asks whether the edges of every undirected graph G with no isolated edge can be assigned weights from $\{1, 2, 3\}$ so that the sum of incident weights at each vertex yields a proper vertex-colouring of G . In this work, we consider a similar problem for oriented graphs. We show that the arcs of every oriented graph \vec{G} can be assigned weights from $\{1, 2, 3\}$ so that every two adjacent vertices of \vec{G} receive distinct sums of outgoing (or ingoing) weights. This result is tight in the sense that some oriented graphs do not admit such an assignment using the weights from $\{1, 2\}$ only. We finally prove that deciding whether two weights are sufficient for a given oriented graph is an NP-complete problem. These results also hold for a product version of the problem.

Keywords: oriented graph, neighbour-sum-distinguishing arc-weighting, complexity, 1-2-3 Conjecture

1 Introduction

Let G be an undirected graph with vertex and edge sets $V(G)$ and $E(G)$, respectively. For every vertex v of G , we denote by $N(v)$ the set of vertices neighbouring v . A k -edge-weighting w of G is an assignment $w : E(G) \rightarrow \{1, 2, \dots, k\}$. From w , one naturally deduces a vertex-colouring ϕ_w of G , where $\phi_w(v) = \sum_{u \in N(v)} w(vu)$ for every vertex v . In other words, the vertex v receives the sum of its incident weights by w as its “colour”. If ϕ_w is proper, i.e. we have $\phi_w(u) \neq \phi_w(v)$ for every two adjacent vertices u and v of G , then we say that w is *neighbour-sum-distinguishing* (*nsd* for short).

The study of neighbour-sum-distinguishing edge-weighting of graphs was initiated in 2004, with Karoński, Łuczak and Thomason posing the following conjecture.

1-2-3 Conjecture [8]. *Every graph with no isolated edge admits a nsd-3-edge-weighting.*

Despite many efforts to tackle it, the 1-2-3 Conjecture is still an open question. The best result towards the 1-2-3 Conjecture at the moment is due to Kalkowski, Karoński and Pfender, who proved that every graph with no isolated edge admits a nsd-5-edge-weighting [7].

Many edge-weighting problems inspired by the 1-2-3 Conjecture have been introduced in the literature. As examples, let us mention the notions of *detectable colouring* [1] or *locally irregular edge-colouring* [3] of graphs. We refer the interested reader to [11], where numerous more variants of the original problem are surveyed. Most of these works are devoted to undirected graphs, but one could wonder about an oriented version of the 1-2-3 Conjecture.

We first introduce some terminology related to oriented graphs. Let \vec{G} be an oriented graph, i.e. a loopless directed graph whose every two vertices are joined by at most one arc in either direction, with vertex and arc sets $V(\vec{G})$ and $A(\vec{G})$, respectively. Given a vertex v of \vec{G} , we denote by $N^-(v)$ (resp. $N^+(v)$) the set $\{u \in V(\vec{G}) : \vec{uv} \in A(\vec{G})\}$ (resp. $\{u \in V(\vec{G}) : \vec{vu} \in A(\vec{G})\}$). The *indegree* (resp. *outdegree*) of v , denoted $d^-(v)$ (resp. $d^+(v)$), is $|N^-(v)|$ (resp. $|N^+(v)|$).

To our knowledge, the only link between the 1-2-3 Conjecture and oriented graphs is the following problem. Let w be a k -arc-weighting of \vec{G} , and let $q_w^-(v)$ and $q_w^+(v)$ be $\sum_{u \in N^-(v)} w(\vec{uv})$ and $\sum_{u \in N^+(v)} w(\vec{vu})$, respectively, for every vertex v . The functions q_w^- and q_w^+ naturally yield a vertex-colouring q_w of \vec{G} , where $q_w(v) = q_w^+(v) - q_w^-(v)$ for every vertex v of \vec{G} . It was proved in [4] that every oriented graph admits a 2-arc-weighting w which yields a *proper* vertex-colouring q_w . A list version of the same result was also proved independently in [2] and [9] using different methods.

We here investigate another problem. As for the undirected case, a k -arc-weighting w of \vec{G} yields a vertex-colouring ϕ_w of \vec{G} where $\phi_w(v) = \sum_{u \in N(v)^+} w(\vec{vu})$. This time, the “colour” of v by ϕ_w , sometimes called its *weighted outdegree* (with respect to w), is the sum of its outgoing weights¹. Again, if ϕ_w has the property to be proper, then we say that w is *neighbour-sum-distinguishing* (*nsd* for short).

A quick investigation on small oriented graphs suggests that all oriented graphs should admit a nsd-3-arc-weighting. Besides, there exist oriented graphs, such as the circuit on 3 vertices, which do not admit a nsd-2-arc-weighting. We hence investigate the following question.

¹Note that the problem where a vertex receives the sum of its ingoing weights as its sum is similar to the one investigated here.

Question 1. *Does every oriented graph admit a nsd-3-arc-weighting?*

Although Question 1 and the 1-2-3 Conjecture are quite similar in essence, the two underlying problems do not seem to share any systematic relationship. The fact that weighting some arc \vec{uv} only affects the weighted outdegree of u makes Question 1 easier to handle. We hence answer this question in the affirmative in Section 2. We then turn our concern on conditions for some specific classes of oriented graphs to admit a nsd-2-arc-weighting in Section 3. We next prove, in Section 4, that an “easy” characterization of oriented graphs which admit a nsd-2-arc-weighting cannot exist unless $P=NP$. For this purpose, we show that the problem of deciding whether an oriented graph admits a nsd-2-arc-weighting is NP-complete. Concluding remarks can be found in Section 5. In particular, we discuss a product version of our weighting problem.

