Discrete Mathematics and Theoretical Computer Science ....
.... **vol. 27:3 #7 (2025)**

Pushing Vertices to Make Graphs Irregular

Julien Bensmail[1] Clara Marcille[2] Mano Orenga[3]

[1] *Université Côte d'Azur, CNRS, Inria, I3S, Nice, France*
[2] *Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Bordeaux, France*
[3] *Univ. Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR 5668, Lyon, France*

revisions 18th Dec. 2024, 31st July 2025; accepted 8th Aug. 2025.

In connection with the so-called 1-2-3 Conjecture, we introduce and study a new problem related to proper labellings. In the regular problem, proper labellings of graphs are designed by assigning strictly positive labels to the edges so that any two adjacent vertices get incident to distinct sums of labels, and the main goal, for a given graph, is to minimise the value of the largest label assigned. In the new problem we introduce, we construct proper labellings through pushing vertices, where pushing a vertex means increasing by 1 the labels assigned to all edges incident to that vertex. We focus on the study of two related metrics of interest, being the total number of times vertices have been pushed, and the maximum number of times a vertex has been pushed, which we aim at minimising, for given graphs. As a contribution, we establish bounds, some of which are tight, on these two parameters, in general and for particular graph classes. We also prove that minimising any of the two parameters is an NP-hard problem. Finally, we also compare our new problem with the original one, and raise directions and questions for further work on the topic.

Keywords: 1-2-3 Conjecture, proper labelling, pushing vertices.

1 Introduction

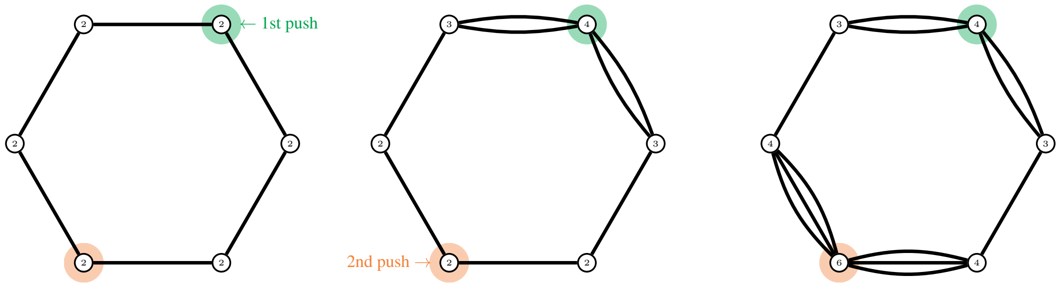
In this work, we introduce a new problem where, given some graph, one aims at designing a **proper labelling** of it by **pushing its vertices**. While the latter concept is a new one, the former one has been well investigated in literature. For this reason, we start in what follows with recalling what proper labellings are about.

The concept of graph regularity is fairly common and understood in graph theory, as *regular graphs* are defined as those graphs in which all vertex degrees are the same. It is pretty natural, now, to wonder how a natural antonym to the notion of graph regularity could be defined, and there are multiple ways to do so. It is well known, in particular, that, besides K_1 (the complete graph of order 1), no simple graph can be totally irregular, in the sense that all its vertex degrees are pairwise distinct. This led authors to consider other notions of graph irregularity over the last decades, such as the notions of *highly irregular graphs* in Alavi et al. (1987) or *locally irregular graphs* in Baudon et al. (2015). Since, in this work, we are primarily interested in the latter notion, let us recall that a graph is *locally irregular* if no two of its adjacent vertices have the same degree. That is, in that notion of graph irregularity, said irregularity is more of a local one. And, obviously, simple graphs can be locally irregular.

In the line of problems and concepts introduced in Chartrand et al. (1988), the notion of proper labelling arises when wondering about ways to make simple graphs somewhat irregular. For the sake of introducing the next notions more naturally, let us focus on local irregularity. If a simple graph G is not locally irregular, then the authors of Chartrand et al. (1988) are interested in making G locally irregular by essentially multiplying its edges, where *multiplying* an edge e of G means replacing it by a certain number of parallel edges, or, in other words, by increasing the multiplicity $\mu(e, G)$ of e in G. In other words, we are interested in finding a locally irregular multigraph M having somewhat the same adjacencies as in G, in the sense that two vertices are adjacent in G if and only if they are adjacent in M.

As is, studying this graph transformation problem is not convenient, which leads to studying it under a labelling point of view, suggested in Chartrand et al. (1988). For a graph G and some $k \geq 1$, a *k-labelling* (or *labelling* for short, if k is unneeded) $\ell : E(G) \rightarrow \{1, \dots, k\}$ is an assignment of labels (from $\{1, \dots, k\}$) to the edges of G. For every vertex u of G, we now define $\sigma(u)$ as the *sum* of labels assigned by ℓ to the

ISSN 1365–8050



(a) Initial configuration (b) After pushing green once (c) After pushing orange twice

Fig. 1: Sequential construction of a proper pushing scheme of C_6. From the initial configuration (a), the vertex highlighted with green is pushed once, resulting in (b). Then, the vertex highlighted with orange is pushed twice, resulting in (c). It can be checked (numbers in vertices indicate their degrees) that the resulting multigraph is locally irregular, and thus that the depicted pushing scheme is indeed proper. Due to this proper pushing scheme, we have $\mathrm{P}^{\mathrm{t}}(C_6) \leq 3$ and $\mathrm{P}^{\mathrm{l}}(C_6) \leq 2$.

edges incident to u, that is, $\sum_{v \in N(u)} \ell(uv)$. And we finally say that ℓ is *proper* if σ indeed forms a proper vertex-colouring of G, that is, if $\sigma(u) \neq \sigma(v)$ for any two adjacent vertices u and v of G. It is not too complicated to observe that turning G into a locally irregular multigraph through edge multiplications is equivalent to finding a proper labelling of G, as, essentially, edge labels model edge multiplicities, while the properness requirement models the local irregularity requirement.

Several aspects of proper labellings have been investigated in literature. In particular, following thoughts from Chartrand et al. (1988), multiplying an edge too many times might be perceived as a costly operation, and, from this point of view, one might be interested in minimising the maximum number of times an edge is multiplied, or, under the labelling point of view, the maximum label value assigned. For a graph G, we are thus interested in the parameter $\chi_\Sigma(G)$, which is the smallest $k \geq 1$ (if any) such that proper k-labellings of G exist. For the sake of keeping the current introduction short, we will not elaborate too much here on what is known on these concerns, and instead refer the interested reader to surveys on the topic such as Bensmail (2020); Gallian (2000); Seamone (2012). First off, one must know that $\chi_\Sigma(G)$ is well defined whenever G is *nice*, that is, does not contain any connected component isomorphic to K_2. For long, the main conjecture on χ_Σ was the 1-2-3 Conjecture from Karoński et al. (2004), which was asserting we should have $\chi_\Sigma(G) \leq 3$ for all nice graphs G. Note that this conjecture was supported in particular by the fact that deciding whether $\chi_\Sigma(G) \leq 2$ holds for a given graph G is NP-complete, see Dudek and Wajc (2011). Constant upper bounds on the parameter χ_Σ have been decreased over the years following the introduction of the 1-2-3 Conjecture, until the conjecture was fully proved recently in Keusch (2024). Please be aware that more information on this topic will be provided throughout, as they are needed to get the extent of the results we present.

As described earlier, the study of proper labellings is mainly motivated by the fact that it models another problem, where one aims at making simple graphs locally irregular by multiplying their edges. From a more general, algorithmic point of view, this way of creating irregularity can be regarded through a different angle, which will motivate our upcoming new notions and concepts. Namely, let G be a graph, and M be a locally irregular multigraph obtained from G by multiplying some of its edges. Algorithmically speaking, note that M can be obtained from G by repeatedly considering a vertex u with an incident edge $e = uv$ that does not have the correct multiplicity (*i.e.*, for the current auxiliary multigraph, G', we have $\mu(e, G') < \mu(e, M)$) and increasing the multiplicity of e by 1. Likewise, this can be described under the labelling point of view, by noting that any proper labelling ℓ of G can be obtained from all edges of G being assigned label 1, by repeatedly considering a vertex u with an incident edge $e = uv$ that does not have the correct label (*i.e.*, it is strictly less than $\ell(e)$), and increasing the label of e by 1.

