

Edge Weights and Vertex Colours: Minimizing Sum Count[☆]

Olivier Baudon^{a,b}, Julien Bensmail^c, Hervé Hocquard^{a,b},
Mohammed Senhaji^{a,b}, Éric Sopena^{a,b}

^a*Univ. Bordeaux, LaBRI, UMR5800, F-33400 Talence, France*

^b*CNRS, LaBRI, UMR5800, F-33400 Talence, France*

^c*Université Côte d'Azur, CNRS, Inria, I3S, France*

Abstract

Neighbour-sum-distinguishing edge-weightings are a way to “encode” proper vertex-colourings via the sums of weights incident to the vertices. Over the last decades, this notion has been attracting, in the context of several conjectures, ingrowing attention dedicated, notably, to understanding, which weights are needed to produce neighbour-sum-distinguishing edge-weightings for a given graph.

This work is dedicated to investigating another related aspect, namely the minimum number of distinct sums/colours we can produce via a neighbour-sum-distinguishing edge-weighting of a given graph G , and the role of the assigned weights in that context. Clearly, this minimum number is bounded below by the chromatic number $\chi(G)$ of G . When using weights of \mathbb{Z} , we show that, in general, we can produce neighbour-sum-distinguishing edge-weightings generating $\chi(G)$ distinct sums, except in the peculiar case where G is a balanced bipartite graph, in which case $\chi(G) + 1$ distinct sums can be generated. These results are best possible. When using k consecutive weights $1, \dots, k$, we provide both lower and upper bounds, as a function of the maximum degree Δ , on the maximum least number of sums that can be generated for a graph with maximum degree Δ . For trees, which, in general, admit neighbour-sum-distinguishing 2-edge-weightings, we prove that this maximum, when using weights 1 and 2, is of order $2 \log_2 \Delta$. Finally, we also establish the NP-hardness of several decision problems related to these questions.

Keywords: Neighbour-sum-distinguishing edge-weightings; Number of sums.

1. Introduction

Let W be a set of integers. For a graph G , a W -edge-weighting $\omega : E(G) \rightarrow W$ is an assignment of weights from W to the edges. From ω , one can compute, for every vertex v of G , the *sum* $\sigma(v)$ of weights incident to v , which is $\sum_{u \in N(v)} \omega(vu)$. We call ω *neighbour-sum-distinguishing* if, for every edge uv of G , we have $\sigma(u) \neq \sigma(v)$.

Neighbour-sum-distinguishing edge-weightings have been intensively studied over the last decades; as a relevant example, we refer the interested reader to e.g. the dynamic survey [4] by Gallian, in which over 250 variations of that notion, which appeared in more than 2500 references to date, are listed. These edge-weighting notions have been particularly studied with respect to several concerns. In general, the first ever question of interest is about determining, for a given graph, whether, under specific circumstances, neighbour-sum-distinguishing edge-weightings exist at all. In situations where such edge-weightings do exist, the next question is about which weights can be used to realize one.

[☆]The second author was supported by PEPS grant “POCODIS”.

For instance, if we restrict ourselves to \mathbb{N}^* -edge-weightings, one way to translate those questions is e.g. to Which graphs admit neighbour-sum-distinguishing \mathbb{N}^* -edge-weightings? and What is the least $k \in \mathbb{N}^*$ such that every such “weightable” graph admits a neighbour-sum-distinguishing $[k]$ -edge-weighting (where $[k] := \{1, \dots, k\}$)? These exact questions are related to the well-known 1-2-3 Conjecture, posed in 2004 by Karoński, Łuczak and Thomason [6], which states that every connected graph G different from K_2 verifies $\chi_\Sigma(G) \leq 3$, *i.e.*, admits a neighbour-sum-distinguishing 3-edge-weighting. In this very context, a graph with no connected component isomorphic to K_2 is called a *nice graph*, for that reason.

One way to motivate neighbour-sum-distinguishing edge-weightings is that they can be regarded as a way to encode proper vertex-colourings, *i.e.*, assignments of colours to the vertices such that no two adjacent ones get the same sum. For instance, the 1-2-3 Conjecture states that, for nearly all graphs G (with the exception of those having K_2 as a connected component), though a proper vertex-colouring might require up to $\Delta(G) + 1$ colours (by Brooks’ Theorem), encoding one via the sums derived from an edge-weighting can be done using the three weights 1, 2 and 3 only.

One consequence, however, of using few distinct weight values in a neighbour-sum-distinguishing edge-weighting ω of a graph G , is that the resulting proper vertex-colouring σ might be far from optimal, *i.e.*, the number of distinct obtained sums might be larger than $\chi(G)$, the *chromatic number* of G (being the least number of colours in a proper vertex-colouring). Consider for instance the case of a *locally irregular graph* G , *i.e.*, no two adjacent vertices of G have the same degree. Clearly, assigning 1 to every edge of G yields a neighbour-sum-distinguishing edge-weighting (so $\chi_\Sigma(G) = 1$); but the number of obtained distinct sums (colours by σ) is exactly the number of distinct degree values over the vertices of G . Obviously, this number can be arbitrarily larger than $\chi(G)$ (consider, for instance, the case where G is bipartite).

This aspect seems of interest to us, as, usually, when designing a proper vertex-colouring, we aim at getting a number of distinct colours being as close to the chromatic number as possible. Intuitively, the minimum number of distinct sums that can be obtained via a neighbour-sum-distinguishing edge-weighting is dependent of the edge weights we are allowed to use. There should thus be some trade-off between using a relatively large number of distinct edge weights and generating a relatively small number of distinct sums. To the best of our knowledge, though this aspect might have been discussed in references of the literature, we are not aware of any one dedicated to this very aspect.

This paper is dedicated to investigating those questions, which are mostly related to the following general parameter: For a given graph G and a set W of integers, we denote by $\gamma_W(G)$ the least number of distinct sums by a neighbour-sum-distinguishing W -edge-weighting of G (if any). As already mentioned, when $\gamma_W(G)$ is defined, we have $\chi(G) \leq \gamma_W(G)$; from this, two interesting and natural questions arise, namely How much larger than $\chi(G)$ can $\gamma_W(G)$ be? and For which sets W do we have $\gamma_W(G) = \chi(G)$?

Due to the number of parameters ($\chi(G)$, W , G) involved in such questions, it seems tough providing ultimate answers. Our contribution in this work is thus providing first step answers to some of them:

- In Section 2, we show, in Theorem 2.1, that we have $\gamma_{\mathbb{Z}}(G) = \chi(G)$, unless when G is a balanced bipartite graph in which case $\gamma_{\mathbb{Z}}(G) = \chi(G) + 1$. This shows that balanced bipartite graphs form a very peculiar family for our considerations.
- In Section 3, we consider these questions when using strictly positive weights $(1, \dots, k)$ only. We first provide, see summarizing Corollary 3.4, general bounds on the maximum value that $\gamma_{[k]}(G)$ can take for a graph G . For the class of nice trees T , which all

verify $\chi_\Sigma(T) = 2$, we prove, see Corollary 3.7, that the maximum value that $\gamma_{[2]}(T)$ can take is roughly $2 \log_2 \Delta(T)$.

- In Section 4, we investigate several complexity aspects related to the parameter γ_W . More precisely, we establish the NP-hardness of several decision problems arising when fixing some of the parameters involved in γ_W . Our main result, Theorem 4.6, states that, for a given bipartite graph G , determining whether $\gamma_{[2]}(G) \leq k$ is NP-hard for every $k \geq 3$. As a side result, we also establish, in Theorem 4.7, that finding the least k such that $\gamma_{[k]}(G) \leq 3$ is NP-hard for bipartite graphs G . The bipartite restriction is here important, as it was recently established that determining $\chi_\Sigma(G)$ for a bipartite graph G can be done in polynomial time [9]. Hence, adding sum restrictions is sufficient to make problems related to neighbour-sum-distinguishing edge-weightings gain a level of complexity.

2. Weighting with elements of \mathbb{Z}

As a main result in this section, we prove that $\gamma_{\mathbb{Z}}(G) = \chi(G)$ holds for every nice graph G , unless G is a balanced bipartite graph in which case $\gamma_{\mathbb{Z}}(G) = \chi(G) + 1$. Recall that a bipartite graph $G = (A \cup B, E)$ (with partite sets A and B) is said *balanced* whenever $|A| = |B|$.