2 All oriented graphs admit a nsd-3-arc-weighting

Our main result states that every oriented graph admits a nsd-3-arc-weighting. This relies on the fact that every oriented graph contains a “convenient” vertex, i.e. a vertex which admits a large number of potential weighted outdegrees regarding its number of neighbours. The existence of such a vertex allows the use of an inductive proof scheme. Our proof also yields a polynomial-time algorithm for finding a nsd-3-arc-weighting of every oriented graph.

Theorem 1. *Every oriented graph \vec{G} admits a nsd-3-arc-weighting.*

Proof. Note that it is enough to consider connected oriented graphs only. The claim is proved by induction on the size of \vec{G} , i.e. its number of arcs. The claim is clearly true when \vec{G} has size 1. Suppose now that the claim is true for every oriented graph with at most $m - 1$ arcs, and assume \vec{G} has size m .

Note that \vec{G} necessarily has a vertex v such that $d^+(v) > 0$ and $d^+(v) \geq d^-(v)$ since otherwise we would have $\sum_{v \in V(\vec{G})} d^-(v) \neq \sum_{v \in V(\vec{G})} d^+(v)$. A nsd-3-arc-weighting of \vec{G} is then obtained as follows. Start by removing the arcs outgoing from v . According to the induction hypothesis, the remaining oriented graph admits a nsd-3-arc-weighting w (if this oriented graph is not connected, then w is an arc-weighting of all the resulting components). Now put back the arcs outgoing from v to \vec{G} , and extend w to these arcs in such a way that the weighted outdegree of v is different from the weighted outdegrees of the $d^-(v) + d^+(v)$ vertices neighbouring v . This is possible since there are $2d^+(v) + 1$ potential weighted outdegrees for v , namely those

from $\{d^+(v), d^+(v) + 1, \dots, 3d^+(v)\}$, while the number of forbidden weighted outdegrees is at most $d^-(v) + d^+(v) < 2d^+(v) + 1$ by the assumption on $d^-(v)$ and $d^+(v)$. Because assigning a weight to the arcs outgoing from v does not affect the weighted outdegree by w of any vertex neighbouring v , the extension of w to \vec{G} remains neighbour-sum-distinguishing. \square

3 Some families of oriented graphs which admit a nsd-2-arc-weighting

By Theorem 1, we know that every oriented graph admits a nsd-3-arc-weighting. Throughout this section, we focus on some oriented graphs which admits a nsd-2-arc-weighting. In particular, we exhibit conditions regarding some common families of oriented graphs.

3.1 Acyclic oriented graphs

An oriented graph is *acyclic* if it does not admit a circuit as an induced subgraph. We show that every such oriented graph admits a nsd-2-arc-weighting.

Theorem 2. *Every acyclic oriented graph admits a nsd-2-arc-weighting.*

Proof. We prove the claim by induction on the order, i.e. the number of vertices, of acyclic oriented graphs. Once again, it is enough to consider connected oriented graphs. As a starting point, note that an oriented graph with only 1 vertex admits a nsd-2-arc-weighting. Suppose now that the claim is true for every acyclic oriented graph with order at most $n - 1$ for some n , and let \vec{G} be an acyclic oriented graph on n vertices.

It is well-known that every acyclic oriented graph has a vertex with indegree 0. Let v be such a vertex of \vec{G} , and consider the graph \vec{G}' obtained by removing v from \vec{G} . Clearly \vec{G}' is acyclic and admits a nsd-2-arc-weighting w according to the induction hypothesis. We now extend w to \vec{G} , i.e. we weight the arcs outgoing from v in such a way that w remains neighbour-sum-distinguishing. Note that there are $d^+(v) + 1$ possible weighted outdegrees for v , namely those from $\{d^+(v), d^+(v) + 1, \dots, 2d^+(v)\}$, while there are at most $d^+(v)$ forbidden weighted outdegrees for v , namely the weighted outdegrees by w of the vertices in $N^+(v)$. Since weighting the arcs outgoing from v does not alter the weighted outdegree of any vertex neighbouring v , we can “choose” an available weighted outdegree for v and weight the arcs outgoing from v consequently. This completes the proof. \square

3.2 Oriented graphs whose underlying graph is k -colourable

Given an undirected graph G , a k -vertex-colouring of G is a partition of $V(G)$ into k parts V_1, \dots, V_k such that each V_i is an independent set. The least number of parts of a vertex-colouring of G is referred to as the *chromatic number* of G , denoted $\chi(G)$. Assuming \vec{G} is any orientation of G , i.e. \vec{G} is obtained by orienting every edge of G in either direction, it should be understood that $\text{und}(\vec{G})$ refers to the *underlying* undirected graph of \vec{G} , that is G .

As pointed out in some references of the literature (see e.g. [1] or [8]), first partitioning a graph into several independent sets before weighting its edges can be a good method for finding a specific edge-weighting. This is also the case regarding neighbour-sum-distinguishing arc-weighting, as shown in the following result.

Theorem 3. *Every oriented graph \vec{G} admits a $\text{nsd-}\chi(\text{und}(\vec{G}))$ -arc-weighting.*

Proof. Let $k = \chi(\text{und}(\vec{G}))$, and V_0, \dots, V_{k-1} be a k -vertex-colouring of $\text{und}(\vec{G})$. Process the vertices of \vec{G} in arbitrary order. If the vertex v belongs to the part V_i , then weight the arcs outgoing from v with weights from $\{1, 2, \dots, k\}$ in such a way that the weighted outdegree of v is congruent to i modulo k . This is possible unless $d^+(v) = 0$ since, in such a situation, the only possible weighted outdegree for v is 0. Once the process is achieved, two adjacent vertices u and v cannot have the same weighted outdegrees since otherwise either u and v would both belong to some V_i , which is impossible since V_i is an independent set, or we would have $d^+(u) = d^+(v) = 0$ with u and v being adjacent, which is impossible. \square

As a corollary of Theorem 3, we get in particular the following result.