This way of describing things is pretty natural, and it leads us to considering the following generalisation, which might be regarded as a new way to introduce irregularity in graphs. Namely, throughout this work, we aim at making graphs locally irregular by repeatedly considering some vertex and increasing by 1 the multiplicity of all its incident edges. This basic operation of increasing by 1 the multiplicity of all edges incident to some vertex, we call a *push*[(i)], and since it is obvious that the order in which vertices of a graph

[(i)] Note that the term "push" is inspired by other concepts of graph theory. Notably, in digraphs, pushing vertices means reversing the are pushed does not matter (w.r.t. the resulting edge multiplicities), the main notion we will investigate throughout is that of **pushing schemes**, where, for a graph G, a *pushing scheme* $\rho : V(G) \rightarrow \mathbb{N}$ is a function indicating, for every vertex $u \in V(G)$, the number of times u is pushed. Clearly, to ρ then corresponds a natural labelling $\ell = \ell(\rho)$ of G, where for every edge $e = uv$ of G, we have $\ell(uv) = 1 + \rho(u) + \rho(v)$. Since we are interested in pushing schemes yielding locally irregular multigraphs, we say ρ is *proper* if $\ell(\rho)$ is proper.

When studying proper labellings, the main goal is to design one so that the maximum label assigned is as small as possible. Although there is a proper labelling associated to every proper pushing scheme, it is obvious that not all proper labellings of a graph correspond to a proper pushing scheme, and because of that we feel that it would be daring to investigate proper pushing schemes minimising the maximum resulting label. Instead, we prefer to regard pushes as expensive operations which one thus aims at minimising. This leads us to the two main parameters we will investigate throughout this work, defined as follows for a graph G (see Figure 1 for an illustration):

- $\mathrm{P^t}(G) = \min \left\{ \sum\limits_{u \in V(G)} \rho(u) : \rho \text{ is a proper pushing scheme of } G \right\}$;
- $\mathrm{P^l}(G) = \min \left\{ \max\limits_{u \in V(G)} \rho(u) : \rho \text{ is a proper pushing scheme of } G \right\}$.

Through the first parameter, $\mathrm{P^t}$, note that we are interested in proper pushing schemes minimising the total number of pushes, while the second parameter, $\mathrm{P^l}$, is more of a local one, as it deals with minimising the maximum number of times a vertex is pushed, through a proper pushing scheme. That is, regarding the latter parameter, we are more particularly interested in proper k-pushing schemes with $k \geq 1$ as small as possible, where we define a *k-pushing scheme* as a pushing scheme where every vertex is pushed at most k times. As will be proved later on in Section 2, all nice graphs admit proper pushing schemes, so $\mathrm{P^t}(G)$ and $\mathrm{P^l}(G)$ are well defined for all nice graphs G. Also, it turns out that bounds on $\mathrm{P^t}$ can be expressed in terms on $\mathrm{P^l}$, which justifies why, throughout this work, we mostly focus on the latter parameter.

This work is organised as follows. We start in Section 2 by raising first, general observations and results on the two parameters $\mathrm{P^t}$ and $\mathrm{P^l}$ (covering their behaviours, their connections, and first general bounds). We then focus on common classes of nice graphs in Section 3, covering trees, complete graphs, and others. Section 4 is dedicated to the complexity of determining $\mathrm{P^t}(G)$ and $\mathrm{P^l}(G)$ for a given graph G. In particular, we prove that the two corresponding problems are NP-complete. We finish off with a conclusion in Section 5, in which we raise open questions and problems for further work on the topic.

2 First observations and results

We begin by making it clear that $\mathrm{P^t}$ and $\mathrm{P^l}$ are well defined for nice graphs only. In other words, we prove that nice graphs admit proper pushing schemes. In the proof and afterwards, for a graph G with a vertex u, and for a pushing scheme ρ of G, when writing $\sigma(u)$ we refer to the same parameter by $\ell(\rho)$, which is nothing but

$$d(u)(1 + \rho(u)) + \sum_{v \in N(u)} \rho(v).$$

Recall that $\Delta(G)$ denotes the maximum degree of G.

Theorem 2.1. *Every connected graph G with $\Delta(G) \geq 2$ admits proper pushing schemes. Consequently, a graph admits proper pushing schemes if and only if it is nice.*

Proof: Let (O, L) be the partition of $V(G)$ where O contains all degree-1 vertices of G and L contains all other vertices. Since $\Delta(G) \geq 2$, we have $|L| \geq 1$. Now let $\mathcal{S} = (v_1, \dots, v_p)$ be an arbitrary ordering over the vertices of L. Let us consider ρ, the pushing scheme of G obtained as follows.

- We do not push any vertex u of O; that is, $\rho(u) = 0$.
- We consider the v_i's one by one in order, following $\mathcal{S}$. For every v_i considered this way, we push, by ρ, vertex v_i a certain number of times, so that:

direction of all their incident arcs. Obviously, this is unrelated to our concerns.

1. the resulting $\sigma(v_i)$ is different from all $\sigma(v_j)$ such that $j < i$ and $v_iv_j \in E(G)$;
2. for every $j < i$ such that $v_iv_j \in E(G)$, the resulting $\sigma(v_j)$ is different from all $\sigma(v_k)$ such that $k \in \{1, \dots, i\} \setminus \{i, j\}$ and $v_jv_k \in E(G)$.

We claim these conditions can always be achieved, by pushing v_i sufficiently many times. Indeed, regarding the first condition, since $d(v_i) \geq 2$, note that pushing v_i exactly once increases $\sigma(v_i)$ by at least 2, while, for every v_j adjacent to v_i, this increases $\sigma(v_j)$ by only 1. Regarding the second condition, for some fixed v_j adjacent to v_i, either v_k is not adjacent to v_i, in which case pushing v_i modifies $\sigma(v_j)$ but not $\sigma(v_k)$; or v_k is adjacent to v_i, in which case pushing v_i modifies both $\sigma(v_j)$ and $\sigma(v_k)$ the same way, but we know $\sigma(v_j) \neq \sigma(v_k)$ due to vertices treated earlier.

Now, because, through any pushing scheme, a degree-1 vertex cannot have the same sum as its unique neighbour (unless that neighbour is of degree 1), we get that, after treating v_p in the process above, the resulting ρ is proper, as desired.

The last part of the statement follows from the fact that K_2, which is the only connected graph G with $\Delta(G) = 1$, obviously admits no proper pushing scheme at all. □

As mentioned in the introductory section, throughout this work, out of our two new parameters, we focus on understanding the more local parameter, P^{l}, mostly because it can be used to upper bound the more global one, P^{t}. Indeed, we obviously have:

Observation 2.2. *For every nice graph G with order n, we have*

$$\mathrm{P}^{\mathrm{l}}(G) \leq \mathrm{P}^{\mathrm{t}}(G) \leq n\mathrm{P}^{\mathrm{l}}(G).$$

Another reason why we believe focusing on the parameter P^{l} might be more interesting, is that, to date, proper k-labellings have been mostly investigated for small values of k. More precisely, due to the 1-2-3 Conjecture, proper labellings that have been investigated the most are proper 3-labellings. Now, $\mathrm{P}^{\mathrm{t}}(G)$ might be large for a graph G, and, consequently, it is hard to retrieve what is the maximum label assigned by any proper labelling associated to a proper pushing scheme realising $\mathrm{P}^{\mathrm{t}}(G)$. On the other hand, if ρ is a proper k-pushing scheme of G realising $\mathrm{P}^{\mathrm{l}}(G)$ (that is, $k = \mathrm{P}^{\mathrm{l}}(G)$), then we know for sure that the associated proper labelling $\ell(\rho)$ is a $(2k + 1)$-labelling. Due to the 1-2-3 Conjecture and previous investigations, an appealing case is thus that of graphs G with $\mathrm{P}^{\mathrm{l}}(G) = 1$, which thus admit proper 3-labellings. These will notably be discussed in later Section 4.

Due to Observation 2.2, one may think that, maybe, for general nice graphs G, designing proper pushing schemes realising $\mathrm{P}^{\mathrm{l}}(G)$ is a good way to design proper pushing schemes realising $\mathrm{P}^{\mathrm{t}}(G)$. We prove this is not the case. For a more formal statement, we will employ the notation $\mathrm{P}^{\mathrm{l,t}}(G)$, which denotes, for a graph G, the minimum total number of pushings among all proper $\mathrm{P}^{\mathrm{l}}(G)$-pushing schemes of G.

Theorem 2.3. *There are connected graphs G for which all proper pushing schemes realising $\mathrm{P}^{\mathrm{l}}(G)$ are, in terms of total number of pushes, arbitrarily bad w.r.t. $\mathrm{P}^{\mathrm{t}}(G)$. In other words, minimising the maximum number of times a vertex is pushed can be arbitrarily bad w.r.t. the minimum total of pushes we can achieve, for a proper pushing scheme. That is, $\mathrm{P}^{\mathrm{l,t}}(G)/\mathrm{P}^{\mathrm{t}}(G)$ can be arbitrarily large, for a graph G.*

Proof: Let $x \geq 8$, and consider T_x the tree obtained as follows.