Theorem 2.1. *For every nice connected graph G , we have $\chi(G) \leq \gamma_{\mathbb{Z}}(G) \leq \chi(G) + 1$. Furthermore, the upper bound is attained if and only if G is a balanced bipartite graph.*

To make the proof of Theorem 2.1 more readable, we prove all its aspects through several auxiliary results. We start off by showing that $\gamma_{\mathbb{Z}}(G) = \chi(G)$ whenever $\chi(G) \geq 3$ (Theorem 2.3). Next, we prove that $\gamma_{\mathbb{Z}}(G) \leq 3 = \chi(G) + 1$ whenever G is bipartite (Theorem 2.4). Finally, we prove that $\gamma_{\mathbb{Z}}(G) = 2 = \chi(G)$ whenever G is a connected unbalanced bipartite graph (Theorem 2.5), while $\gamma_{\mathbb{Z}}(G) > 2 = \chi(G)$ whenever G is a connected balanced bipartite graph (Theorem 2.6).

Let ω be a neighbour-sum-distinguishing edge-weighting of some graph. Throughout this paper, assuming S denotes the set of sums obtained on the vertices by ω , we call ω a *neighbour-sum-distinguishing edge-weighting with sums from S* . In most of the upcoming proofs, we will often have to modify weights through multiplications, resulting in \mathbb{Q} -edge-weightings. To eventually get \mathbb{Z} -edge-weightings, we will make use of the following obvious claim.

Observation 2.2. *Let $\alpha \neq 0$ be a non-zero integer. When multiplying all edge weights of a neighbour-sum-distinguishing $\{w_1, \dots, w_k\}$ -edge-weighting ω by α , we get a neighbour-sum-distinguishing $\{\alpha w_1, \dots, \alpha w_k\}$ -edge-weighting ω' . Furthermore, if ω takes sums from $\{s_1, \dots, s_\ell\}$, then ω' takes sums from $\{\alpha s_1, \dots, \alpha s_\ell\}$.*

Theorem 2.3. *For every connected graph G with $\chi(G) \geq 3$, we have $\gamma_{\mathbb{Z}}(G) = \chi(G)$.*

Proof. Let $\phi : V(G) \rightarrow [\chi]$ be a proper χ -vertex-colouring of G , where $\chi := \chi(G) \geq 3$. In what follows, we produce a neighbour-sum-distinguishing \mathbb{Q} -edge-weighting ω of G where $\sigma(v) = \phi(v)$ for every $v \in V(G)$. This implies the result for $\gamma_{\mathbb{Z}}(G)$, as one can then just multiply all edge weights by a same judicious integer (Observation 2.2).

We start off with ω assigning arbitrary weights (e.g. 0) to the edges of G . Obviously, ω might be far from being neighbour-sum-distinguishing and from fulfilling the required additional sum condition. We thus consider all vertices of G one by one, and, for every

considered vertex v , if $\sigma(v) \neq \phi(v)$, then we apply a fixing procedure for v , which does not alter the sums of the other vertices.

The fixing procedure is as follows. Because G is not bipartite, there has to exist an *odd-length closed walk* W containing v , *i.e.*, a cycle with odd length containing v having, possibly, vertices or edges repeating. Such a W can be found *e.g.* as follows. Since G is not bipartite, it has to contain an odd-length cycle C . If v is a vertex of C , then we can consider $W = C$. Otherwise, W can be obtained by considering a shortest path P from v to a vertex u of C , then going all the way around C back to u , and eventually going back to v through P again.

Since $\sigma(v) \neq \phi(v)$, we have $\beta := \sigma(v) - \phi(v) \neq 0$. Then we go through all edges of W starting from v , applying $+\beta/2, -\beta/2, +\beta/2, \dots$ alternately to the weights by ω to the traversed edges, until we go back to v . Note that this does not alter the sum of any vertex of W different from v (it is actually altered by 0), while the sum of v is altered by precisely $\beta/2 + \beta/2 = \beta$, due to the odd length of W . Hence we get $\sigma(v) = \phi(v)$, as required.

Repeating this procedure until all deficient vertices of G have been fixed, we eventually turn ω to a neighbour-sum-distinguishing \mathbb{Q} -edge-weighting of G with sums from $[\chi]$. \square

Theorem 2.4. *For every nice connected bipartite graph G , we have $\gamma_{\mathbb{Z}}(G) \leq 3 = \chi(G) + 1$.*

Proof. Let $G = (A \cup B, E)$ be such a nice bipartite graph, and let $s_A, s_B \in \mathbb{Z}$ be two distinct integers. We below describe a procedure for deducing (in general) a neighbour-sum-distinguishing \mathbb{Q} -edge-weighting ω such that “most” vertices of A have sum s_A while “most” vertices of B have sum s_B . Again, recall that such a neighbour-sum-distinguishing \mathbb{Q} -edge-weighting can eventually be turned into a neighbour-sum-distinguishing \mathbb{Z} -edge-weighting generating the same number of distinct sums, according to Observation 2.2.

We first describe how to prove the claim for G being a nice tree, as this case will naturally imply the whole claim. Let r be a vertex of G with degree at least 2. Regarding r as the *root* of G , we naturally come up with a natural root-to-leaves (virtual) orientation of G , where every vertex $v \neq r$ has a unique *parent*, *i.e.*, a vertex that is closer to r than v is. Conversely, v is a *child* of its parent. For every non-root vertex, we call its unique incident edge to its parent the *parent edge*. Finally, we define the *level* of any vertex to be its distance to r .

Still assuming that G is a tree, we deduce the desired ω through several modification steps. Initially, we start by assigning weight 0 to all edges of G . We then modify ω level by level from the deepest ones so that, once a level $i > 0$ has been treated, we do have the desired sum (s_A or s_B) by ω for every vertex of that level. This is done by adjusting the weight of the parent edges adequately. More precisely, assume that all vertices of the $(i+1)$ th level have been treated, and let v be a vertex of the i th level. Assume the i th level is included in A ; so the $(i+1)$ th level is included in B and its vertices currently all have sum s_B . Assuming v currently has sum s , we just assign weight $s_A - s$ to the parent edge (which current has weight 0) of v , so that v gets sum s_A . This adjustment alters the sum of the other end of the parent edge, but that vertex will be treated later in the procedure.

Assume $r \in A$. Once all vertices of the first level have been treated, we have $\sigma(a) = s_A$ for every $a \in A \setminus \{r\}$ and $\sigma(b) = s_B$ for every $b \in B$. If $\sigma(r) \neq s_B$, then we are done (we get either two sums in case $\sigma(r) = s_A$, or three sums otherwise). Otherwise, ω is not neighbour-sum-distinguishing because, for every edge rv , we have $\sigma(r) = \sigma(v) = s_B$. Recall that there are at least two such edges, by our choice of r . Assuming the sum s of r is currently strictly smaller (resp. bigger) than s_A , we add (resp. remove) $(s_A - s)/d(r)$ to (resp. from) the weight of every edge incident to r . This way, the sum of r gets equal

to s_A , while every vertex of the first level gets sum

$$s_B + (s_A - s)/d(r) \neq s_A \geq s_B + 2[(s_A - s)/d(r)]$$

(resp. $s_B - (s_A - s)/d(r) \neq s_A \leq s_B - 2[(s_A - s)/d(r)]$). So ω becomes neighbour-sum-distinguishing, and at most three sums are assigned to the vertices (being s_A , s_B , or the new sum of the vertices of the first level).

To get the claim for every nice connected bipartite graph G , it suffices to consider any spanning tree T of G , then deduce a neighbour-sum-distinguishing \mathbb{Z} -edge-weighting of T as described above, and extend it to G by assigning weight 0 to all edges of $E(G) \setminus E(T)$. \square

Theorem 2.5. *For every connected unbalanced bipartite graph G , we have $\gamma_{\mathbb{Z}}(G) = \chi(G) = 2$.*

Proof. Let G be such a graph with bipartition (A, B) , where we set $n_A := |A|$ and $n_B := |B|$. We prove that G always admits a \mathbb{Z} -edge-weighting ω such that all vertices of A have sum n_B , while all vertices of B have sum n_A . Since $n_A \neq n_B$, this ensures that ω is neighbour-sum-distinguishing, and it assigns two sums only.