Corollary 4. *Every oriented graph \vec{G} whose underlying graph is bipartite admits a nsd-2 -arc-weighting.*

3.3 Tournaments

Our strategy for weighting the arcs of any tournament \vec{T} is based on the following lemma, which could be also deduced thanks to a result of Landau (see [10]).

Lemma 5. *For every $k \in \{1, 2, \dots, |V(\vec{T})|\}$, let $n_k \geq 0$ denote the number of vertices with outdegree at most k of any tournament \vec{T} . Then $n_k \leq 2k + 1$.*

Proof. Let k be fixed, with $1 \leq k \leq |V(\vec{T})|$. Denote by $X \subseteq V(\vec{T})$ the set of the n_k vertices of \vec{T} whose outdegree is at most k , and by s the sum

of outdegrees of the vertices in X . Naturally, we have $s \leq n_k k$. We also have $s \geq \frac{n_k(n_k-1)}{2}$ since X induces a tournament, and there may be arcs of \vec{T} whose tails lie in X , and whose heads do not lie in X . We hence get $\frac{n_k(n_k-1)}{2} \leq n_k k$, which implies that $n_k \leq 2k + 1$. \square

We now give a characterization of tournaments which admit a nsd-2-arc-weighting.

Theorem 6. *For every $k \in \{1, 2, \dots, |V(\vec{T})|\}$, let $n_k \geq 0$ denote the number of vertices with outdegree at most k of any tournament \vec{T} . Then \vec{T} admits a nsd-2-arc-weighting if and only if $n_k \leq k + 1$ for every $k \in \{1, 2, \dots, |V(\vec{T})|\}$.*

Proof. The proof is based on the following simple weighting scheme for \vec{T} . Process the vertices of \vec{T} in increasing order of their outdegrees. For each vertex v , weight the arcs outgoing from v in such a way that the weighted outdegree of v gets the smallest possible value which does not appear among the weighted outdegrees of the neighbours of v considered in earlier steps of the process. Assume $k = d^+(v)$. Since the weighted outdegree of v can take any value from $\{k, k + 1, \dots, 2k\}$ and at most $n_k - 1 < k + 1$ vertices have been considered in earlier steps of the process, there is necessarily one non-conflicting value which can be chosen as the weighted outdegree of v . We then just have to weight the arcs outgoing from v consequently.

The “only if” condition of the statement follows from the fact that if we have $n_k \in \{k + 2, k + 3, \dots, 2k + 1\}$ for some value of k (the upper bound on n_k follows from Lemma 5), then we need to increase the range of possible values as the weighted outdegree of v , i.e. to allow the use of a third weight. \square

3.4 Cartesian products of oriented graphs

Let \vec{G} and \vec{H} be two oriented graphs. The *Cartesian product* of \vec{G} and \vec{H} , denoted $\vec{G} \square \vec{H}$, is the oriented graph with vertex set $V(\vec{G}) \times V(\vec{H})$, and whose two vertices (u_1, v_1) and (u_2, v_2) are joined by an arc from (u_1, v_1) towards (u_2, v_2) if and only if $u_1 = u_2$ and $\overrightarrow{v_1 v_2} \in E(\vec{H})$, or $\overrightarrow{u_1 u_2} \in E(\vec{G})$ and $v_1 = v_2$.

The Cartesian product of graphs is a classical graph operation which has been studied a lot since its introduction [6]. The reason for focusing on the Cartesian product of oriented graphs is that if \vec{G} and \vec{H} both admit a nsd- k -arc-weighting for some value of k , one could expect $\vec{G} \square \vec{H}$ to require more than k weights to obtain a neighbour-sum-distinguishing arc-weighting. In the next result, we show that the existence of a nsd- k -arc-weighting of \vec{G} and \vec{H} implies the existence of a nsd- k -arc-weighting of $\vec{G} \square \vec{H}$.

Theorem 7. Assume \vec{G} and \vec{H} admit a nsd- k - and nsd- ℓ -arc-weighting, respectively. Then $\vec{G} \square \vec{H}$ admits a nsd- $\max\{k, \ell\}$ -arc-weighting.

Proof. Let $w_{\vec{G}}$ and $w_{\vec{H}}$ be nsd- k - and nsd- ℓ -arc-weighting of \vec{G} and \vec{H} , respectively. Let w be a $\max\{k, \ell\}$ -arc-weighting of $\vec{G} \square \vec{H}$ defined as follows:

$$w(\overrightarrow{(u_1, v_1)(u_2, v_2)}) = \begin{cases} w_{\vec{H}}(\overrightarrow{v_1 v_2}) & \text{if } u_1 = u_2, \\ w_{\vec{G}}(\overrightarrow{u_1 u_2}) & \text{otherwise.} \end{cases}$$

Assume $\overrightarrow{(u_1, v_1)(u_2, v_2)}$ is an arc of $\vec{G} \square \vec{H}$. Then we have $\phi_w(u_1, v_1) = \phi_{w_{\vec{G}}}(u_1) + \phi_{w_{\vec{H}}}(v_1)$ and $\phi_w(u_2, v_2) = \phi_{w_{\vec{G}}}(u_2) + \phi_{w_{\vec{H}}}(v_2)$. Since (u_1, v_1) and (u_2, v_2) are adjacent, we have either $u_1 = v_1$ or $u_2 = v_2$ by construction. Assume $u_1 = v_1$ without loss of generality. Then $\phi_{w_{\vec{G}}}(u_1) = \phi_{w_{\vec{G}}}(v_1)$. Now, because $w_{\vec{H}}$ is neighbour-sum-distinguishing, we have $\phi_{w_{\vec{H}}}(u_2) \neq \phi_{w_{\vec{H}}}(v_2)$. It then follows that $\phi_w(u_1, v_1) \neq \phi_w(u_2, v_2)$. \square

An immediate corollary of Theorem 7 is the following result.