- We start from a vertex u, being adjacent to x vertices $v_1, \dots, v_x$.
- For every $i \in \{1, \dots, x\}$, we make v_i adjacent to seven new vertices $a_i, b_i, c_i, d_i, e_i, f_i, g_i$. We then make v_i adjacent to $x - 8 \geq 0$ new leaves. This way, $d(v_i) = x$.
- For every $i \in \{1, \dots, x\}$, we add new leaves adjacent to a_i until $d(a_i) = x$. We then do the same with other vertices (in $\{b_i, c_i, d_i, e_i, f_i, g_i\}$) so that $d(b_i) = d(c_i) = x + 1$, $d(d_i) = d(e_i) = x + 2$, $d(f_i) = 2x - 1$, and $d(g_i) = 2x$.

Due to this structure, note that $T_x - u$ has x connected components, each containing one of the v_i's. For every $i \in \{1, \dots, x\}$, we denote by S_i the subtree of T_x containing the vertices of the connected component of $T_x - u$ containing v_i.

Note that the v_i's have the same degree as u, and as the a_i's; thus, T_x is not locally irregular, and $\mathrm{P}^{\mathrm{l}}(T_x) > 0$. We claim $\mathrm{P}^{\mathrm{l}}(T_x) = 1$. To see this is true, just consider the pushing scheme ρ of T_x where we push exactly once all d_i's and e_i's. As a result, we still have $\sigma(u) = x$, while $\sigma(v_i) = x + 2$ for all $i \in \{1, \dots, x\}$, while $\sigma(a_i) = x$, $\sigma(b_i) = \sigma(c_i) = x + 1$, $\sigma(d_i) = \sigma(e_i) = 2(x + 2)$, $\sigma(f_i) = 2x - 1$, and $\sigma(g_i) = 2x$. Recall also that leaves cannot be involved in sum conflicts, so they cannot cause conflicts here.

We claim also that for any such proper 1-pushing scheme ρ of T_x, for every $i \in \{1, \dots, x\}$ there must be at least one vertex of S_i pushed once. Indeed, towards a contradiction, assume w.l.o.g. this is not the case for $i = 1$. Then we have either $\sigma(v_1) = x$ (if u is not pushed) or $\sigma(v_1) = x + 1$ (otherwise). So, we must have a conflict between v_1 and either a_1 or b_1, respectively, a contradiction to ρ being proper. Thus, if ρ is a proper 1-pushing scheme of T_x, then at least one vertex of each S_i is pushed, and the total number of pushes is at least x. So, $\mathrm{P}^{\mathrm{l,t}}(T_x) \geq x$.

Now consider the pushing scheme of T_x obtained by just pushing u exactly three times. As a result, for every $i \in \{1, \dots, x\}$ we get $\sigma(v_i) = x + 3$, while we still have $x + 3 \notin \{\sigma(a_i), \dots, \sigma(g_i)\}$ (note indeed that pushing u did not alter the sums in this set). Meanwhile, $\sigma(u) = 4x$. This pushing scheme is thus proper, and we have $\mathrm{P}^{\mathrm{t}}(G) \leq 3$. Since $\mathrm{P}^{\mathrm{l,t}}(T_x) \geq x$, the result follows. □

Now that we have clarified a bit the relationship between the two parameters P^{t} and P^{l}, and explained why focusing on the latter may be more interesting for now, we focus on providing a first upper bound. Actually, our proof of Theorem 2.4 yields the following:

Theorem 2.4. *If G is a nice graph, then* $\mathrm{P}^{\mathrm{l}}(G) \leq \Delta(G)^2$.

Proof: Consider the proper pushing scheme ρ we construct in the proof of Theorem 2.1. Recall that the only vertices u with $\rho(u) > 0$ are those from L. Consider thus any $v_i \in L$. We claim that, when dealing with v_i, there is an integer $x \in \{0, \dots, \Delta(G)^2\}$ such that, upon pushing v_i exactly x times, we do not create any of the conflicts described in items 1. and 2. in the proof of Theorem 2.1. Namely:

1. For each v_j with $j < i$ that is adjacent to v_i, there is at most one integer $y \in \{0, \dots, \Delta(G)^2\}$ such that we get $\sigma(v_i) = \sigma(v_j)$ upon pushing v_i exactly y times (recall $d(v_i) \geq 2$). Since there are at most $\Delta(G)$ such v_j's, such constraint thus forbid at most $\Delta(G)$ values from $\{0, \dots, \Delta(G)^2\}$ as x.

2. For every $j < i$ such that $v_i v_j \in E(G)$, and every $k < i$ such that $k \in \{1, \dots, i\} \setminus \{i, j\}$ and $v_j v_k \in E(G)$ (and $v_i v_k \notin E(G)$), there is at most one integer $y \in \{0, \dots, \Delta(G)^2\}$ such that we get $\sigma(v_j) = \sigma(v_k)$ upon pushing v_i exactly y times. Since there are at most $\Delta(G)$ such v_j's, and, for each such fixed v_j, at most $\Delta(G) - 1$ such v_k's, such constraints thus forbid at most $\Delta(G)(\Delta(G) - 1)$ values from $\{0, \dots, \Delta(G)^2\}$ as x.

In total, all these constraints around v_i thus forbid at most $\Delta(G) + \Delta(G)(\Delta(G) - 1) = \Delta(G)^2$ as x. Since $|\{0, \dots, \Delta(G)^2\}| = \Delta(G)^2 + 1$, there is thus a desired x in $\{0, \dots, \Delta(G)^2\}$, and we have our conclusion. □

Observation 2.2 and Theorem 2.4 now yield:

Corollary 2.5. *If G is a nice graph with order n, then* $\mathrm{P}^{\mathrm{t}}(G) \leq n\Delta(G)^2$.

Now that we have Theorem 2.4 and Corollary 2.5, one could wonder, in general, about accurate ways to express bounds on the parameters P^{t} and P^{l}. For reasons explained earlier, we will mostly investigate this question for the latter parameter. As will be seen in the next section, there is no constant upper bound on the parameter P^{l}, and we can even come up with examples of graphs G where $\mathrm{P}^{\mathrm{l}}(G) = \Delta(G)$, for $\Delta(G)$ being arbitrarily large. For this reason, and due to Theorem 2.4, we think that bounding P^{l} in terms of the maximum degree might be the way to go. We feel the following might hold true:

Conjecture 2.6. *If G is a nice graph, then* $\mathrm{P}^{\mathrm{l}}(G) \leq \Delta(G)$.

3 Some classes of graphs

In this section, we mostly focus on studying the parameter P^{l} for particular graph classes. As a consequence, we get to confirming Conjecture 2.6 for certain graph classes, and, notably through Observation 2.2, we get to establishing upper bounds on the more global parameter P^{t}. In particular, we consider complete bipartite graphs, complete graphs, graphs with maximum degree 2, and trees.

3.1 Complete bipartite graphs

Since complete bipartite graphs $K_{n,m}$ have a very restricted, well-identified structure, proper labellings are completely understood here. In particular, see *e.g.* Chang et al. (2011), since $K_{n,m}$ is locally irregular if $n \neq m$, we have $\chi_\Sigma(K_{n,m}) = 1$ whenever $n \neq m$. Otherwise, when $n = m$, a proper 2-labelling can be obtained by choosing any vertex, making all its incident edges labelled 2, and keeping all other edges assigned label 1; thus, $\chi_\Sigma(K_{n,m}) = 2$ when $n = m$. Such a proper labelling can, obviously, be realised through pushing vertices. From this fact, we can thus derive the following easy, optimal result:

Theorem 3.1. *For any $n \geq 1$ and $m \geq 2$ with $n \leq m$, we have $\mathrm{P}^{\mathrm{l}}(K_{n,m}) = 0$ if $n \neq m$, and $\mathrm{P}^{\mathrm{l}}(K_{n,m}) = 1$ otherwise.*

Proof: As mentioned above, $K_{n,m}$ is locally irregular when $n \neq m$, in which case $\mathrm{P}^{\mathrm{l}}(K_{n,m}) = 0$. When $n = m$, just pick any vertex u, and push it once. Then, we have $\sigma(u) = 2n$, while for all vertices v in the same part as u we have $\sigma(v) = n$, and for all vertices v in the other part we have $\sigma(v) = n+1$. Since $n \geq 2$ by niceness of $K_{n,m}$ in this case, the resulting 1-pushing scheme is proper and the claim follows. □

From the proof of Theorem 3.1, note that we also deduce that $\mathrm{P}^{\mathrm{t}}(K_{n,m}) = 0$ when $n \neq m$, and $\mathrm{P}^{\mathrm{t}}(K_{n,m}) = 1$ otherwise, which obviously is best possible. This illustrates a situation where the lower bound in Observation 2.2 is tight, and where the parameter P^{t} does not have to be large (which is rather expected, again, given the structure of $K_{n,m}$).

Note also that we get that Conjecture 2.6 holds for complete bipartite graphs.