For the same reasons as in the proof of Theorem 2.4, free to assign weight 0 to some edges, we may assume that G is an unbalanced tree with root $r \in A$. Recall that the level i of a vertex is its distance to r . Since we assume that $r \in A$, every i -level vertex with i odd is in B , while it is in A otherwise. For any $i \geq 1$, when referring to the *edges of the i th level* of G , we mean all edges with one end in the i th level and other end in the $(i - 1)$ th level. We denote by E_i the set of edges of the i th level. Since G is assumed to be a tree, for every vertex of any level i , there is exactly one incident edge in E_i . For every level i , we denote by x_i the number of vertices of the i th level.

We deduce ω by just applying the exact same bottom-up procedure as that described in the proof of Theorem 2.4. That is, we start from all edges assigned weight 0, and then we consider the vertices level by level from the deepest one to the first one. Whenever considering a vertex v of the i th level, it has some initial sum emerging from the weighting of the edges of E_{i+1} . We then make sure that v gets the desired sum (n_B if i is even, n_A otherwise) by altering the weight of the parent edge of v accordingly.

Once all levels have been treated, all vertices in $A \setminus \{r\}$ have sum n_B while all vertices of B have sum n_A . We show that under all our assumptions, actually also r has sum n_B ; hence our conclusion.

Let d denote the deepest vertex level, and consider the x_d vertices in level d . Assume these vertices are in A , without loss of generality. Since all these vertices have sum n_B by ω , we know that $\sum_{e \in E_d} \omega(e) = x_d \cdot n_B$. Now consider all x_{d-1} vertices in level $d - 1$. Since these vertices have sum n_A , we have

$$x_{d-1} \cdot n_A = \sum_{e \in E_d} \omega(e) + \sum_{e \in E_{d-1}} \omega(e),$$

which implies that $\sum_{e \in E_{d-1}} \omega(e) = (x_{d-1} \cdot n_A) - (x_d \cdot n_B)$.

Repeating this argument level by level starting from the deepest one, it can easily be checked that we have

$$\sum_{e \in E_i} \omega(e) = (x_i + x_{i+2} + x_{i+4} + \dots) \cdot n_A - (x_{i+1} + x_{i+3} + x_{i+5} + \dots) \cdot n_B$$

for every odd i , while we have

$$\sum_{e \in E_i} \omega(e) = (x_i + x_{i+2} + x_{i+4} + \dots) \cdot n_B - (x_{i+1} + x_{i+3} + x_{i+5} + \dots) \cdot n_A$$

for every even i . Recall that $r \in A$, and note that $\sigma(r) = \sum_{e \in E_1} \omega(e)$. From this, we have

$$\sigma(r) = (x_1 + x_3 + x_5 + \dots) \cdot n_A - (x_2 + x_4 + x_6 + \dots) \cdot n_B.$$

Since $x_1 + x_3 + x_5 + \dots = n_B$ and $x_2 + x_4 + x_6 + \dots = n_A - 1$, this yields $\sigma(r) = n_B$. \square

Theorem 2.6. *For every nice connected balanced bipartite graph G , we have $\gamma_W(G) > 2$ for any set W of integers. In particular, $\gamma_{\mathbb{Z}}(G) > 2$.*

Proof. Assume that for such a bipartite graph $G = (A \cup B, E)$, where $|A| = |B|$, there exists a neighbour-sum-distinguishing W -edge-weighting ω such that $\sigma(a) = s_A$ for all $a \in A$ and $\sigma(b) = s_B$ for all $b \in B$ (where $s_A \neq s_B$ by definition). From the point of view of A , we have $\sum_{a \in A} \sigma(a) = s_A \cdot |A|$, while, from the point of view of B , we have $\sum_{b \in B} \sigma(b) = s_B \cdot |B|$. Since G is connected, this implies that $s_A \cdot |A| = s_B \cdot |B|$, hence that $s_A = s_B$ since $|A| = |B|$. So all vertices of G have the same sum by ω , a contradiction to the fact that it is neighbour-sum-distinguishing. \square

3. Weighting with elements of \mathbb{N}^*

In this section, we now consider general properties, for a given graph G , of $\gamma_{[k]}(G)$ for fixed values of k , which finds several connections to the literature, as neighbour-sum-distinguishing k -edge-weightings (assigning strictly positive weights only) are one of the most investigated variants. We start off by establishing general bounds (involving the maximum degree) on the maximum value that $\gamma_{[k]}(G)$ can reach. Then, we focus on the particular case of $\gamma_{[2]}(T)$ for T being a nice tree, which is of interest as $\chi_{\Sigma}(T) \leq 2$ always holds.

3.1. Bounds on $\gamma_{[k]}$ in the general case

A general upper bound on $\gamma_{[k]}(G)$ can be derived for $k \geq 5$ from the best known result towards the 1-2-3 Conjecture by Kalkowski, Karoński and Pfender [5], which states that $\chi_{\Sigma}(G) \leq 5$ for every nice graph G . Since, by a neighbour-sum-distinguishing 5-edge-weighting, the sum of every vertex of G lies in $\{\delta(G), \dots, 5\Delta(G)\}$, this result directly implies the following:

Theorem 3.1. *For every nice graph G , we have $\gamma_{[5]}(G) \leq 5\Delta(G)$.*

The exact same approach yields an improved upper bound for graphs G verifying $\chi_{\Sigma}(G) < 5$. That is, $\gamma_{[\chi_{\Sigma}(G)]}(G) \leq \chi_{\Sigma}(G)\Delta(G)$. The 1-2-3 Conjecture, if true, would imply that $\gamma_{[3]}(G) \leq 3\Delta(G)$ for every nice graph G (and, by inclusion, the similar result for any set $[k]$ with $k > 3$).

More specific improvements can sometimes be obtained. For instance, nice subcubic graphs verify the 1-2-3 Conjecture [6], and we thus have $\gamma_{[3]}(G) \leq 9$ for these graphs G . Actually, the proof in [6] yields that for every proper 3-vertex-colouring $V_0 \cup V_1 \cup V_2$ of a nice 3-colourable subcubic graph G , we can deduce a neighbour-sum-distinguishing 3-edge-weighting such that the vertices in V_0 have sum 3, 6 or 9 (multiple of 3), the vertices in V_1 have sum 1, 4 or 7 (multiple of 3 plus 1), while the vertices in V_2 have sum 2, 5 or 8 (multiple of 3 plus 2). Note that, by considering a proper 3-vertex-colouring where all degree-1 vertices lie in V_0 or V_2 , we can then deduce neighbour-sum-distinguishing 3-edge-weighting such that no vertex gets sum 1. Thus, $\gamma_{[3]}(G) \leq 8$ for nice 3-colourable subcubic graphs G . Since this is also true for K_4 , this bound holds for all nice subcubic graphs as well.

A first lower bound on $\gamma_{[k]}(G)$ arises, by definition, from the connection between neighbour-sum-distinguishing edge-weightings and the proper vertex-colourings they encode. That is:

Observation 3.2. *For every nice graph G and $k \geq \chi_\Sigma(G)$, we have $\chi(G) \leq \gamma_{[k]}(G)$.*

More elaborated lower bounds on $\gamma_{[k]}(G)$ arise from the potential existence, in G , of vertices with sufficiently different degrees. Precisely, if G has two vertices u and u' with degree d and d' , respectively, such that $\{d, \dots, kd\} \cap \{d', \dots, kd'\} = \emptyset$, then, necessarily, $\sigma(u) \neq \sigma(u')$ by every neighbour-sum-distinguishing k -edge-weighting of G . Another way to have different sums in a neighbour-sum-distinguishing edge-weighting is to have several sets of vertices with about the same degree being adjacent. Using these two natural ideas, we can construct, for every $k \geq 2$, graphs with “large” value of $\gamma_{[k]}$. In particular, $\gamma_{[k]}(G)$ is not, in general, bounded away from $\chi(G)$ by an additive constant term.