Corollary 8. Assume \vec{G} and \vec{H} both admit a nsd-2-arc-weighting. Then $\vec{G} \square \vec{H}$ admits a nsd-2-arc-weighting.

4 Algorithmic complexity

In this section, we focus on the complexity of the following decision problem.

NEIGHBOUR-SUM-DISTINGUISHING k -ARC-WEIGHTING - k -NSDAW

Instance: An oriented graph \vec{G} .

Question: Does \vec{G} admits a nsd- k -arc-weighting?

An oriented graph \vec{G} admits a nsd-1-arc-weighting if and only if every two adjacent vertices of \vec{G} have distinct outdegrees. Since this property can be checked in polynomial time, the problem 1-NSDAW is in P. Besides, every problem k -NSDAW with $k \geq 3$ is also in P since the answer to every of its instances is *yes*, according to Theorem 1.

We deal with the complexity of the remaining problem, i.e. 2-NSDAW. We show this problem to be NP-complete in Theorem 11 below, by reduction from 3-SAT. For this purpose, we first introduce several gadgets to “force” the propagation of a nsd-2-arc-weighting along an oriented graph.

We first introduce two kinds of *forbidding* gadgets. A forbidding gadget \vec{F} is composed of one *root vertex* with outdegree 0 adjacent to *forcing vertices*. The weighting property of \vec{F} is that its forcing vertices have the same weighted outdegree, say x_1, x_2, \dots, x_k , by every nsd-2-arc-weighting of

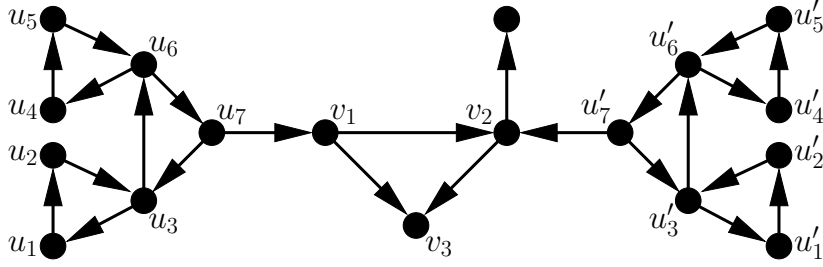


Figure 1: The forbidding gadget $\overrightarrow{F}_{3,4}$.

\overrightarrow{F} . Because of this property, note that, after having identified the root of \overrightarrow{F} with a vertex v of some graph \overrightarrow{G} , the vertex v cannot have weighted outdegree x_1, x_2, \dots, x_k by any nsd-2-arc-weighting of \overrightarrow{G} because of the forcing vertices of \overrightarrow{F} neighbouring v .

First, we define a $(2k-1, 2k)$ -forbidding gadget, denoted $\overrightarrow{F}_{2k-1, 2k}$, for every integer $k \geq 2$. These gadgets are defined inductively. First, the gadget $\overrightarrow{F}_{3,4}$ is the one depicted in Figure 1. We call v_3 the root of $\overrightarrow{F}_{3,4}$, while v_1 and v_2 are its forcing vertices. Now, for any value of $k \geq 3$ such that the oriented graphs $\overrightarrow{F}_{2k'-1, 2k'}$ have been defined for every $k' < k$, the oriented graph $\overrightarrow{F} = \overrightarrow{F}_{2k-1, 2k}$ is constructed as follows. Let v_1^k, v_2^k and v_3^k be three distinct vertices joined by $\overrightarrow{v_1^k v_3^k}, \overrightarrow{v_2^k v_3^k}$ and $\overrightarrow{v_1^k v_2^k}$. Now, for every $k' \in \{2, 3, \dots, k-1\}$, identify v_1^k and the root of a copy of $\overrightarrow{F}_{2k'-1, 2k'}$. Repeat the same procedure but with v_2^k instead of v_1^k and new copies of the forbidding gadgets. Finally add an arc from v_1^k towards $k-2$ new vertices with outdegree 0, and similarly from v_2^k towards $k-1$ new vertices with outdegree 0. The root of \overrightarrow{F} is v_3^k , while its forcing vertices are v_1^k and v_2^k .

Lemma 9. *Let $k \geq 2$ be fixed. In every nsd-2-arc-weighting of $\overrightarrow{F}_{2k-1, 2k}$, one of the forcing vertices has weighted outdegree $2k-1$, while the other forcing vertex has weighted outdegree $2k$.*

Proof. We prove the claim by induction on k . At each step, let w be a nsd-2-arc-weighting of the considered forbidding gadget. Start with $\overrightarrow{F} = \overrightarrow{F}_{3,4}$. Since u_1 and u_2 are adjacent and both have outdegree 1, we have $\{\phi_w(u_1), \phi_w(u_2)\} = \{1, 2\}$. By the same argument, we have $\{\phi_w(u_4), \phi_w(u_5)\} = \{1, 2\}$. Since u_3 and u_6 are adjacent, are both adjacent to vertices with weighted outdegree 2, and have outdegree 2, we necessarily have $\{\phi_w(u_3), \phi_w(u_6)\} = \{3, 4\}$. Because u_7 is adjacent to u_3 and u_6 and has outdegree 2, we necessarily get $\phi_w(u_7) = 2$. Repeating the same arguments to the oriented subgraph of \overrightarrow{F} induced by the u'_i 's, we also obtain $\phi_w(u'_7) = 2$. Finally, since v_1 and v_2 are adjacent, are both adjacent to a vertex with weighted

