



On the most imbalanced orientation of a graph

Walid Ben-Ameur, Antoine Glorieux, José Neto

► To cite this version:

Walid Ben-Ameur, Antoine Glorieux, José Neto. On the most imbalanced orientation of a graph. Journal of Combinatorial Optimization, 2018, 36 (2), pp.637-669. 10.1007/s10878-017-0117-1 . hal-01497902

HAL Id: hal-01497902

<https://hal.science/hal-01497902>

Submitted on 29 Mar 2017

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

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

On the most imbalanced orientation of a graph

Walid Ben-Ameur, Antoine Glorieux, and José Neto

SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay

9 Rue Charles Fourier, 91011 Evry Cedex, France

walid.benameur@telecom-sudparis.eu, antoine.glorieux@telecom-sudparis.eu,

jose.neto@telecom-sudparis.eu

Abstract. We study the problem of orienting the edges of a graph such that the minimum over all the vertices of the absolute difference between the outdegree and the indegree of a vertex is maximized. We call this minimum *the imbalance* of the orientation, i.e. the higher it gets, the more imbalanced the orientation is. The studied problem is denoted by MAXIM. We first characterize graphs for which the optimal objective value of MAXIM is zero. Next we show that MAXIM is generally NP-hard and cannot be approximated within a ratio of $\frac{1}{2} + \varepsilon$ for any constant $\varepsilon > 0$ in polynomial time unless $P = NP$ even if the minimum degree of the graph δ equals 2. Then we describe a polynomial-time approximation algorithm whose ratio is almost equal to $\frac{1}{2}$. An exact polynomial-time algorithm is also derived for cacti. Finally, two mixed integer linear programming formulations are presented. Several valid inequalities are exhibited with the related separation algorithms. The performance of the strengthened formulations is assessed through several numerical experiments.

1 Introduction and notation

Let $G = (V, E)$ be an undirected simple graph (with node set V and edge set E), we denote by δ_G the minimum degree of the vertices of G . An orientation Λ of G is an assignment of a direction to each undirected edge uv in E , i.e. any function on E of the form $\Lambda(uv) \in \{\vec{uv}, \vec{vu}\}$, $\forall \{uv\} \in E$ where $\vec{uv}$ and $\vec{vu}$ denote the edge uv oriented from u to v . For each vertex v of G we denote by $d_G(v)$ or $d(v)$ the unoriented degree of v in G and by $d_\Lambda^+(v)$ or $d^+(v)$ (resp. $d_\Lambda^-(v)$ or $d^-(v)$) the outdegree (resp. indegree) of v in G w.r.t. Λ . Graph orientation is a well studied area in graph theory and combinatorial optimization, a large variety of constrained orientations as well as objective functions have been considered so far.

Among those arise the popular degree-constrained orientation problems: in 1976, Frank & Gyárfás [14] gave a simple characterization of the existence of an orientation such that the outdegree of every vertex is between a lower and an upper bound given for each vertex. Asahiro et al. in [1,2,3] proved the NP-hardness of the weighted version of the problem where the maximum outdegree is minimized, gave some inapproximability results, and studied similar problems for different classes of graphs. Chrobak & Eppstein proved that for every planar graph a 3-bounded outdegree orientation and a 5-bounded outdegree acyclic orientation can be constructed in linear time [7].

Other problems involving other criteria on the orientation have been studied such as acyclicity, diameter or connectivity. Robbins' theorem (1939) for example states that

the graphs that have strong orientations are exactly the 2-edge-connected graphs [21] and later (1985), Chung et al. provided a linear time algorithm for checking whether a graph has such an orientation and finding one if it does [8]. Then in 1960, Nash-Williams generalized Robbin's theorem showing that an undirected graph has a k -arc-connected orientation if and only if it is $2k$ -edge-connected [20]. The problem called oriented diameter that consists in finding a strongly connected orientation with minimum diameter was introduced in 1978 by Chvátal & Thomassen: they proved that the problem is NP-hard for general graphs [9]. It was then proven to be NP-hard even if the graph is restricted to a subset of chordal graphs by Fomin et al. (2004) who gave also approximability and inapproximability results [12].

For an orientation Λ of $G = (V, E)$ and a vertex v we call $|d_\Lambda^+(v) - d_\Lambda^-(v)|$ the *imbalance* of v in G w.r.t Λ and we call $\min_{v \in V} |d_\Lambda^+(v) - d_\Lambda^-(v)|$ the *imbalance* of Λ . Biedl et al. studied the problem of finding an acyclic orientation of unweighted graphs minimizing the imbalance of each vertex: they proved that it is solvable in polynomial time for graphs with maximum degree at most three but NP-complete generally and for bipartite graphs with maximum degree six and gave a $\frac{13}{8}$ -approximation algorithm [6]. Then Kára et al. closed the gap proving the NP-completeness for graphs with maximum degree four. Furthermore, they proved that the problem remains NP-complete for planar graphs with maximum degree four and for 5-regular graphs [17].