3.2 Complete graphs

When it comes to nice complete graphs K_n (thus $n \geq 3$), it is well known, see *e.g.* Chang et al. (2011), that $\chi_\Sigma(K_n) = 3$. Actually, for any $k \geq 3$, there is a nice, easy way to design proper k-labellings of K_n: start from the edges of K_3 being assigned pairwise distinct labels in $\{1, \dots, k\}$, and, repeatedly, letting $i \in \{1, k\}$ be any label such that no vertex of the current complete graph has all its incident edges assigned label i, add a universal vertex with all its edges assigned label i. After any number $x \geq 0$ of steps, this results in a proper k-labelling of K_{3+x}. And since we can essentially start the process with any k-labelling of K_3, there are plenty of such proper k-labellings, as k and x grow.

Unfortunately, designing labellings through pushing vertices is a very peculiar operation, and it can be checked that one cannot construct labellings as above through pushing vertices. The situation is even worse, in the following sense:

Observation 3.2. *For any $n \geq 3$, a pushing scheme of K_n is proper if and only if every two vertices are not pushed the same number of times. Consequently, we have $\mathrm{P}^{\mathrm{l}}(K_n) = n - 1 = \Delta(K_n)$, and thus there is no fixed $k \geq 1$ such that $\mathrm{P}^{\mathrm{l}}(K_n) \leq k$ for all $n \geq 3$.*

Proof: Since, for any two distinct vertices u and v of K_n, we have $N(u) \setminus \{v\} = N(v) \setminus \{u\}$, and, by any pushing scheme ρ, we have

$$\sigma(u) = (n-1)(\rho(u) + 1) + \sum_{w \in N(u)} \rho(w)$$

and

$$\sigma(v) = (n-1)(\rho(v) + 1) + \sum_{w \in N(v)} \rho(w),$$

so that $\sigma(u) \neq \sigma(v)$ we must have $\rho(u) \neq \rho(v)$. So, ρ is proper if and only if any two vertices of K_n are not pushed the same number of times. The proper pushing scheme of K_n that minimises the maximum number of times a vertex is pushed, is thus that where vertices are pushed $0, 1, \dots, n-1$ times. Hence, $\mathrm{P}^{\mathrm{l}}(K_n) = n - 1 = \Delta(K_n)$. □

Observation 3.2 justifies why we believe Conjecture 2.6 might be a reasonable conjecture. Another consequence of Observation 3.2 is that, by Observation 2.2, we deduce that $\mathrm{P}^{\mathrm{t}}(K_n) \leq n\Delta(K_n) = \Delta(K_n)(\Delta(K_n) + 1)$ holds for every nice complete graph K_n. However, note that the exact arguments in the proof of Observation 3.2 show that we have

$$\mathrm{P}^{\mathrm{t}}(K_n) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \frac{1}{2}\Delta(K_n)(\Delta(K_n) + 1).$$

This also illustrates that the upper bound in Observation 2.2 is far from optimal, in general.

3.3 Graphs with maximum degree 2

In the context of connected graphs with maximum degree 2, proper labellings are very easy to design, as, for a labelling of a path P_n (of order n) or cycle C_n (of order/size n) to be proper, all is required is that edges at distance 2 apart get assigned distinct labels. In particular, we have $\chi_\Sigma(P_3) = 1$, $\chi_\Sigma(P_n) = 2$ for all $n > 3$, $\chi_\Sigma(C_n) = 2$ for all $n \geq 3$ with $n \equiv 0 \bmod 4$, and $\chi_\Sigma(C_n) = 3$ otherwise (see *e.g.* Chang et al. (2011)). This can also be exploited in our context. In particular, for paths, we can exploit the natural degeneracy to get an easy, optimal result.

Theorem 3.3. *For any* $n \geq 3$*, we have* $\mathrm{P}^{\mathrm{l}}(P_n) = 0$ *if* $n = 3$*, and* $\mathrm{P}^{\mathrm{l}}(P_n) = 1$ *otherwise.*

Proof: If $n = 3$, then P_n is locally irregular and thus $\mathrm{P}^{\mathrm{l}}(P_n) = 0$; otherwise, P_n is not locally irregular and thus $\mathrm{P}^{\mathrm{l}}(P_n) > 0$. When $n > 3$, we obtain a proper 1-pushing scheme ρ of P_n in the following way. Consider the vertices $v_1, \dots, v_n$ of P_n in order, from one end-vertex to the other. Note that, initially, $\sigma(v_1) = 1$ and $\sigma(v_2) = 2$, so there is no conflict between v_1 and v_2. Now, for every $i \geq 3$ considered in order, we have that v_{i-2} and v_{i-1} exist. If v_{i-2} and v_{i-1} are not in conflict, then proceed with v_{i+1}. Otherwise, just push v_i once to get rid of that conflict. Eventually, it should be clear that ρ is proper. In particular, since we treat the v_i's in order, any possible conflict between two adjacent vertices v_{i-2} and v_{i-1} of degree 2 was dealt with upon reaching v_i. It should be noted as well that, by the definition of pushing schemes, a degree-1 vertex cannot be in conflict with its unique neighbour. Thus, ρ is proper, and the claim follows. □

Things are a bit more tricky for cycles C_n. In particular, a point worth mentioning is that there is a straight classification, in terms of P^{l}, w.r.t. the parity of n; this contrasts a bit with χ_Σ, for which the classification is made modulo 4.

Theorem 3.4. *For any* $n \geq 3$*, we have* $\mathrm{P}^{\mathrm{l}}(C_n) = 1$ *if* n *is even, and* $\mathrm{P}^{\mathrm{l}}(C_n) = 2$ *otherwise.*

Proof: Throughout, we denote by $v_0, \dots, v_{n-1}$ the consecutive vertices of C_n (where, all along, operations over the subscripts of the v_i's are modulo n). Since C_n is not locally irregular, we have $\mathrm{P}^{\mathrm{l}}(C_n) > 0$. Actually, we claim $\mathrm{P}^{\mathrm{l}}(C_n) > 1$ whenever n is odd. Indeed, for some $n \geq 3$, consider ρ, a proper 1-pushing scheme of C_n. A crucial observation is that, because all vertices of C_n have degree 2, if $\rho(v_i) = \rho(v_{i+1})$ for some $i \in \{0, \dots, n-1\}$, then we must have $\{\rho(v_{i-1}), \rho(v_{i+2})\} = \{0, 1\}$; this is because, otherwise, we would have $\sigma(v_i) = \sigma(v_{i+1})$. Now, since $\mathrm{P}^{\mathrm{l}}(C_n) > 0$, some vertices of C_n must be pushed by ρ while some others are not. From these observations, we get that if we denote by (P, Q) the partition of $V(C_n)$ where P contains the vertices pushed by ρ while Q contains the others, then all $p \geq 1$ connected components $P_1, \dots, P_p$ of $C_n[P] = C_n - Q$ contain exactly one or three vertices each, and similarly for all $q \geq 1$ connected components $Q_1, \dots, Q_q$ of $C_n[Q] = C_n - P$. In other words, along C_n, the v_i's are arranged into alternating blocks of one or three vertices each, with the same status w.r.t. ρ. Thus, $p = q$, and $|P| \equiv |Q| \bmod 2$. So $|P| + |Q| = n$ must be even, from which we get that ρ cannot exist if n is odd.

We are now ready to prove the whole claim. Set $n = 4k + r$, where $r = n \bmod 4$. Consider ρ, the pushing scheme of C_n obtained by pushing exactly once every v_i in $\{v_0, \dots, v_{4k-1}\}$ with $i \equiv 0 \bmod 4$. If $r = 0$, then we obtain $\sigma(v_i) = 4$ for all $i \equiv 0 \bmod 4$, $\sigma(v_i) = 3$ for all $i \equiv 1, 3 \bmod 4$, and $\sigma(v_i) = 2$ for all $i \equiv 2 \bmod 4$; this implies ρ is proper here. When $r > 0$, we also perform the following:

- If $r = 1$, then we also push v_{4k} twice. Then $\sigma(v_{4k-1}) = 4$, $\sigma(v_{4k}) = 7$, and $\sigma(v_0) = 6$.
- If $r = 2$, then we also push each of v_{4k} and v_{4k+1} once. Then $\sigma(v_{4k-1}) = 3$, $\sigma(v_{4k}) = 5$, $\sigma(v_{4k+1}) = 6$, and $\sigma(v_0) = 5$.
- If $r = 3$, then we also push v_{4k} twice. Then $\sigma(v_{4k-1}) = 4$, $\sigma(v_{4k}) = 6$, $\sigma(v_{4k+1}) = 4$, $\sigma(v_{4k+2}) = 3$, and $\sigma(v_0) = 4$.