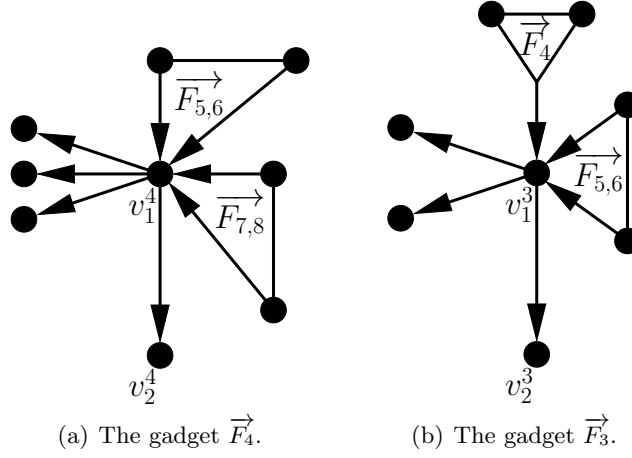


Figure 2: Two examples of forbidding gadgets. A triangle represents a forbidding gadget.

outdegree 2, and have outdegree 2, we have $\{\phi_w(v_1), \phi_w(v_2)\} = \{3, 4\}$ as claimed.

Assume the claim is true for every k up to $i - 1$, and consider $k = i$. Let $\vec{F} = \vec{F}_{2k-1, 2k}$. Because v_1^k and v_2^k have outdegree k by construction, their weighted outdegree by w can only take value from $\{k, k + 1, \dots, 2k\}$. However, since these two vertices are both identified with the roots of forbidding gadgets $\vec{F}_{3,4}, \vec{F}_{5,6}, \dots, \vec{F}_{2k-3, 2k-2}$, their weighted outdegree cannot take value from $\{3, 4, \dots, 2k - 3, 2k - 2\}$ according to the induction hypothesis. Therefore, we have $\{\phi_w(v_1^k), \phi_w(v_2^k)\} = \{2k - 1, 2k\}$ since v_1^k and v_2^k are adjacent. \square

We now define a k -forbidding gadget, denoted \vec{F}_k , for every integer $k \geq 3$. The oriented graph \vec{F}_k originally consists in an arc $\vec{v}_1^k v_2^k$. We call v_2^k and v_1^k the root and the forcing vertex of \vec{F}_k , respectively. Next add an arc from v_1^k towards $k - 1$ new vertices with outdegree 0. The end of the construction depends on the parity of k . If k is even, then identify v_1^k and the root of each of the forbidding gadgets $\vec{F}_{k+1, k+2}, \vec{F}_{k+3, k+4}, \dots, \vec{F}_{2k-1, 2k}$. Otherwise, i.e. if k is odd, then identify v_1^k and the roots of \vec{F}_{k+1} , and $\vec{F}_{k+2, k+3}, \vec{F}_{k+4, k+5}, \dots, \vec{F}_{2k-1, 2k}$. The gadgets \vec{F}_3 and \vec{F}_4 are depicted in Figure 2.

Lemma 10. *Let $k \geq 3$ be fixed. In every nsd-2-arc-weighting of \vec{F}_k , the forcing vertex has weighted outdegree k .*

Proof. Let w be a nsd-2-arc-weighting of $\vec{F} = \vec{F}_k$. Assume k is even. Since v_1^k has outdegree k , its weighted outdegree by w can only take value

from $\{k, k+1, \dots, 2k\}$. But, because v_1^k is the root of forbidding gadgets $\overrightarrow{F_{k+1, k+2}}, \overrightarrow{F_{k+3, k+4}}, \dots, \overrightarrow{F_{2k-1, 2k}}$, the vertex v_1^k is adjacent to vertices with weighted outdegrees $k+1, k+2, \dots, 2k$ according to Lemma 9. Hence, the only remaining weighted outdegree for v_1^k by w is k . The claim follows similarly when k is odd, the value $k+1$ being forbidden as the weighted outdegree of v_1^k because it was identified with the root of a forbidding gadget $\overrightarrow{F_{k+1}}$ with $k+1$ being even. \square

Thanks to the two kinds of forbidding gadgets introduced above, we can “force” a vertex of some oriented graph to have a specific weighted outdegree by any nsd-2-arc-weighting. Let v be a vertex of some oriented graph \overrightarrow{G} , and $D \subseteq \{d^+(v), d^+(v)+1, \dots, 2d^+(v)\}$ be a set of “allowed” weighted outdegrees for v . We say that v is a D -vertex if its weighted outdegree by every nsd-2-arc-weighting of \overrightarrow{G} can only take value among D .

By “turning v into a D -vertex”, we refer to the following operations. Let $k \geq 1$ be fixed, and assume $d^+(v) \leq k$. First add arcs from v towards $k - d^+(v)$ new vertices with outdegree 0 so that v has outdegree k . Then, identify v and the respective root of each of the forbidding gadgets $\overrightarrow{F_i}$ with $i \in \{d^+(v), d^+(v)+1, \dots, 2d^+(v)\} - D$. Because of the forcing vertices neighbouring v , it is easy to check that v is a D -vertex.

We are now ready to introduce our hardness reduction.