Theorem 3.3. *For every $k \geq 2$, there exist graphs G with arbitrarily large maximum degree Δ (independent of k) and*

$$\gamma_{[k]}(G) \geq \left\lceil \frac{k\Delta}{k-1} \cdot \left(1 - \frac{1}{k^{\lceil \log_k \Delta \rceil}}\right) \right\rceil - \lceil \log_k \Delta \rceil.$$

Proof. For a vertex with degree d , the possible sums by a k -edge-weighting lie in $\{d, \dots, kd\}$. This implies that vertices with degree Δ , $\lfloor (\Delta - 1)/k \rfloor$, $\lfloor (\lfloor (\Delta - 1)/k \rfloor - 1)/k \rfloor$, $\lfloor (\lfloor (\lfloor (\Delta - 1)/k \rfloor - 1)/k \rfloor - 1)/k \rfloor$, etc. necessarily receive distinct sums by such an edge-weighting.

To make the computations easier, we consider a value of Δ in the series $1, k+1, k(k+1)+1, \dots$, *i.e.*, the series defined by $u_0 = 1$ and $u_n = ku_{n-1} + 1$ (which is $k^n - 1$) for $n > 0$. Consider, as G , the following graph having “many” vertices with the same degree inducing a clique. Start from disjoint complete graphs Q_1, Q_2, Q_3, \dots where Q_1 has $d_0 := \Delta$ vertices, Q_2 has $d_1 := (\Delta - 1)/k$ vertices, Q_3 has $d_2 := ((\Delta - 1)/k - 1)/k$ vertices, and so on. For each Q_i , let u_i, v_i be two distinct vertices, if any (in case Q_i has only one vertex, we define it to be v_i). Next, for every i such that Q_{i+1} exists (*i.e.*, all i 's but the last one), join Q_i and Q_{i+1} via the edge $u_i v_{i+1}$. Finally, add pendant vertices to G so that the vertices of Q_1 have degree d_0 , the vertices of Q_2 have degree d_1 , and so on.

It is not complicated checking that a graph G with such a structure indeed admits neighbour-sum-distinguishing 2-edge-weightings ω , essentially because complete graphs verify the 1-2 Conjecture [8] (each vertex sum can be altered locally by at most 2), the total version of the 1-2-3 Conjecture, and degree-1 vertices cannot be involved in sum conflicts. Furthermore, according to the arguments above, the d_0 vertices of Q_1 will get d_0 distinct sums from $\{\Delta, \dots, k\Delta\}$, the d_1 vertices of Q_2 will get d_1 distinct sums from $\{(\Delta - 1)/k, \dots, k(\Delta - 1)/k\}$, and so on, while all these sets of sums are non-intersecting due to our choice of Δ .

For each $n > 0$, we have

$$d_n = \frac{d_{n-1} - 1}{k} = \frac{d_{n-1}}{k} - \frac{1}{k} = \frac{\Delta}{k^n} - \frac{\sum_{i=0}^{n-1} k^i}{k^n} > \frac{\Delta}{k^n} - 1.$$

So, in general, the number of distinct sums by ω is at least

$$d_0 + d_1 + d_2 + \dots > \Delta + \left(\frac{\Delta}{k} - 1\right) + \left(\frac{\Delta}{k^2} - 1\right) + \dots$$

Now, since up to $\lceil \log_k \Delta \rceil$ cliques Q_i were added in order to construct G ,

$$d_0 + d_1 + d_2 + \dots + d_{\lceil \log_k \Delta \rceil} \geq \Delta \left(1 + \frac{1}{k} + \frac{1}{k^2} + \dots + \frac{1}{k^{\lceil \log_k \Delta \rceil}}\right) - \lceil \log_k \Delta \rceil,$$

which yields the claimed upper bound. \square

Summarizing Theorems 3.1 and 3.3, we have the following.

Corollary 3.4. *There are arbitrarily large values of Δ for which*

$$\left\lceil \frac{5\Delta}{4} \cdot \left(1 - \frac{1}{5^{\lceil \log_5 \Delta \rceil}}\right) \right\rceil - \lceil \log_5 \Delta \rceil \leq \max_{\{G: \Delta(G)=\Delta\}} \gamma_{[5]}(G) \leq 5\Delta.$$

3.2. Bounds on $\gamma_{[2]}$ in the tree case

It is known that, for nice trees T , we have $\chi_\Sigma(T) \leq 2$ (see [2]). A natural question is thus to wonder what is the maximum value of $\gamma_{[2]}(T)$, for a tree T . Here we prove that this value is of order $2 \log_2 \Delta(T)$.

The logarithmic lower bound can be derived from a slight modification of the construction in the proof of Theorem 3.3. This time, though, note that we cannot have many adjacent vertices with the same degree being all adjacent.

Theorem 3.5. *There exist trees T with arbitrarily large maximum degree Δ and $\gamma_{[2]}(T) \geq 2 \lceil \log_2 \Delta \rceil$.*

Proof. Choose any value of Δ in the series $1, 3, 7, 15, \dots$, i.e., the series defined recursively as $u_0 = 1$ and $u_n = 2u_{n-1} + 1$ (which is $2^n - 1$) for $n > 0$. Now consider any tree T having two adjacent vertices u_1, u'_1 with degree Δ , two adjacent vertices u_2, u'_2 with degree $(\Delta - 1)/2$, two adjacent vertices u_3, u'_3 with degree $((\Delta - 1)/2) - 1/2$, and so on. Note that, by our choice of Δ , we create $\lceil \log_2 \Delta \rceil$ such pairs u_i, u'_i .

Now, in any neighbour-sum-distinguishing 2-edge-weighting ω of T (there exist some, since $\chi_\Sigma(T) \leq 2$), necessarily u_i and u'_i get different sums for every i ; more precisely, u_1 and u'_1 get two distinct sums from $\{\Delta, \dots, 2\Delta\}$, vertices u_2 and u'_2 get two distinct sums from $\{(\Delta - 1)/2, \dots, 2(\Delta - 1)/2\}$, and so on, while it is easily seen that any two of these sets are non-intersecting due to our choice of Δ . \square

We finish off by exhibiting an algorithm showing that every nice tree T admits a neighbour-sum-distinguishing 2-edge-weighting using at most $2 \lceil \log_2(\Delta(T) - 2) \rceil + 5$ sums whenever $\Delta(T) \geq 3$. We do not consider when $\Delta(T) = 2$, as, in that case, T is a path and the exact value of $\gamma_{[2]}(T)$ can be computed easily.

Theorem 3.6. *For every nice tree T with maximum degree $\Delta \geq 3$, we have $\gamma_{[2]}(T) \leq 2 \lceil \log_2(\Delta - 2) \rceil + 5$.*

Proof. We produce a neighbour-sum-distinguishing 2-edge-weighting ω of T such that, for every vertex degree $d \in \{1, \dots, \Delta(G)\}$, all vertices with degree d (if any) have sum in a particular set S_d defined as follows. We set $S_1 := \{1, 2\}$ and $S_2 := \{2, 3, 4\}$. The remaining sets S_i with $i \geq 3$ are then defined accordingly to the following procedure: For the next $2^0 = 1$ values of i , that is for $i = 3$, we set $S_3 := \{4, 5\}$. For the next $2^1 = 2$ values of i , that is for $i = 4, 5$, we set $S_4, S_5 := \{6, 7\}$. For the next $2^2 = 4$ values of i , that is for $i = 6, 7, 8, 9$, we set $S_6, S_7, S_8, S_9 := \{10, 11\}$. In general, for the next 2^j values of i , assuming these are $i = \ell, \dots, \ell + j - 1$, we set $S_\ell, \dots, S_{\ell+j-1} := \{2\ell - 2, 2\ell - 1\}$.

The main property of interest of these sets is that, for every $i \geq 2$, there are, in S_i , two distinct values whose unique decomposition $\alpha + 2\beta$ into α 1's and β 2's (where $\alpha + \beta = i$) verifies $\alpha \geq 1$, and, similarly, two distinct values with such a decomposition fulfilling $\beta \geq 1$. Indeed, for $i = 2$, we note that $2 = 1 + 1$ and $3 = 1 + 2$ ($\alpha \geq 1$), and $4 = 2 + 2$ and $3 = 1 + 2$ ($\beta \geq 1$) fulfil this property in S_2 . For every $i \geq 3$, this follows from the fact that the

exactly two values included in S_i are, by construction, not the smallest or biggest element of the set $\{i, \dots, 2i\}$. Furthermore, we note that the set $\mathcal{S} := S_1 \cup \dots \cup S_\Delta$ contains

$$2(\lfloor \log_2(\Delta - 2) \rfloor + 1) + 3 = 2\lfloor \log_2(\Delta - 2) \rfloor + 5$$

values.