As a result, ρ is proper in each case, and ρ pushes vertices at most once (when n is even) or twice (otherwise). By earlier arguments, this is best possible, and the claim holds. □

For both of Theorems 3.3 and 3.4, in terms of the parameter P^{t}, note that the proper pushing schemes we design push about a quarter of the vertices, at most once (except for odd-length cycles, for which a small, constant number of additional pushes are performed). Thus, we believe the tight values of $\mathrm{P}^{\mathrm{t}}(P_n)$ and $\mathrm{P}^{\mathrm{t}}(C_n)$ should be about $\lfloor n/4 \rfloor$, in general. Certainly this should be provable through arguments similar to those employed in the proofs of Theorems 3.3 and 3.4; we voluntarily do not provide full proofs because we doubt this would be too interesting to the reader. Again, a bound of about $\lfloor n/4 \rfloor$ would be better than a simple application of previous Observation 2.2.

The fact that, for nice paths and cycles, we can design proper 2-pushing schemes, also implies the following:

Corollary 3.5. *Conjecture 2.6 holds for nice graphs G with $\Delta(G) = 2$.*

3.4 Trees

Recall that nice trees G have $\chi_\Sigma(G) \leq 2$, which is tight in general (see *e.g.* Chang et al. (2011)). A proper 2-labelling ℓ of any nice tree G can actually be constructed quite easily, taking advantage of the bipartite structure for G. More precisely, one can construct such an ℓ with the property that vertices in one part of the bipartition all have odd sums while vertices in the other part all have even sums, except perhaps for leaves which, as was remarked in a previous proof, can never be in conflict with their unique neighbour since G is nice and we are assigning strictly positive labels only. For instance, such an ℓ can be obtained by starting from all edges assigned label 1, choosing any vertex r, applying a BFS algorithm from r, and for every non-leaf vertex u with at least one child v considered during the run, changing, if needed, the label of uv to 2 if the parity of $\sigma(u)$ is wrong.

It turns out that constructing proper labellings following a proper k-vertex-colouring of the vertices (that is, making sure that vertices in distinct parts have distinct sums modulo k) is a very common approach, see *e.g.* Karoński et al. (2004), and one can thus wonder whether this can be applied in our context. While this can be tricky in most contexts, due to the fact that pushing vertices is a very constraining way of building labellings, trees actually form a context where such ideas can be set to work. This leads to the next result.

Theorem 3.6. *If G is a nice tree, then $\mathrm{P}^{\mathrm{l}}(G) \leq 1$.*

Proof: As described above, let $\psi : V(G) \to \{0,1\}$ be a proper $\{0,1\}$-vertex-colouring of G, and choose any vertex r of G. We construct a proper 1-pushing scheme ρ of G by essentially considering vertices in any order provided by a BFS algorithm performed from r, and pushing vertices, if needed, so that $\sigma(v) \equiv \psi(v) \bmod 2$ for all vertices $v \in V(G)$, saved maybe for leaves, which can never be in conflict with their unique neighbour.

Let v be any vertex considered during the course of the BFS algorithm. If $r = v$, then we do nothing. Otherwise, let u denote the parent of v in G w.r.t. the BFS algorithm, that is, u is the unique vertex of G that is adjacent to v and was considered earlier in the process. If $\sigma(u) \equiv \psi(u) \bmod 2$, then we do nothing. Otherwise, we push v exactly once, so that $\sigma(u)$ increases by exactly 1 and thus u has a desired sum modulo 2. Once all vertices have been treated, note that we have $\sigma(u) \equiv \psi(u) \bmod 2$ for all non-leaf vertices u, since $\sigma(u)$ was perhaps adjusted, if needed, when considering one of its neighbours v later in the process. As mentioned earlier, recall that we might have $\sigma(u) \not\equiv \psi(u) \bmod 2$ when u is a leaf, but we know, since strictly positive labels are assigned by $\ell(\rho)$, that u cannot have the same sum as its unique neighbour. Thus, ρ is a proper 1-pushing scheme of G. □

By Observation 2.2 and Theorem 3.6, we obtain that every nice tree G with order n fulfils $\mathrm{P}^{\mathrm{t}}(G) \leq n$. Obviously, this bound does not seem quite accurate again; an argument is *e.g.* that, for the proper 1-pushing schemes we build in the proof of Theorem 3.6, every vertex can have at most two vertices in its neighbourhood pushed (its parent and one of its children, w.r.t. the BFS ordering). In particular, if G, leaves apart, has all its vertices having degree at least $d \geq 3$, then the upper bound above is immediately improved to $\lfloor n/d \rfloor$. This might indicate that nice paths could be the most annoying trees in terms of P^{t}. Following our discussion by the end of Subsection 3.3, we thus suspect we might have $\mathrm{P}^{\mathrm{t}}(G) \leq \lfloor n/4 \rfloor$ for all nice trees G with order n, since, along long paths, we need to push about a quarter of the vertices anyways.

Regarding Conjecture 2.6, we also get a confirmation for trees, from Theorem 3.6.

4 Complexity aspects

In this section, we study the complexity of determining any of $\mathrm{P}^{\mathrm{t}}(G)$ and $\mathrm{P}^{\mathrm{l}}(G)$, for a given graph G. More precisely, we prove an NP-completeness result for both parameters.

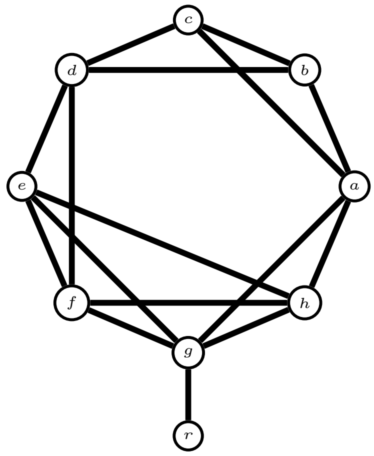


Fig. 2: The M-gadget.

4.1 Determining P^1

The problem we consider in this section is the following (defined for any $k \geq 0$):

k-PUSHABILITY
Input: a graph G.
Question: does G have a proper k-pushing scheme? That is, do we have $\mathrm{P}^1(G) \leq k$?

Note that an instance of 0-PUSHABILITY is positive if and only if G is locally irregular; thus, this problem can be solved in polynomial time. In what follows, we prove 1-PUSHABILITY is NP-complete. For comparison, determining whether $\chi_\Sigma(G) \leq k$ holds for a graph G is NP-complete for $k = 2$ (see *e.g.* Dudek and Wajc (2011)), but polynomial-time solvable for all $k \neq 2$ (see Keusch (2024)). A point of interest also, is that we saw earlier that $\mathrm{P}^1(G) \leq 1$ holds for several common graphs G. Thus, one could legitimately wonder whether graphs with this property are "easy" to characterise. Our result implies this is not the case, unless P=NP.

Before we get to proving our main result here, we need some preparation first. Namely, we need to introduce a crucial gadget to be used in our reduction. This graph, which we call the *M-gadget* throughout, is depicted in Figure 2. All along, assuming no ambiguity is possible, we will deal with the vertices and edges of this gadget using the very terminology from the figure. Copies of the M-gadget will be used by attaching them, through their vertex r, in given graphs. Formally, assuming we have a graph G with a vertex v, by *attaching an M-gadget at v*, we mean adding a copy M of the M-gadget to the graph, and identifying v and r. The point is that if the resulting graph admits proper 1-pushing schemes, then such pushing schemes behave in a particular way around M. More formally:

Lemma 4.1. *Let M be a copy of the M-gadget, and ρ be a 1-pushing scheme of M. If (omitting the possible conflict between r and g) ρ is proper, then:*

- $\rho(r) = \rho(g) = 0$*, and*
- $\sigma(g) = 7$.

Furthermore, such a proper 1-pushing scheme ρ of M exists.

Proof: First off, note that $N(b) \setminus \{c\} = N(c) \setminus \{b\}$. Thus, so that $\sigma(b) \neq \sigma(c)$, it must be that exactly one vertex in $\{b, c\}$ is pushed. For a similar reason, exactly one vertex in $\{e, f\}$ must be pushed. W.l.o.g., assume b and e are pushed once by ρ, while c and f are not. Note now that, so far, a has one neighbour, b, pushed once, one, c, not pushed, and two other neighbours, g and h. Likewise, h has one neighbour, e, pushed once, one, f, not pushed, and two other neighbours, g and a. Thus, so that $\sigma(a) \neq \sigma(h)$ it must be that exactly one vertex in $\{a, h\}$ is pushed. We claim this vertex must be a. Indeed, assume, towards a contradiction, that h is pushed once while a is not. Note that, at this point, if d is not pushed, then $\sigma(b) = \sigma(d) = 6$ since we have identified the status (pushed once or not) of all vertices in $N(b) \cup N(d) \setminus \{d\}$. Thus, assume d is pushed once. For the same reasons, but regarding a and b, observe that g must not be