Theorem 11. *The problem 2-NSDAW is NP-complete.*

Proof. Given a 2-arc-weighting w of \overrightarrow{G} , one can first compute the vertex-colouring ϕ_w of \overrightarrow{G} from w , and then check whether it is proper. Since this procedure can be achieved in polynomial time, 2-NSDAW is in NP.

We now prove that 2-NSDAW is NP-hard by reduction from the following classical NP-complete problem [5].

3-SAT

Instance: A 3CNF formula F over clauses C_1, \dots, C_m and variables x_1, \dots, x_n .

Question: Does F admit a satisfying truth assignment?

Note that we can assume that every possible literal appears in F . Indeed, if ℓ_i does not appear in any clause of F , then the 3CNF formula $F \wedge (\ell_i \vee \ell_i \vee \overline{\ell_i})$ is satisfiable if and only if F is satisfiable. By repeating this procedure for all literals which do not appear in F , we obtain a formula equivalent to F but involving all possible literals over its variables. This procedure is achieved in polynomial time.

We introduce some more terminology regarding an instance of 3-SAT. The $2n$ literals of F over its n variables are denoted by $\ell_1, \ell_2, \dots, \ell_{2n}$, the ordering being arbitrary. By $n_i \geq 1$, we refer to the number of distinct clauses

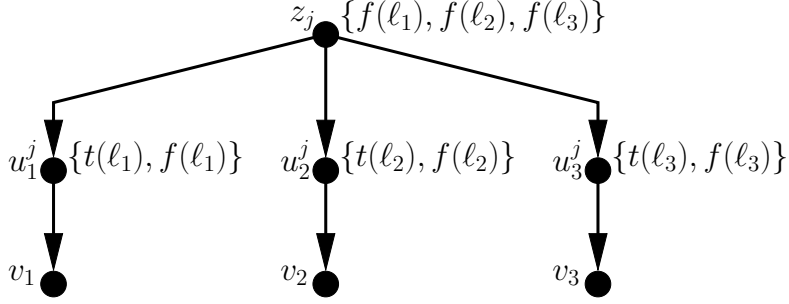


Figure 3: Partial resulting clause gadget for a clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$. The integer sets represent the allowed weighted outdegrees at each vertex by a nsd-2-arc-weighting of \overrightarrow{G}_F .

of F that contain the literal ℓ_i for every $i \in \{1, 2, \dots, 2n\}$. By $c_i \in \{1, 2, 3\}$, we denote the number of distinct literals which appear in the clause C_i for every $i \in \{1, 2, \dots, m\}$. In the case where $c_j = 1$, i.e. C_j is of the form $(\ell_i \vee \ell_i \vee \ell_i)$, note that ℓ_i is set to true by every satisfying truth assignment of F . In such a situation, we say that ℓ_i is *forced* to true by C_j .

Our hardness reduction is described below. From a 3CNF formula F , we construct an oriented graph \overrightarrow{G}_F such that F is satisfiable if and only if \overrightarrow{G}_F admits a nsd-2-arc-weighting w_F .

Let t and f be two injective mappings from $\{x_1, x_2, \dots, x_n\}$ to $\{2n, 2n + 1, \dots, 3n - 1\}$ and $\{3n, 3n + 1, \dots, 4n\}$, respectively. Assuming $\ell_j = x_i$ and $\ell_{j'} = \bar{\ell}_j$, i.e. ℓ_j and $\ell_{j'}$ are the literals associated with the variable x_i , we set $t(\ell_j) = f(\ell_{j'}) = t(x_i)$ and $f(\ell_j) = t(\ell_{j'}) = f(x_i)$.

First, for every literal ℓ_i of F , add a vertex v_i in \overrightarrow{G}_F . Now consider every clause C_j of F . We associate a *clause gadget* in \overrightarrow{G}_F with C_j , its structure depending on the value of c_j . Denote by $\ell_{j_1}, \ell_{j_2}, \dots, \ell_{j_{c_j}}$ the distinct literals of C_j . Let $\overrightarrow{u_{j_1}^j v_{j_1}}, \overrightarrow{u_{j_2}^j v_{j_2}}, \dots, \overrightarrow{u_{j_{c_j}}^j v_{j_{c_j}}}$ be c_j arcs of \overrightarrow{G}_F , where $u_{j_1}^j, u_{j_2}^j, \dots, u_{j_{c_j}}^j$ are new vertices. If $c_j = 1$, i.e. ℓ_{j_1} is the only literal of C_j , then turn $u_{j_1}^j$ into a $\{t(\ell_{j_1})\}$ -vertex. Otherwise, i.e. $c_j \in \{2, 3\}$, turn each vertex $u_{j_i}^j$ into a $\{t(\ell_{j_i}), f(\ell_{j_i})\}$ -vertex, add a vertex z_j to \overrightarrow{G}_F , add arcs from z_j towards $u_{j_1}^j, u_{j_2}^j, \dots, u_{j_{c_j}}^j$, and turn z_j into a $\{f(\ell_{j_1}), f(\ell_{j_2}), \dots, f(\ell_{j_{c_j}})\}$ -vertex. This construction is depicted in Figure 3.