We now describe how to construct ω in such a way that it has sums from \mathcal{S} , hence our conclusion. More precisely, every vertex with degree d will take sum from S_d . We reuse the terminology used in the proof of Theorem 2.5 to describe the process. Root T at any vertex r , and define the levels from r . We propagate ω level by level, starting from level 0 (r), and going towards the leaves. Assuming r has degree d , we consider any element $s_d \in S_d$ and the unique decomposition $s_d = \alpha + 2\beta$ of s_d into 1's and 2's (where $\alpha + \beta = d$). Then we assign weight 1 to any α edges incident to r , and weight 2 to the remaining β edges. This way, we get $\sigma(r) = s_d$.

Consider now any vertex v (assumed of degree d) of level i such that all vertices from level $i - 1$ have been treated. Then the parent edge of v has already been assigned a weight x , being either 1 or 2. By construction, there are, in S_d , two distinct values s_d, s'_d whose unique decomposition $\alpha + 2\beta$ with $\alpha + \beta = d$ verifies $\alpha \geq 1$ if $x = 1$, and $\beta \geq 1$ if $x = 2$. By weighting the edges going to the children of v only, we can thus still make the sum of v reach any of s_d and s'_d (no matter what is x). Then, weighting these edges, we realize any of these two sums being different from the sum of the parent of v . This makes v being not involved in a sum conflict with its parent.

Going on this way towards the leaves, we propagate ω until all edges of T are weighted, and no sum conflict arises. Recall, in particular, that the leaves of T cannot be involved in any sum conflict. By construction, all sums obtained on the vertices belong to \mathcal{S} . \square

Corollary 3.7. *There are arbitrarily large values of Δ for which*

$$2\lfloor \log_2 \Delta \rfloor \leq \max_{\text{tree } T, \Delta(T)=\Delta} \gamma_{[2]}(T) \leq 2\lfloor \log_2(\Delta - 2) \rfloor + 5.$$

4. Complexity aspects

In this section, we consider complexity aspects related to neighbour-sum-distinguishing edge-weightings yielding particular sums. Several aspects and parameters seem of interest to us, such as 1) the edge weights to be assigned, 2) the maximum number of vertex sums we want to get, and 3) the structure of the graph to weight. By playing with some of these parameters, we get the following natural decision problems, which are the main ones we consider below:

[2]-EDGE-WEIGHTING WITH GIVEN SUMS

Input: A graph G , and a set S of sums.

Question: Does G admit a neighbour-sum-distinguishing [2]-edge-weighting with sums from S ?

[2]-EDGE-WEIGHTING WITH k SUMS

Input: A graph G .

Question: Do we have $\gamma_{[2]}(G) \leq k$?

These problems are clearly in NP, as, given an edge-weighting ω of a graph G , one can compute σ and check that all requirements (types of weights, obtained sums) are met. The whole procedure can clearly be achieved in polynomial time. Thus, below, we focus on proving the NP-hardness of those problems, to establish their NP-completeness.

For a given bipartite graph G , determining whether $\chi_\Sigma(G) \leq 2$ can be done in polynomial time [9], while the problem is NP-complete for general graphs [3]. To show that the sum requirement in our two problems above does add a level of complexity, we establish their hardness when restricted to bipartite graphs. In the first of our two complexity results, we even get a hardness restriction to locally irregular bipartite graphs, which we think is interesting as, for the original problem of determining χ_Σ , locally irregular graphs form a trivial class.

Theorem 4.1. *[2]-EDGE-WEIGHTING WITH GIVEN SUMS is NP-hard, even when restricted to instances where $|S| = 3$ and G is locally irregular and bipartite.*

Proof. We show the result by reduction from CUBIC MONOTONE 1-IN-3 SAT, which was proved NP-complete in [7]. An instance of this problem is a 3CNF formula F with positive variables only, each appearing in exactly three clauses. In other words, F can be modelled by a cubic bipartite graph with the clauses in one part, the variables in the other part, and in which an edge indicates that a given variable appears in a given clause. The question is whether F can be *1-in-3 satisfied*, meaning whether there is a *1-in-3 truth assignment*, i.e., a truth assignment to the variables such that every clause has exactly one true variable. Note that, in F , we may assume that no clause contains the same variable twice, as otherwise F could be simplified.

Note further that the formula $F' := F \wedge F \wedge F$ is 1-in-3 satisfiable if and only if F is. Furthermore, under the assumption that F is cubic, in F' all variables appear in exactly nine distinct clauses. From F' , we construct (in polynomial time) a bipartite graph $G = (A \cup B, E)$ and a 3-set S such that F' is 1-in-3 satisfiable if and only if G admits a neighbour-sum-distinguishing [2]-edge-weighting with sums from S . The construction of G is straightforward: For every clause C of F' , we add a *clause vertex* a_C to A , while, for every variable x of F' , we add a *variable vertex* b_x ; finally, for every variable x appearing in clause C of F' , we add the edge $a_C b_x$ to G . Note that G is indeed locally irregular, with degrees 3 and 9. The set S we consider is $S := \{5, 9, 18\}$.

Note that in every neighbour-sum-distinguishing [2]-edge-weighting ω with sums from S of G , necessarily all clause vertices have sum 5. This is because they have degree 3, so their possible sums lie in $\{3, 4, 5, 6\}$. Furthermore, for a degree-3 vertex to have sum 5, necessarily one incident edge must be weighted 1, while the other two incident edges must be weighted 2. On the other hand, since the variable vertices have degree 9, their possible sums lie in $\{9, \dots, 18\}$, which means that, by ω , each of them has sum 9 (all incident edges weighted 1) or 18 (all incident edges weighted 2).

To see that the equivalence between the two instances holds, just consider that having an edge $a_C b_x$ weighted 1 (resp. 2) models the fact that variable x brings value *true* (resp. *false*) to C . In ω , the fact that every clause vertex has incident edges with weights 1, 2, 2 models the fact that, by a 1-in-3 truth assignment, each clause must contain only one true variable. On the other hand, each variable vertex must be incident to edges weighted 1 only or 2 only. This models the fact that, by a 1-in-3 truth assignment, a variable brings the same truth value to all clauses containing it. From these arguments, we can easily deduce a 1-in-3 truth assignment of F' from a neighbour-sum-distinguishing [2]-edge-weighting with sums from S of G , and vice versa. \square

The proof of our main result in this section, Theorem 4.5, which establishes the NP-hardness of [2]-EDGE-WEIGHTING WITH 3 SUMS, is based on a modification of the reduction in the proof of Theorem 4.1. The construction of the reduced graph is by means of several operations and pieces, which we call *gadgets* below, connected in a particular

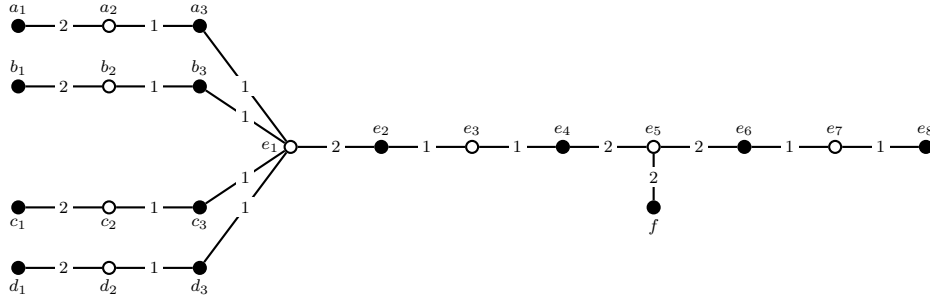


Figure 1: The initiating gadget for the proof of Theorem 4.5, and the way a neighbour-sum-distinguishing [2]-edge-weighting with three sums is propagated towards e_8 (neglecting e_8).

fashion. To ease the exposition, we first introduce all gadgets we need and point out some of their properties of interest.

Our gadgets are to be connected by identifying some of their vertices or edges. For that reason, in every upcoming proof establishing the property of a specific gadget H , the degree of some vertices of H cannot be regarded as fully established. Whenever considering [2]-edge-weightings with particular sums of H , we thus voluntarily *neglect* the sums of such vertices, meaning that such a neighbour-sum-distinguishing edge-weighting is considered correct even when a neglected vertex is involved in a sum conflict.