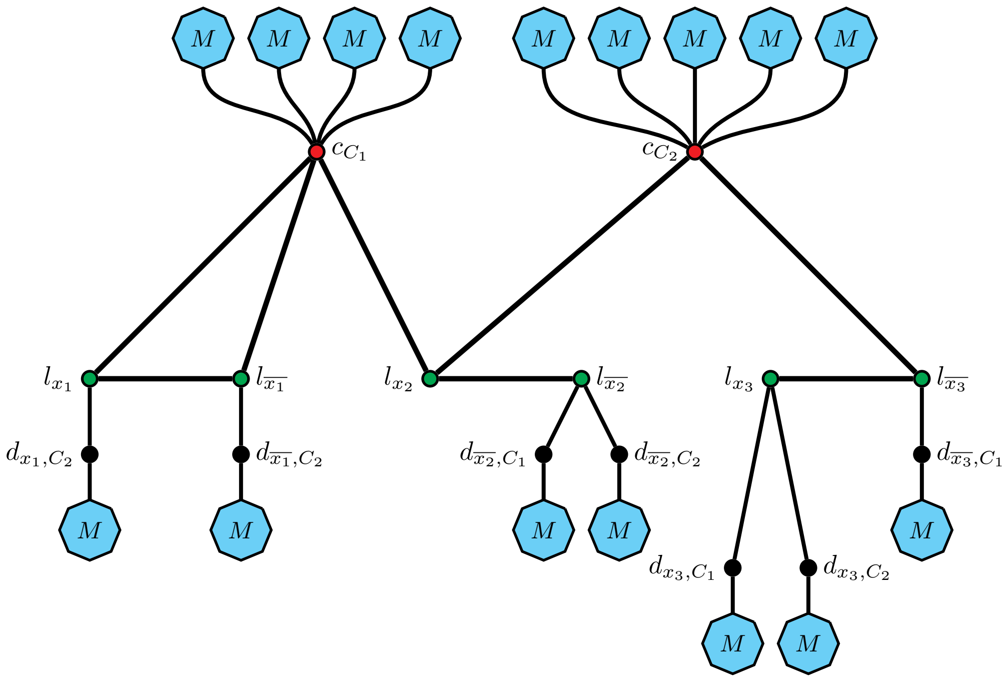


Fig. 3: Illustration of the reduction in the proof of Theorem 4.2, for the formula $F = C_1 \wedge C_2$ where $C_1 = (x_1 \vee \overline{x_1} \vee x_2)$ and $C_2 = (x_2 \vee x_2 \vee \overline{x_3})$ (the set of variables being thus $\{x_1, x_2, x_3\}$). Red vertices are clause vertices, while green vertices are literal vertices. Blue octagonal shapes with an "M" inside indicate where M-gadgets are attached.

pushed, as otherwise we would have $\sigma(a) = \sigma(b) = 7$. But then we necessarily have $\sigma(e) = \sigma(d) = 10$, a contradiction.

Thus, we can now go back to the situation where a, b, e are pushed while c, f, h are not. Regardless of whether g is pushed or not, note that d must not be pushed, as otherwise we would have $\sigma(f) = \sigma(h) \in \{6, 7\}$. Likewise, now, g must not be pushed as otherwise we would get $\sigma(f) = \sigma(d) = 6$. In turn, r must not be pushed, as otherwise we would have $\sigma(g) = \sigma(e) = 8$. As a result, we obtain $\sigma(a) = 9$, $\sigma(b) = 7$, $\sigma(c) = 5$, $\sigma(d) = 6$, $\sigma(e) = 8$, $\sigma(f) = 5$, and $\sigma(g) = 7$, thus no further objection to the properness of ρ. □

Lemma 4.1 implies that if we attach an M-gadget at some vertex v in some graph G, then, assuming the resulting graph admits a proper 1-pushing scheme ρ, necessarily v is not pushed by ρ, and we cannot have $\sigma(v) = 7$.

We are now ready to prove our main result here.

Theorem 4.2. 1-PUSHABILITY *is NP-complete.*

Proof: Since the problem is obviously in NP, we focus on proving its NP-hardness. This is done by reduction from the 3-SATISFIABILITY problem, which is well known to be NP-hard. Recall that an instance of 3-SATISFIABILITY consists of a conjunction F of m clauses $C_1, \ldots, C_m$ each being a disjunction of three literals defined over n variables $x_1, \ldots, x_n$ (that is, every literal is either a variable or a negated variable). The task is to determine if there is a truth assignment to the variables of F such that each clause is satisfied, *i.e.*, has a true literal. From an instance F of 3-SATISFIABILITY, we construct, in polynomial time, a graph G such that F can be satisfied if and only if G admits proper 1-pushing schemes.

The construction of G goes as follows (see Figure 3 for an illustration). We start off from G having vertices and edges depicting the structure of F. That is, for every clause C of F we have a *clause vertex* c_C in G, for every literal l over the variables of F we have a *literal vertex* l_l, and whenever some literal l belongs to some clause C of F we have the *formula edge* $l_l c_C$ in G. Now:

- For every variable x of F, we add the edge $l_x l_{\overline{x}}$ to G.

- For every clause C of F, assuming C contains exactly x distinct literals, we attach, in G, exactly $7 - x$ M-gadgets at c_C. As a result, $d(c_C) = 7$.
- For every possible literal l over the variables of F, and for every clause C of F that does not contain l, we add a new vertex $d_{l,C}$ to G, attach an M-gadget at $d_{l,C}$, and add the edge $l_l d_{l,C}$. As a result, $d(l_l) = m + 1$.

Regarding the last item, note that, since we can suppose m (the total number of clauses of F) is not constant (as otherwise F could be solved by exhaustively checking all possible solutions), we can assume throughout $m \geq 9$, and thus $d(l_l) > 9$ for every possible literal l. Note also that the whole construction of G is clearly achieved in polynomial time.

To see that we have the desired equivalence between F and G, let us state some properties that a proper 1-pushing scheme ρ of G must fulfil (assuming such a ρ exists):

1. By Lemma 4.1, besides vertices within the attached M-gadgets, note that the only vertices of G that can be pushed by ρ are literal vertices.
2. For every pair $\{l_l, l_{\bar{l}}\}$ of opposite literal vertices of G, recall that we have $d(l_l) = d(l_{\bar{l}})$. Since, by the previous item, only literal vertices of G can be pushed by ρ, so that $\sigma(l_l) \neq \sigma(l_{\bar{l}})$, it must be that exactly one vertex in $\{l_l, l_{\bar{l}}\}$ is pushed. This implies also that we must have $\{\sigma(l_l), \sigma(l_{\bar{l}})\} = \{m + 2, 2(m + 1)\}$, and, thus, $\sigma(l_l), \sigma(l_{\bar{l}}) \geq 11$.
3. Each clause vertex c_C has degree 7, is adjacent to at least one literal vertex, and has at least one M-gadget attached. By Lemma 4.1 and the previous items, c_C cannot be pushed by ρ, and, actually, in $N(c_C)$ only literal vertices adjacent to c_C can be pushed. Furthermore, still by Lemma 4.1, in the attached M-gadgets there must be neighbours of c_C with sum 7 by ρ. So, so that c_C is not in conflict with its such neighbours, at least one literal vertex adjacent to c_C must be pushed by ρ. Since c_C has one, two, or three such neighbours, we deduce $\sigma(c_C) \in \{8, 9, 10\}$.
4. Every vertex $d_{l,C}$ has degree 2, has an M-gadget attached, and is adjacent to a literal vertex. By Lemma 4.1 and the previous items, we must have $\sigma(d_{l,C}) \in \{2, 3\}$.

From all these arguments, the desired equivalence between F and G is easy to visualise. Indeed, imagine that, by a proper 1-pushing scheme ρ of G, having some literal vertex l_l pushed models that literal l is assigned truth value *true* by some truth assignment ϕ to F, and *false* otherwise. For every pair $\{l_l, l_{\bar{l}}\}$ of opposite literal vertices of G, the property in the second item above models that, by ϕ, a variable and its negation are assigned distinct truth values. The properties in the third item above models that, by ϕ, a clause is considered satisfied if it has at least one true literal. Thus, a satisfying truth assignment to the variables of F can be determined from a proper 1-pushing scheme of G, and *vice versa*. In particular, recall that the M-gadgets attached in G can be pushed as required (recall Lemma 4.1), and observe that, by a proper pushing scheme ρ of G behaving as described above, literal vertices can be in conflict with neither clause vertices nor $d_{l,C}$'s. □

4.2 Determining P^{t}

For any fixed $k \geq 0$, note that the problem of determining whether $\mathrm{P}^{\mathrm{t}}(G) = k$ holds for a given graph G can be solved in polynomial time, by essentially considering all $|V(G)|^k$ ways to push k vertices of G, and checking whether the corresponding pushing scheme is proper. Thus, the problem of determining whether $\mathrm{P}^{\mathrm{t}}(G) \leq k$ holds for any $k \geq 0$ and given graph G can be solved easily, which justifies, herein, to focus on the following problem (where, in opposition to k-PUSHABILITY, the parameter of interest, k, is part of the input):

MINIMUM TOTAL PUSHABILITY
Input: an integer $k \geq 0$, and a graph G.
Question: does G have a proper pushing scheme performing, in total, at most k pushes? That is, do we have $\mathrm{P}^{\mathrm{t}}(G) \leq k$?