Claim 1. *Let $C_j = (\ell_{j_1} \vee \ell_{j_2} \vee \ell_{j_3})$ be a clause of F . Then one of $t(\ell_{j_1})$, $t(\ell_{j_2})$ and $t(\ell_{j_3})$ belongs to $\{\phi_{w_F}(u_{j_1}^j), \phi_{w_F}(u_{j_2}^j), \phi_{w_F}(u_{j_3}^j)\}$.*

Proof. The claim is true when $c_j = 1$ since $u_{j_1}^j$ is a $\{t(\ell_{j_1})\}$ -vertex. When $c_j \in \{2, 3\}$, note that we cannot have $\phi_{w_F}(u_{j_1}^j) = f(\ell_{j_1})$, $\phi_{w_F}(u_{j_2}^j) = f(\ell_{j_2})$,

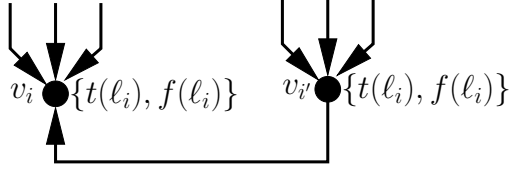


Figure 4: Partial resulting subgraph of \overrightarrow{G}_F for two literals l_i and $l_{i'}$ such that $l_{i'} = \bar{l}_i$. The integer sets represent the allowed weighted outdegrees at each vertex by a nsd-2-arc-weighting of \overrightarrow{G}_F .

..., $\phi_{w_F}(u_{j_{c_j}}^j) = f(\ell_{j_{c_j}})$ since z_j is a $\{f(\ell_{j_1}), f(\ell_{j_2}), \dots, f(\ell_{j_{c_j}})\}$ -vertex. On the contrary, note that if there is an $i \in \{1, 2, \dots, c_j\}$ such that $\phi_{w_F}(u_{j_i}^j) = t(\ell_{j_i})$, then we can weight the arcs outgoing from z_j in such a way that the weighted outdegree of z_j by w_F is $f(\ell_{j_i})$. \square

Let $i \in \{1, 2, \dots, 2n\}$. Note that, so far, the vertex v_i has indegree n_i . Consider $i' \in \{1, 2, \dots, 2n\}$ such that $l_{i'} = \bar{l}_i$. To finish the construction of \overrightarrow{G}_F , add the arc $\overrightarrow{v_{i'}v_i}$, and turn v_i and $v_{i'}$ into $\{t(\ell_i), f(\ell_i)\}$ -vertices. This step of the construction is illustrated in Figure 4.

Claim 2. Let $i \in \{1, 2, \dots, 2n\}$, and i_1, i_2, \dots, i_{n_i} be the indexes of the distinct clauses of F that contain l_i . Then $\phi_{w_F}(u_i^{i_1}) = \phi_{w_F}(u_i^{i_2}) = \dots = \phi_{w_F}(u_i^{i_{n_i}})$.

Proof. Recall that the $u_i^{i_j}$'s can only have weighted outdegree $t(\ell_i)$ or $f(\ell_i)$ by w_F . Now note that if one of the $u_i^{i_j}$'s has weighted outdegree $t(\ell_i)$ by w_F while another such vertex has weighted outdegree $f(\ell_i)$, then w_F cannot be extended to the arcs outgoing from v_i since v_i is a $\{t(\ell_i), f(\ell_i)\}$ -vertex. On the contrary, if all the $u_i^{i_j}$'s neighbouring v_i have the same weighted outdegree, say $t(\ell_i)$, then the arcs outgoing from v_i can be weighted in such a way that $\phi_{w_F}(v_i) = f(\ell_i)$. \square

Claim 3. Let $i, i' \in \{1, 2, \dots, 2n\}$ be two integers such that $l_{i'} = \bar{l}_i$. Then $\phi_{w_F}(v_i) \neq \phi_{w_F}(v_{i'})$.

Proof. The claim follows from the fact that v_i and $v_{i'}$ are adjacent. \square

We claim that F has a satisfying truth assignment if and only if \overrightarrow{G}_F admits its nsd-2-arc-weighting w_F . Assume $C_j = (\ell_{j_1} \vee \ell_{j_2} \vee \ell_{j_3})$ is a clause of F , and that having $\phi_{w_F}(u_{j_i}^j) = t(\ell_{j_i})$ (resp. $f(\ell_{j_i})$) simulates the assignment of ℓ_{j_i} to true (resp. false) in C_j by a truth assignment of F . Then, by Claim 1, every clause gadget of \overrightarrow{G}_F must have a vertex $u_{j_i}^j$ whose weighted outdegree by w_F is $t(\ell_{j_i})$. This simulates the fact that every clause of F must

have one true literal by a satisfying truth assignment of F . Claim 2 depicts the fact that, by a truth assignment of F , every literal of F provides the similar truth value to every clause it appears in. Finally, Claim 3 represents the fact that two opposite literals cannot be assigned the same truth value by a truth assignment of F . With these arguments, we can deduce a satisfying truth assignment of F from w_F , and vice-versa. \square

5 Discussion

We can extend the problem investigated in this work to undirected graphs.

Question 2. *What is the least $k \in \{1, 2, 3\}$ such that every undirected graph admits an orientation which admits a nsd- k -arc-weighting?*

Recall that an oriented graph admits a nsd-1-arc-weighting if and only if every two of its adjacent vertices have distinct outdegrees. According to a result from [4], the answer to Question 2 is 1. We give an alternative proof of this statement using our terminology.

Lemma 12. *Every undirected graph G admits an orientation in which every two adjacent vertices have distinct outdegrees.*

Proof. We prove this result by induction on the order n of G . Since the result is true for $n = 1$, we assume the claim is true for every n up to $i - 1$, and now consider $n = i$. Let v be a vertex whose degree is maximum in G . According to the induction hypothesis, the graph $G' = G - \{v\}$ admits an orientation \vec{G}' in which every two adjacent vertices have distinct outdegrees. Note that in \vec{G}' , the outdegree of every vertex in $N(v)$ is at most $d(v) - 1$ since v has maximum degree. Now start from \vec{G}' , and let \vec{G} be the orientation of G obtained by orienting all edges incident with v from v towards its neighbours. Since the outdegree of v in \vec{G} then is $d(v)$, and the outdegrees of all vertices neighbouring v are not altered, the orientation still satisfies the claim. \square

Corollary 13. *Every undirected graph admits an orientation which admits a nsd-1-arc-weighting.*

As for the undirected case, one can consider a variant of the problem investigated in this work where the weighted outdegree of a vertex is the *product* of its outgoing weights rather than their sum (see e.g. [12]). Formally, from a k -arc-weighting w of some oriented graph \vec{G} one obtains a vertex-colouring ρ_w defined as $\rho_w(v) = \prod_{u \in N^+(v)} w(\vec{vu})$ for every $v \in V(\vec{G})$.