We start off with the *initiating gadget*, depicted in Figure 1, which we deal with using the notations given in the figure. Its vertex e_8 is called the *root*, as it will be used to attach the gadget to another graph. As described earlier, this means that the degree of e_8 will be altered upon attachment, which is why, in Proposition 4.2 below, we neglect it.

Roughly speaking, we will mainly use the initiating gadget to “force” precise sums by a neighbour-sum-distinguishing [2]-edge-weighting with at most three sums. More precisely, this gadget has the following properties:

Proposition 4.2. *Let ω be any neighbour-sum-distinguishing [2]-edge-weighting with sums from S , neglecting e_8 , of the initiating gadget. If $|S| \leq 3$, then:*

- 1) $S = \{2, 3, 6\}$;
- 2) $\omega(e_7e_8) = 1$ and $\sigma(e_7) = 2$.

Proof. Figure 1 illustrates how such an edge-weighting propagates in the initiating gadget. If $\omega(a_1a_2) = 1$, then, so that $\sigma(a_2) \neq \sigma(a_3)$, we have $\omega(a_3e_1) = 2$. In that case, S is either $\{1, 2, 3\}$ (case where $\omega(a_2a_3) = 1$), or $\{1, 3, 4\}$ (otherwise). However, we note that $d(e_1) = 5$, and, thus, it is not possible to get $\sigma(e_1) \in S$, a contradiction. So $\omega(a_1a_2) = 2$. Again, we have $\omega(a_3e_1) = 1$; and thus we have $\omega(a_2a_3) = 1$ as otherwise we would get $S = \{2, 3, 4\}$, which again would not be compatible with the degree of e_1 . These arguments apply similarly along the paths $b_1b_2b_3e_1$, $c_1c_2c_3e_1$ and $d_1d_2d_3e_1$.

So we have $\omega(a_3e_1) = \omega(b_3e_1) = \omega(c_3e_1) = \omega(d_3e_1) = 1$, and $\{2, 3\} \subset S$. Suppose now that $\omega(e_1e_2) = 1$; then $\sigma(e_1) = 5$, and $S = \{2, 3, 5\}$. So that $\sigma(e_2) \neq \sigma(e_3)$, we need $\omega(e_3e_4) = 2$, and thus $\omega(e_2e_3) = 1$ as otherwise e_3 would have sum $4 \notin S$. So $\omega(e_3e_4) = 2$ and $\sigma(e_3) = 3$; then we get to the point where there is no correct weight for e_4e_5 . Hence, we necessarily have $\omega(e_1e_2) = 2$, which implies that $\sigma(e_1) = 6$ and that $S = \{2, 3, 6\}$, which proves Item 1). By similar arguments as above, we have $\omega(e_2e_3) = \omega(e_3e_4) = 1$ and $\omega(e_4e_5) = 2$; and $\sigma(e_2) = \sigma(e_4) = 3$, while $\sigma(e_3) = 2$.

Now, since f is a leaf, we necessarily have $\omega(e_5f) = 2$; and $\omega(e_5e_6) = 2$ so that $\sigma(e_5) = 6$. Then, again, we have $\omega(e_7e_8) = 1$ so that $\sigma(e_6) \neq \sigma(e_7)$. Lastly, since

$S = \{2, 3, 6\}$, we have $\omega(e_6e_7) = 1$, which yields $\sigma(e_6) = 3$ and $\sigma(e_7) = 2$. We have $\sigma(e_8) = 1 \notin S$, but recall that we neglect e_8 . This proves Item 2). \square

An operation we need is the *fork operation*. Let G be a graph with two pendant edges uv and xw , where $d(v) = d(w) = 1$ and $u \neq x$. By *forking* uv and xw , we mean identifying v and w , and joining a new pendant vertex y to the resulting identified vertex. The edges uv and xw that served for the forking are called the *inputs* of the resulting *fork*, while the pendant edge incident to y is called its *output*.

The main property of interest of the fork operation is that forks can be used as a mechanism, given a neighbour-sum-distinguishing [2]-edge-weighting with sums from $\{2, 3, 6\}$, to check that the two input edges are assigned the same weight:

Proposition 4.3. *Let G be a graph with a fork with inputs uv, xv and output vy , and let ω be any neighbour-sum-distinguishing [2]-edge-weighting with sums from $S := \{2, 3, 6\}$, neglecting y , of G . Then:*

- 1) $\omega(uv) = \omega(xv)$;
- 2) if $\omega(uv) = \omega(xv) = 1$ and $\sigma(u), \sigma(x) \neq 3$, then $\omega(vy) = 1$ (and $\sigma(v) = 3$);
- 3) if $\omega(uv) = \omega(xv) = 2$ and $\sigma(u), \sigma(x) \neq 6$, then $\omega(vy) = 2$ (and $\sigma(v) = 6$).

Proof. Item 1) follows from the fact that if $\{\omega(uv), \omega(xv)\} = \{1, 2\}$, then $\sigma(v) \in \{4, 5\}$, and thus $\sigma(v) \notin S$. Items 2) and 3) follow from the fact that once $\omega(uv)$ and $\omega(xv)$ (which must be equal) are known, there is only one possibility for $\omega(vy)$ that makes $\sigma(v)$ to lie in S (assuming $\sigma(u)$ and $\sigma(x)$ are not equal to that value). Recall that we neglect y , so we do not have to consider whether it is involved in a sum conflict. \square

We will also have to prolong some pendant paths to make a neighbour-sum-distinguishing [2]-edge-weighting with sums from $\{2, 3, 6\}$ to propagate in a certain way. For a given graph G with a pendant edge uv (where v is the degree-1 vertex), by *delaying* uv we mean subdividing it twice, resulting in a new pendant path uwv of length 3 (where w, x have degree 2 and v has degree 1). We call xv the *output* (resulting from the delaying).

Assuming we are considering neighbour-sum-distinguishing [2]-edge-weightings with sums from $\{2, 3, 6\}$, delaying an edge is a way to “propagate” (actually invert) an edge weight, without having vertices with sum 6 nearby. This is actually a more general property of “long” pendant paths; more precisely:

Proposition 4.4. *Let G be a nice graph with a pendant path uwv of length 3, where $d(u) \geq 2$ and $d(x) = 1$, and let ω be any neighbour-sum-distinguishing [2]-edge-weighting with sums from $S := \{2, 3, 6\}$, neglecting x , of G . Then:*

- 1) $\omega(uv) \neq \omega(wx)$ and $\omega(vw) = 1$;
- 2) necessarily $\sigma(u) \in \{3, 6\}$, and:
 - 2.1) if $\sigma(u) = 6$ and $\omega(uv) = 1$, then $\sigma(w) = 3$;
 - 2.2) if $\sigma(u) = 6$ and $\omega(uv) = 2$, then $\sigma(w) = 2$;
 - 2.3) if $\sigma(u) = 3$, then necessarily $\omega(uv) = 1$, and $\sigma(w) = 3$.

Proof. Item 1) follows from the fact that if $\omega(uv) = \omega(wx)$, then we would have $\sigma(v) = \sigma(w)$; So $\omega(uv) \neq \omega(wx)$, and having $\omega(vw) = 2$ would make one of $\sigma(v), \sigma(w)$ to have value $4 \notin S$. We now prove Item 2). Since $d(u) \geq 2$, the only way to have $\sigma(u) = 2$ is to

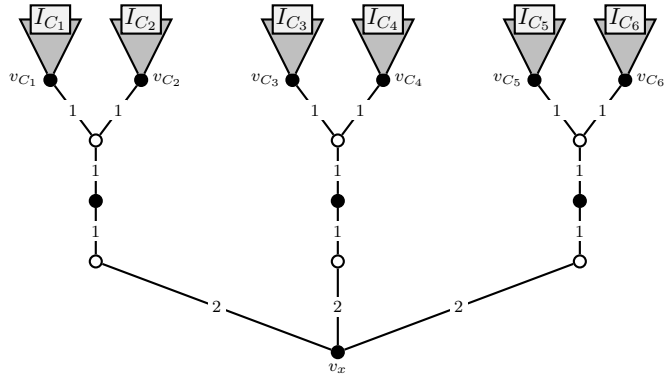


Figure 2: Illustration of (part of) the reduction in the proof of Theorem 4.5 for a variable x appearing in clauses C_1, \dots, C_6 , and the way a neighbour-sum-distinguishing [2]-edge-weighting with sums from $\{2, 3, 6\}$ is propagated towards v_x .

have $d(u) = 2$ and $\omega(uv) = 1$. Under that assumption, Item (1) implies that $\sigma(v) = \sigma(u)$, a contradiction. So $\sigma(u) \in \{3, 6\}$. Item (1) now directly implies Items (2.1), (2.2) and (2.3). We do not discuss the sum of x , as it is assumed neglected. \square

We now have all tools in hands for describing the main reduction in this section. To ease the exposition, when talking of a gadget (either initiating or fork), we mean a new copy of it.