We prove MINIMUM TOTAL PUSHABILITY is NP-complete in general.

Theorem 4.3. MINIMUM TOTAL PUSHABILITY *is NP-complete.*

Proof: The problem is clearly in NP; thus, we focus on proving its NP-hardness. As in the proof of Theorem 4.2, this is done by reduction from the 3-SATISFIABILITY problem. Given an instance F of 3-SATISFIABILITY, we construct, in polynomial time, a graph G such that F can be satisfied if and only if G admits a proper pushing scheme performing at most k pushes in total, for some k being a (polynomial) function of G.

As in the proof of Theorem 4.2, we start from G being the graph modelling the structure of F (thus with clause vertices c_C, literal vertices l_l, formula edges, and edges joining each pair of opposite literals). We further modify G as follows:

- As in the proof of Theorem 4.2, for every literal l over the variables of F not appearing in some clause C, we add the edge $l_l d_{l,C}$, where $d_{l,C}$ is here a new degree-1 vertex. Thus, at this point, for every literal l of F, in G we have $d(l_l) = 1 + m$ (where, recall, m is the total number of clauses in F).
- For every clause C of F, we make, in G, clause vertex c_C adjacent to a new vertex c'_C, which, assuming C contains $x \in \{1, 2, 3\}$ distinct literals, we make adjacent to x new degree-1 vertices. This way, we get $d(c_C) = d(c'_C) = x + 1$.
- Set $d = m+n(n+1)+1$ (where, recall, m and n are the number of clauses and variables, respectively, of F). For every literal l over the variables of F, and every $i \in \{d, \dots, d + n\} \setminus \{d + 1\}$, we make, in G, literal vertex l_l adjacent to $n + 1$ new vertices $a_{l,i,1}, \dots, a_{l,i,n+1}$. Then, for every $i \in \{d, \dots, d + n\} \setminus \{d + 1\}$ and $j \in \{1, \dots, n + 1\}$, we add $i - 1$ new degree-1 vertices adjacent to $a_{l,i,j}$ so that $d(a_{l,i,j}) = i$. Thus, eventually, every literal vertex l_l of G satisfies $d(l_l) = d$, and is adjacent (among others) to $n + 1$ vertices of degree $d, d + 2, d + 3, \dots, d + n$, respectively.

Clearly, the whole construction of G is achieved in polynomial time. For the instance of MINIMUM TOTAL PUSHABILITY to be complete, the integer k we consider is n.

Let us remark that, in G, initially the only pairs of adjacent vertices with the same degree are: every pair $\{l_l, l_{\bar{l}}\}$ for some literal l of F, every pair $\{c_C, c'_C\}$ for some clause C of F, and every pair $\{l_l, a_{l,d,i}\}$ (for all $i \in \{1, \dots, n + 1\}$) for some literal l of F. Also, we can assume the degree, d, of the literal vertices is strictly larger than the degree, which is at most 4, of the clause vertices, as otherwise we must have $m = 1$ and F is trivially satisfiable.

The key point for the equivalence between F and G (and additional input $k = n$) to hold, is the fact that, in any proper pushing scheme ρ of G performing at most n pushes, for every variable x of F it must be that exactly one vertex in $\{l_x, l_{\overline{x}}\}$ is pushed. Towards a contradiction, assume this is not the case. Recall that l_x is adjacent to and has the same degree as $l_{\overline{x}}$ and $a_{x,d,1}, \dots, a_{x,d,n+1}$, and $l_{\overline{x}}$ is also adjacent and has the same degree as $a_{\overline{x},d,1}, \dots, a_{\overline{x},d,n+1}$. Each of l_x and $l_{\overline{x}}$ is also adjacent to $n+1$ vertices with degree $d+2, \dots, d+n$ (of the form $a_{x,i,j}$ and $a_{\overline{x},i,j}$, respectively, for $i \in \{d + 2, \dots, d + n\}$ and $j \in \{1, \dots, n + 1\}$). Now, by performing at most n pushes (not pushing l_x or $l_{\overline{x}}$) by ρ, note that $\sigma(l_x)$ and $\sigma(l_{\overline{x}})$ cannot leave the set $\{d, \dots, d+n\}$. Also, note that, for any $i \in \{d, \dots, d+n\} \setminus \{d+1\}$, we cannot push all of $a_{x,i,1}, \dots, a_{x,i,n}$ or $a_{\overline{x},i,1}, \dots, a_{\overline{x},i,n}$. This implies that, whatever n pushes (not pushing l_x or $l_{\overline{x}}$) are performed by ρ, for every $i \in \{d, \dots, d + n\} \setminus \{d + 1\}$, there must be, by ρ, some $a_{x,i,j}$ and some $a_{\overline{x},i,j}$ with $\sigma(a_{x,i,j}) = i$ and $\sigma(a_{\overline{x},i,j}) = i$. Since by pushing once any neighbour of l_x or $l_{\overline{x}}$ we increase $\sigma(l_x)$ or $\sigma(l_{\overline{x}})$, respectively, by exactly 1, due to the restricted number of pushes, and because the smallest available sums (which must be different) for l_x and $l_{\overline{x}}$ are $d + 1$ and $d + n + 1$, we cannot get rid of all conflicts involving l_x and $l_{\overline{x}}$ by pushing their neighbours only. We thus have a contradiction; thus, ρ must push at least one vertex in $\{l_x, l_{\overline{x}}\}$. On the other hand, note that if, say, l_x is pushed once while $l_{\overline{x}}$ is not, then, assuming no other vertex in their neighbourhoods is pushed, we get $\sigma(l_x) = 2d > d + n$ and $\sigma(l_{\overline{x}}) = d + 1$, which two values do not appear, as is, as the degree of any neighbour of l_x and $l_{\overline{x}}$, respectively.

Now, by a proper pushing scheme ρ of G performing at most n pushes, we must have the following properties:

1. By the arguments above, at least one vertex in any pair $\{l_x, l_{\overline{x}}\}$ must be pushed. Since there are exactly n such pairs, this implies, for every variable x of F, that ρ pushes exactly once exactly one vertex in $\{l_x, l_{\overline{x}}\}$. Again due to the restricted number of pushes, this implies only literal vertices are pushed, at most once each.
2. For every clause C of F, recall that $d(c_C) = d(c'_C)$ in G. By the previous item, to get rid of this conflict by ρ, there must be a formula edge $c_C l_l$ such that $\rho(l_l) = 1$.

3. Still due to the restricted number of pushes, and the distinct degrees, note that clause vertices and adjacent literal vertices cannot be in conflict by ρ. Likewise, any l_l cannot be in conflict with an adjacent $d_{l,C}$. All other non-discussed adjacencies include a degree-1 vertex, which, by ρ, cannot be in conflict with its neighbour.

From these arguments, we have the desired equivalence between satisfying F, and finding a proper pushing scheme ρ of G performing at most n pushes. We regard the fact that ρ pushes once some literal vertex l_l as setting literal l to *true*, and setting it to *false* otherwise. The fact that exactly one of l_l and $l_{\bar{l}}$ must be pushed once by ρ for every literal l of F thus models that, by a truth assignment, a variable and its negation receive distinct truth values. The second property above then models that, by a truth assignment to the variable of F, a clause is considered satisfied only if it has at least one true literal. From these arguments, and the third set of properties above, we get that a satisfying truth assignment to the variables of F can be deduced from a proper pushing scheme of G performing at most n pushes, and *vice versa*. □

5 Conclusion

In this work, we have introduced the concept of proper pushing schemes and the two associated parameters P^{l} and P^{t}, which we have studied in general and for several graph classes. Recall that our main intent was to study particular types of proper labellings, namely those which can be obtained by the somewhat natural pushing operation. We focused more precisely on the parameter P^{l}, which we feel is more natural and manageable to study, and came up with Conjecture 2.6. This conjecture we verified for several graph classes along Section 3. We also focused on more algorithmic aspects, and proved, through Section 4, that determining either of P^{l} and P^{t} is tough in general (unless P=NP).

Our results only stand as a very first approach towards understanding proper pushing schemes, and the problems we considered here only form a very restricted sample of all directions we could have investigated. In particular, we believe the following questions and directions are appealing, and could deserve further consideration.