If ρ_w is proper, then we say that w is *neighbour-product-distinguishing* (*npd* for short).

Regarding npd-arc-weightings, note that the range of possible weighted outdegrees for a vertex is wider in the product version than in the sum version when the weights from $\{1, 2, 3\}$ are allowed. Hence, our proof of Theorem 1 is also a proof that every oriented graph admits a npd-3-arc-weighting.

Note that there are $k + 1$ possible weighted outdegrees for a vertex with outdegree k by a npd-2-arc-weighting of some oriented graph, namely those from $\{1, 2, 4, \dots, 2^k\}$. Since there are as many possible weighted outdegrees for a vertex by a nsd-2-arc-weighting and a npd-2-arc-weighting, our results from Section 3 also hold regarding npd-2-arc-weightings.

Finally, we could adapt the reduction scheme from Section 4 to prove that it is NP-complete to decide whether a given oriented graph admits a npd-2-arc-weighting. The forbidding gadgets would be obtain, for instance, as follows. Start from the circuit $\overrightarrow{u_1 u_2 u_3 u_1}$ on 3 vertices, and add an arc $\overrightarrow{u_1 u_4}$ where u_4 is a new vertex. This resulting oriented graph $\overrightarrow{F_4}$ is a 4-forbidding gadget since it is easy to check that u_1 would necessarily get weighted outdegree 4 by any npd-2-arc-weighting. The root of $\overrightarrow{F_4}$ is u_4 . Now consider another oriented graph $\overrightarrow{F_{1,2}}$ with vertices v_1, v_2, v_3 and v_4 such that $\overrightarrow{v_1 v_2}, \overrightarrow{v_1 v_3}, \overrightarrow{v_2 v_3}$ and $\overrightarrow{v_2 v_4}$ are arcs, and v_1 and v_2 are each identified with the root of one copy of $\overrightarrow{F_4}$. Clearly, since v_1 and v_2 are adjacent vertices with outdegree 2, and they are both identified with the root of a gadget $\overrightarrow{F_4}$, their weighted outdegree can only be 1 and 2 without loss of generality, and $\overrightarrow{F_{1,2}}$ is thus a (1, 2)-forbidding gadget with root v_3 . Now to obtain a 2^k -forbidding gadget $\overrightarrow{F_{2^k}}$ assuming that a $2^{k'}$ -forbidding gadget has been defined for every $k' < k$ (with the exception that there is a (1, 2)-forbidding gadget rather than a 1-forbidding gadget and a 2-forbidding gadget), start from the arc $\overrightarrow{w_1 w_2}$, then add arcs from w_1 towards $k - 1$ new vertices so that w_1 has outdegree k , and finally identify w_1 and the roots of all the forbidding gadgets constructed in previous steps. Clearly, w_1 can only have weighted outdegree 2^k by every npd-2-arc-weighting of $\overrightarrow{F_{2^k}}$. Thus, $\overrightarrow{F_{2^k}}$ is a 2^k -forbidding gadget with root w_2 . With these forbidding gadgets, our reduction scheme can then be directly adapted for the product version of the problem.

References

- [1] L. Addario-Berry, R.E.L. Aldred, K. Dalal, and B.A. Reed. Vertex colouring edge partitions. *J. Combin. Theory, Ser. B*, 94(2):237 – 244, 2005.

- [2] T. Bartnicki, J. Grytczuk, and S. Niwczyk. Weight choosability of graphs. *J. Graph Theory*, 60(3):242–256, 2009.
- [3] O. Baudon, J. Bensmail, J. Przybyło, and M. Woźniak. On decomposing regular graphs into locally irregular subgraphs. *Preprint MD 065*, <http://www.ii.uj.edu.pl/preMD/index.php>, 2013.
- [4] M. Borowiecki, J. Grytczuk, and M. Piłśniak. Coloring chip configurations on graphs and digraphs. *Inform. Process. Lett.*, 112(1-2):1–4, 2012.
- [5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [6] W. Imrich and S. Klavžar. *Product Graphs: Structure and Recognition*. Wiley-Interscience, New York, 2000.
- [7] M. Kalkowski, M. Karoński, and F. Pfender. Vertex-coloring edge-weightings: Towards the 1-2-3 conjecture. *J. Combin. Theory, Ser. B*, 100:347–349, 2010.
- [8] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *J. Combin. Theory, Ser. B*, 91(1):151 – 157, 2004.
- [9] M. Khatirinejad, R. Naserasr, M. Newman, B. Seamone, and B. Stevens. Digraphs are 2-weight choosable. *Electr. J. Comb.*, 18(1), 2011.
- [10] J. W. Moon. *Topics on tournaments*. Holt, Rinehart and Winston, 1968.
- [11] B. Seamone. The 1-2-3 conjecture and related problems: a survey. *Technical report*, available at <http://arxiv.org/abs/1211.5122>, 2012.
- [12] J. Skowronek-Kaziów. 1,2 conjecture - the multiplicative version. *Inform. Process. Lett.*, 107(3-4):93–95, 2008.