Theorem 4.5. [2]-EDGE-WEIGHTING WITH 3 SUMS is NP-hard, even when restricted to instances where G is bipartite.

Proof. The reduction builds upon that described in the proof of Theorem 4.1. This time, though, we consider the 3CNF formula $F' := F \wedge F$, with F being an instance of CUBIC MONOTONE 1-IN-3 SAT. So every clause has exactly three variables, each variable appearing in exactly six clauses. We construct, in polynomial time, a bipartite graph G , such that F' is 1-in-3 satisfiable if and only if $\gamma_{[2]}(G) \leq 3$.

The construction of G is performed as follows. For every clause C of F' , we add one initiating gadget I_C , and call its root v_C the *clause vertex* associated to C . For every variable x in every clause C , we add an edge $v_C v_{x,C}$, where $v_{x,C}$ is a new vertex. Since all three variables in every clause C of F' are different, this yields three edges per C . We now consider all variables of F' in turn. Assume we are currently considering variable x , and let C_1, \dots, C_6 be the six distinct clauses containing x . See Figure 2 for an illustration of the upcoming explanations. We first delay each of the six edges $v_{C_1} v_{x,C_1}, \dots, v_{C_6} v_{x,C_6}$. Calling e_1, \dots, e_6 the six resulting output edges, we then fork e_1 and e_2 , then e_3 and e_4 , and then e_5 and e_6 . Let e'_1, e'_2, e'_3 denote the three resulting outputs. Next, we delay each of e'_1, e'_2, e'_3 , resulting in three output edges e''_1, e''_2, e''_3 . Finally, we identify the three pendant vertices of e''_1, e''_2, e''_3 to a single *variable vertex* v_x . Note that G is indeed bipartite, as the initiating gadget is a tree, the graph modelling the variable-clause membership of F' is bipartite, and the delaying and fork operations, when performed consistently as we did, preserve bipartiteness (as notably illustrated in Figure 2).

The equivalence between finding a 1-in-3 truth assignment ϕ of F' and finding a neighbour-sum-distinguishing [2]-edge-weighting ω with three sums of G follows from arguments similar to those used in the proof of Theorem 4.1. To make all arguments clear, let us describe, step by step, how ω behaves in G , starting from the initiating gadgets towards the variable vertices. First of all, due to the initiating gadgets in G , necessarily ω creates

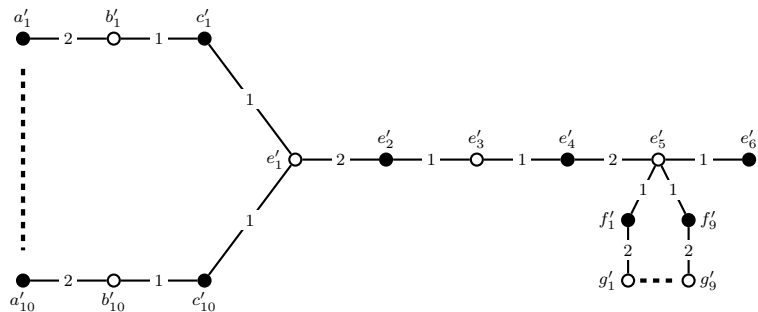


Figure 3: An extension for the initiating gadget, and the way a neighbour-sum-distinguishing [2]-edge-weighting is propagated towards e'_6 (neglecting e'_6). Vertices e'_1 and e'_5 are supposed to have degree 11, their neighbours being $e'_2, c'_1, \dots, c'_{10}$ and $e'_4, e'_6, f'_1, \dots, f'_9$, respectively.

sums from $S := \{2, 3, 6\}$, according to Proposition 4.2. Still according to Proposition 4.2, the edge e_7e_8 (where $e_8 = v_C$) of every initiating gadget I_C is weighted 1, and $\sigma(e_7) = 2$. Since $d(v_C) = 4$, necessarily we have $\sigma(v_C) = 6 \in S$. Assuming $C = (x_1 \vee x_2 \vee x_3)$, this implies that $\{\omega(v_{x_1,C}), \omega(v_{x_2,C}), \omega(v_{x_3,C})\} = \{1, 2, 2\}$, which, as in the reduction in the proof of Theorem 4.1, models the fact that, by ϕ , every clause is required to have exactly one true variable.

For every variable x of F' , all forking operations over the edges $v_{C_1}v_{x,C_1}, \dots, v_{C_6}v_{x,C_6}$ (where C_1, \dots, C_6 are the clauses containing x) were made to make sure that all these edges are assigned the same weight by ω . This is because, under all our assumptions, for every fork it is mandatory that the two inputs have the same weight, recall Proposition 4.3. The vertex v_x also serves this purpose as, if its three incident edges receive different weights, we get $\sigma(v_x) \notin S$. Using the properties exhibited in Propositions 4.4 and 4.3, it is a simple matter checking, as illustrated in Figure 2, that ω can correctly be extended towards the v_x 's under the assumption that all edges corresponding to a given variable x are weighted the same way. In particular, the delaying operation is necessary to “propagate” a given weight with making sure that the sum 6 is not used anywhere nearby. This is mandatory, notably for forks whose two inputs (and thus output) are weighted 2. \square

Slight modifications of the reduction in the proof of Theorem 4.5 yield a generalization of the result for sets of k sums, for every $k \geq 3$.

Theorem 4.6. *For every $k \geq 3$, [2]-EDGE-WEIGHTING WITH k SUMS is NP-hard, even when restricted to instances where G is bipartite.*

Proof. Assume [2]-EDGE-WEIGHTING WITH $k - 1$ SUMS is known to be NP-hard (for some $k - 1 \geq 3$), and consider proving the same claim for [2]-EDGE-WEIGHTING WITH k SUMS. The idea is to modify the reduced graph G in such a way that a unique big-degree vertex v^* is added to the graph. Indeed, if $d(v^*)$ is sufficiently larger than all the other vertex degrees, then it will force S to include a sum s^* dedicated to v^* only. Meanwhile, all the other vertices will have sums in $S \setminus \{s^*\}$, because their degrees are too small. Hence, from the point of view of the rest of the graph, this is like having S containing one less sum, which is hence equivalent to an instance of [2]-EDGE-WEIGHTING WITH $k - 1$ SUMS.

Let us give a concrete modification example for $k = 4$, as the NP-hardness of [2]-EDGE-WEIGHTING WITH 3 SUMS was established in Theorem 4.5 (but the modifications generalize naturally to bigger values of k). Let G be a graph constructed in the proof of Theorem 4.5 from an instance of CUBIC MONOTONE 1-IN-3 SAT. Note that $\Delta(G) = 5$, because of the initiating gadgets. Thus, by a [2]-edge-weighting of G , the maximum value of $\sigma(v)$ over all vertices v is 10.

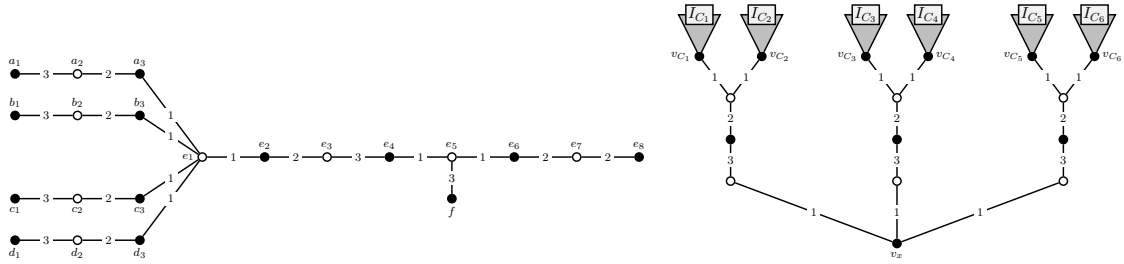


Figure 4: How to get a neighbour-sum-distinguishing 3-edge-weighting with sums from $\{3, 4, 5\}$, mentioned in the proof of proof of Theorem 4.7, of the reduced graphs constructed in the proof of Theorem 4.5.