- Regarding Conjecture 2.6, one could consider establishing a bound on P^{l} that is a linear function of the maximum degree. Recall indeed that we only proved a quadratic bound, through Theorem 2.4. Let us mention that Theorem 2.4 can be improved slightly in some contexts, by refining the proof. For instance, when G is connected with maximum degree $\Delta = \Delta(G)$ but G is not Δ-regular, then it is better to consider, as $\mathcal{S} = (v_1, \dots, v_p)$, an ordering over the vertices of L such that each v_i is adjacent to at least one vertex of O or some v_j with $i < j$, and $d(v_p) < \Delta$. Such an $\mathcal{S}$ can be obtained *e.g.* by reversing the order in which the vertices of L are traversed during the course of a BFS algorithm performed from any vertex of degree strictly less than Δ. This implies that every v_i now has at most $\Delta - 1$ neighbours v_j with $j < i$; thus the number of constraints in the first item becomes at most $\Delta - 1$, while it becomes at most $(\Delta - 1)^2$ in the second one. The bound deduced here is thus $\mathrm{P}^{\mathrm{l}}(G) \leq \Delta - 1 + (\Delta - 1)^2 = \Delta^2 - \Delta$.

  This apart, it could be good to confirm Conjecture 2.6 for more classes of graphs. Recall that, in Corollary 3.5, we proved the conjecture for graphs of maximum degree 2; thus, as a first step, perhaps one could wonder about graphs of maximum degree 3. If G is a nice graph with $\Delta(G) = 3$, then we have $\mathrm{P}^{\mathrm{l}}(G) \leq 9$ by Theorem 2.4, while according to the conjecture we should have $\mathrm{P}^{\mathrm{l}}(G) \leq 3$. Thus, there is some gap here.

- In the very same line, we proved Conjecture 2.6 for graphs with very low degeneracy, recall Corollary 3.5 and Theorems 3.6, and we wonder whether one can prove the conjecture for other classes of graphs with this property. For instance, we wonder about 2-degenerate graphs, or more generally about graphs with bounded maximum average degree (mad). From a more general perspective, we wonder whether one could express bounds on P^{l} in terms of the maximum average degree. In particular, we were not able to come up with graphs G such that $\mathrm{P}^{\mathrm{l}}(G)$ is larger than $\mathrm{mad}(G)$.

  We would also like to bring up the particular case of cacti (graphs in which no two cycles share an edge), which are 2-degenerate and often form a natural class to consider after considering trees. Because cacti are 2-degenerate, they admit proper 3-vertex-colourings. From this, and following the discussion we had at the beginning of Subsection 3.4, according to arguments *e.g.* from Karoński et al. (2004), we have that $\chi_\Sigma(G) \leq 3$ holds for every nice cactus G. This is because every nice cactus

G admits a proper 3-labelling where adjacent vertices are essentially distinguished by their sums modulo 3. We tried to adapt such modulo methods to pushing schemes in cacti, to prove the optimal result that $\mathrm{P}^{\mathrm{l}}(G) \leq 2$ holds for every nice cactus G (which would indeed be best possible *e.g.* because of odd-length cycles, recall Theorem 3.4), and to confirm Conjecture 2.6 for cacti. Although we believe we managed to get a functional proof, and to prove that modulo methods can be applied to this context, said proof relied heavily on the use of computer programs to exhaustively check for the existence of particular pushing schemes, so heavily that a corresponding proof would not be legible. For this reason, we do not provide a formal proof, and instead raise the question of designing a readable, more straight proof of an optimal result for cacti.

- Regarding our complexity results from Section 4, one could first wonder whether our two NP-hardness results (regarding P^{l} and P^{t}) hold when restricted to particular graph classes. In particular, our proof of Theorem 4.2 does not work when restricted to bipartite graphs, since the M-gadget we introduced is a crucial tool that is not bipartite; thus, we wonder about the complexity of 1-PUSHABILITY when restricted to bipartite graphs, especially since the problem of deciding whether $\chi_\Sigma(G) \leq 2$ holds for a bipartite graph G lies in P, see Thomassen et al. (2016).

  Let us remark as well that, because our reductions in the proofs of Theorems 4.2 and 4.3 do not fulfill some properties, we do not get that these results hold for some restricted classes of graphs of interest. In particular, although 3-SATISFIABILITY remains NP-complete when restricted to instances where literals appear in a bounded number of clauses each (see *e.g.* Tovey (1984)), in our reductions the graphs we construct have their maximum degree being a function of the formula F; so, we do not get that any of Theorems 4.2 and 4.3 holds when restricted to graphs of bounded maximum degree. Likewise, 3-SATISFIABILITY remains NP-complete when restricted to planar formulas, see Lichtenstein (1982), but it turns out that our M-gadget from Figure 2 is not planar (to see this is true, note that we obtain a K_5 when identifying all of a, b, c, d to a single vertex). Thus, some efforts would be needed to prove that 1-PUSHABILITY and MINIMUM TOTAL PUSHABILITY remain NP-complete for planar graphs. More generally, we wonder whether there are other interesting classes of graphs for which the two problems are NP-complete, and whether they are polynomial-time solvable for others.

  We also expect k-PUSHABILITY to be NP-complete for all $k \geq 2$, and we would be curious to see a proof of that. In particular, we would be interested in generalisations of the M-gadget to larger values of k.

- Regarding the more global parameter P^{t}, we raised Observation 2.2 but had numerous occasions to see that the stated upper bound (function of P^{l}) is bad in general. We thus wonder about natural and tight bounds on P^{t}. Recall, as discussed by the end of Subsection 3.2, that there are graphs G for which $\mathrm{P}^{\mathrm{t}}(G)$ is a quadratic function of $|V(G)|$. However, in these graphs, $|V(G)|$ is very close to $\Delta(G)$, so we are not sure how this lower bound should be better interpreted.

From a more general perspective, we believe creating irregularity in graphs through pushing vertices is an interesting concept, and could be worth studying for other notions of graph irregularity. Recall indeed that, in the notion of proper pushing scheme we considered throughout, we focused on the notion of local irregularity, while other such notions of graph irregularity exist. For instance, one could wonder about pushing schemes encoding labellings where no two vertices (adjacent or not) have the same sum, thus encoding total irregularity (w.r.t. multigraphs in which no two vertices have the same degree). It is worth mentioning that the associated notion of irregular labelling have been studied for long, more than a decade prior the introduction of the 1-2-3 Conjecture, through the notion of irregularity strength of graphs from Alavi et al. (1987). While some of our results in this paper on the more local version of the problem generalise to this more global one, some others do not, and we thus believe this could all be worth investigating.

References

Y. Alavi, G. Chartrand, F. R. K. Chung, P. Erdös, R. L. Graham, and O. R. Oellermann. Highly irregular graphs. *Journal of Graph Theory*, 11(2):235–249, 1987. doi: 10.1002/jgt.3190110214. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190110214`.

O. Baudon, J. Bensmail, J. Przybyło, and M. Woźniak. On decomposing regular graphs into locally irregular subgraphs. *European Journal of Combinatorics*, 49:90–104, 2015. ISSN 0195-6698. doi:

10.1016/j.ejc.2015.02.031. URL `https://www.sciencedirect.com/science/article/pii/S0195669815000530`.

J. Bensmail. *A contribution to distinguishing labellings of graphs*. Habilitation à diriger des recherches, Université côte d'azur, 2020. URL `https://hal.science/tel-03081889`.

G. J. Chang, C. Lu, J. Wu, and Q. Yu. Vertex-coloring Edge-weightings of Graphs. *Taiwanese Journal of Mathematics*, 15(4):1807–1813, 2011. URL `https://doi.org/10.11650/twjm/1500406380`.

G. Chartrand, M. Jacobon, J. Lehel, O. Oellermann, S. Ruiz, and F. Saba. Irregular networks. *Congressus Numerantium*, 64:197–210, 1988.

A. Dudek and D. Wajc. On the complexity of vertex-coloring edge-weightings. *Discrete Mathematics & Theoretical Computer Science*, Vol. 13 no. 3, 2011. ISSN 1365–8050. doi: 10.46298/dmtcs.548. URL `https://dmtcs.episciences.org/548`.

J. Gallian. A dynamic survey of graph labeling. *Electronic Journal of Combinatorics, DS6*, 19, 2000. doi: 10.37236/11668.

M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, 91(1):151–157, 2004. ISSN 0095-8956. doi: 10.1016/j.jctb.2003.12.001. URL `https://www.sciencedirect.com/science/article/pii/S0095895603001412`.

R. Keusch. A solution to the 1-2-3 conjecture. *Journal of Combinatorial Theory, Series B*, 166:183–202, 2024. ISSN 0095-8956. doi: 10.1016/j.jctb.2024.01.002. URL `https://www.sciencedirect.com/science/article/pii/S0095895624000030`.

D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982. doi: 10.1137/0211025.

B. Seamone. The 1-2-3 conjecture and related problems: a survey, 2012. URL `https://arxiv.org/abs/1211.5122`.

C. Thomassen, Y. Wu, and 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. ISSN 0095-8956. doi: 10.1016/j.jctb.2016.06.010. URL `https://www.sciencedirect.com/science/article/pii/S009589561630051X`.

C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1): 85–89, 1984. ISSN 0166-218X. doi: 10.1016/0166-218X(84)90081-7. URL `https://www.sciencedirect.com/science/article/pii/0166218X84900817`.