We will add to G , without spoiling the existing degrees too much, new gadgets in such a way that G has, among others, vertices with degree 2, vertices with degree 5 and vertices with degree 11. Note that all these vertices are incompatible in terms of sums (when only weights 1 and 2 are used), in the sense that at least one sum in $\{2, \dots, 4\}$ of S has to be dedicated to the degree-2 vertices, and similarly for at least one sum in $\{5, \dots, 10\}$ for the degree-5 vertices, and for at least one sum in $\{11, \dots, 22\}$ for the degree-11 vertices. Since $|S| \leq 4$, this will actually leave at most two sums of S for the adjacent degree-2 vertices.

We remove, in G , the vertices a_1, a_2, a_3 of any one initiating gadget, and, instead, identify its vertex e_1 and the vertex e'_6 from a copy of the gadget H depicted in Figure 3. Now G has both vertices with degree 5 and 11, so the property above applies. Consider H . We claim that, in every neighbour-sum-distinguishing [2]-edge-weighting ω with sums from S' (with $|S'| \leq 3$), neglecting e'_6 , of H , necessarily 1) S' must be $\{2, 3, 12\}$, 2) $\omega(e'_5 e'_6) = 1$, and 3) $\sigma(e'_5) = 12$. If this is true, note that, in G , a copy of H acts similarly as the three edges we have removed.

So that only two sums appear onto the vertices e'_2, e'_3, e'_4 , necessarily the edges $e'_1 e'_2, e'_2 e'_3, e'_3 e'_4, e'_4 e'_5$ must be weighted 2, 2, 1, 1, or 1, 2, 2, 1, or 1, 1, 2, 2 or 2, 1, 1, 2. The first possibility is not correct, as the degree-2 vertices would already generate at least three small sums in S' . The second possibility is also not correct, as none of 1 and 2 would belong to S' , while H has degree-1 vertices. So $2, 3 \in S'$, and, in both remaining cases, $\omega(e'_4 e'_5) = 2$. Now, we know that $\omega(a'_i b'_i) = 2$ and $\omega(c'_i e'_1) = 1$ for every $i = 1, \dots, 10$, implying that $\sigma(e'_1) = 11$ (third pattern) or $\sigma(e'_1) = 12$ (fourth pattern). Similarly, $\omega(g'_i f'_i) = 2$ and $\omega(f'_i e'_5) = 1$ for every $i = 1, \dots, 9$. The third pattern is not compatible with e'_5 , since, due to its degree, we have $\sigma(e'_5) \in \{12, 13\}$. So the fourth pattern is the only way for ω to be correct, and $\omega(e'_5 e'_6) = 1$. All of Items 1), 2) and 3) are thus proved, which concludes the proof.

Let us point out that the straight generalization of this method has not impact on the bipartiteness of the reduced graphs; we thus end up with the claimed graph restriction. \square

Our proof of Theorem 4.5 also implies the NP-hardness of the following problem:

WEIGHT MINIMIZATION FOR 3 SUMS

Input: A graph G .

Question: What is the smallest k such that $\gamma_{[k]}(G) \leq 3$?

Theorem 4.7. *WEIGHT MINIMIZATION FOR 3 SUMS is NP-hard, even when restricted to instances where G is bipartite.*

Proof. This follows from the fact that every reduced graph G obtained via the reduction described in the proof of Theorem 4.5 admits a neighbour-sum-distinguishing 3-edge-weighting with sums from $\{3, 4, 5\}$. Hence, the smallest k in the definition of WEIGHT

MINIMIZATION FOR 3 SUMS is always at most 3, and it is exactly 2 if and only if the original instance F of CUBIC MONOTONE 1-IN-3 SAT is positive. To see that the upper bound of 3 indeed holds, consider the 3-edge-weighting with sums from $\{3, 4, 5\}$, neglecting e_8 , of the initiating gadget in Figure 4 (left). In G , this weighting of every such gadget I_C brings, via the edge e_7e_8 , weight 2 to v_C . Then the three remaining edges incident to v_C can be weighted 1 so that $\sigma(v_C) = 5$. The remaining edges of the graph can then be weighted correctly, as depicted in Figure 4 (right). \square

5. Conclusion and perspectives

In this work, we have studied, for some sets W , the least possible number of distinct sums that one can generate by a neighbour-sum-distinguishing W -edge-weighting of a given graph G . When $W = \mathbb{Z}$, we have proved that one can design such a weighting generating a number of sums that is either the natural lower bound, $\chi(G)$, or this lower bound plus 1, $\chi(G) + 1$. Furthermore, we were able to identify that the second value is only attained for a peculiar class of graphs, namely balanced bipartite graphs. Thus, we have completely settled the case where $W = \mathbb{Z}$.

There is much more room for improvement when $W = [k]$. Our results show that, in general, the maximum value that $\gamma_{[5]}(G)$ can take for a graph G lies in between about $\Delta(G)$ and $5\Delta(G)$. The 1-2-3 Conjecture, if shown true, would lower the upper bound to $3\Delta(G)$ for $\gamma_{[3]}(G)$, while our lower bound is also slightly improved for $k = 3$ (recall Theorem 3.3). An interesting question would thus be to investigate whether there are graphs G for which $\gamma_{[3]}(G)$ is rather close to $3\Delta(G)$.

For nice trees T , for which $\chi_{\Sigma}(T) \leq 2$, we have shown that the maximum value of $\gamma_{[2]}(T)$ is roughly $2 \log_2 \Delta(T)$. One possible direction for further research could be to consider other classes of graphs G for which $\chi_{\Sigma}(G) \leq 2$, and investigate the same question. Bipartite graphs with this property, which were recently identified [9], could be a natural candidate case.

Although bounded-degree graphs G have $\chi_{\Sigma}(G) = 3$ in general (this is even true for subcubic graphs, see [1]), the possible sums we can generate by a 3-edge-weighting is quite limited for such graphs. As a first step, we believe that determining the maximum value that $\gamma_{[3]}(G)$ can take for a subcubic graph G might be interesting. As described in Section 3, this maximum is bounded above by 8, while it is bounded below by 4 (consider e.g. K_4).

Finally, regarding the complexity results we have established in Section 4, there are still some holes. In particular, we wonder whether [2]-EDGE-WEIGHTING WITH 2 SUMS (or even [2]-EDGE-WEIGHTING WITH GIVEN SUMS with $|S| = 2$) remains NP-hard when restricted to bipartite graphs, or whether there is a polynomial-time algorithm for solving such instances.

References

- [1] A. Ahadi, A. Dehghan, M-R. Sadeghi. Algorithmic complexity of proper labeling problems. *Theoretical Computer Science*, 495:25–36, 2013.
- [2] G.J. Chang, C. Lu, J. Wu, Q. Yu. Vertex-coloring edge-weightings of graphs. *Taiwanese Journal of Mathematics*, 15(4):1807-1813, 2011.
- [3] A. Dudek, D. Wajc. On the complexity of vertex-coloring edge-weightings. *Discrete Mathematics Theoretical Computer Science*, 13(3):45-50, 2011.

- [4] J.A. Gallian. A Dynamic Survey of Graph Labeling. *Electronic Journal of Combinatorics*, DS6, 1997.
- [5] M. Kalkowski, M. Karoński, F. Pfender. Vertex-coloring edge-weightings: towards the 1-2-3 Conjecture. *Journal of Combinatorial Theory, Series B*, 100:347-349, 2010.
- [6] M. Karoński, T. Łuczak, A. Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, 91:151–157, 2004.
- [7] C. Moore, J.M. Robson. Hard Tiling Problems with Simple Tiles. *Discrete and Computational Geometry*, 26(4):573-590, 2001.
- [8] J. Przybyło, M. Woźniak. On a 1,2 Conjecture. *Discrete Mathematics and Theoretical Computer Science*, 12(1):101-108, 2010.
- [9] C. Thomassen, Y. Wu, C.-Q. Zhang. The 3-flow conjecture, factors modulo k , and the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 121:308-325, 2016.