Landau's famous theorem [18] gives a condition for a sequence of non-negative integers to be the score sequence or outdegree sequence of some tournament (i.e. oriented complete graph) and later, Harary & Moser characterized score sequences of strongly connected tournaments [15]. Analogous results for the “imbalance sequences” of directed graphs are given by Mubayi et al. [19]. In 1962, Ford & Fulkerson characterized the mixed graphs (i.e. partially oriented graphs) whose orientation can be completed in an eulerian orientation, that is to say, an orientation for which the imbalance of each vertex equals zero [13]. Many other results related to orientation have been proposed. Some of them are reviewed in [4].

Let us denote by $\vec{\mathcal{O}}(G)$ the set of all the orientations of G , we consider the problem of finding an orientation with maximized imbalance:

$$(\text{MAXIM}) \quad \text{MAXIM}(G) = \max_{\Lambda \in \vec{\mathcal{O}}(G)} \min_{v \in V} |d_\Lambda^+(v) - d_\Lambda^-(v)|$$

and we call $\text{MAXIM}(G)$ the value of MAXIM for G . The minimum degree δ_G of a graph G is a trivial upper bound for $\text{MAXIM}(G)$.

The rest of this paper is organized as follows. In the section 2, we give several characterizations of the the graphs verifying $\text{MAXIM}(G) = 0$. In Section 3, we show that MAXIM is generally NP-complete even for graphs with minimum degree 2 and inapproximable within a ratio $\frac{1}{2} + \varepsilon$ for any constant $\varepsilon > 0$ and then give an approximation algorithm whose ratio is almost equal to $\frac{1}{2}$. In Section 4, we present a polynomial-time exact algorithm for cactus. Section 5 is devoted to mixed integer linear programming formulations of MAXIM where families of valid inequalities are presented for the most promising formulation. These formulation have been implemented and the computational results are reported in Section 6.

Since the value of MAXIM for a graph is the minimum of the values of MAXIM on its connected components, from here on out, all the graphs we consider are assumed to

be connected. For a graph G and H a subgraph of G , we will use the notations $V(H)$ and $E(H)$ to refer to the set of vertices of G and the set of edges of H , respectively.

2 Characterizing the graphs for which $\text{MAXIM}(G) = 0$

Now we characterize the graphs verifying $\text{MAXIM}(G) = 0$. We start by unveiling several necessary conditions and properties of such graphs. First we can show that concerning such a graph, we can find an orientation satisfying several additional properties.

Proposition 1 *Let G be a graph such that $\text{MAXIM}(G) = 0$ and $u \in V$. Then there exists an orientation $\Lambda \in \vec{\mathcal{O}}(G)$ such that u is the only vertex of G with imbalance equal to zero w.r.t. Λ .*

Proof. Let $\Lambda \in \vec{\mathcal{O}}(G)$ be an orientation minimizing $|\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\}|$. We suppose that $|\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\}| \geq 2$. We choose two distinct vertices v and w in $\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\}$ and a path $p = (v = u_0, \dots, u_n = w)$ between v and w . If we switch the orientation of the edge u_0u_1 , then the imbalance of u_0 becomes positive and necessarily the imbalance of u_1 becomes zero otherwise the resulting orientation would contradict the minimality of Λ . Using the same reasoning, if we switch the orientation of all the edges $u_0u_1, \dots, u_{n-2}u_{n-1}$, we obtain an orientation where both u_{n-1} and u_n have an imbalance equal to zero while the imbalance is positive on all the vertices $u_0, \dots, u_{n-2}$ and unchanged on all other vertices. So now if we switch the orientation of the edge $u_{n-1}u_n$ as well, then the resulting orientation contradicts the minimality of Λ . Hence, $|\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\}| = 1$.

Now let v be this unique vertex of G such that $|d_\Lambda^+(v) - d_\Lambda^-(v)| = 0$. Let $u \neq v$ be an arbitrary vertex and let $p = (v = u_0, \dots, u_n = u)$ be a path between v and u . By switching the orientation of all the edges $u_0u_1, \dots, u_{n-2}u_{n-1}$, we obtain an orientation Λ' where u has an imbalance equal to zero while the imbalance is positive for u_0 and unchanged on all other vertices. $\square$

This yields the following necessary condition: if G is a graph such that $\text{MAXIM}(G) = 0$, then G is eulerian. For let $u \in V$, we know there exists $\Lambda \in \vec{\mathcal{O}}(G)$ such that $\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\} = \{u\}$. Then $d_\Lambda^+(u) = d_\Lambda^-(u)$, hence $d(u) = d_\Lambda^+(u) + d_\Lambda^-(u) = 2d_\Lambda^+(u)$ is even. The following lemma about eulerian graphs will be useful for the proof of our characterization.

Lemma 2 *If G is an eulerian graph, then there exists an elementary cycle (hereafter just called cycle) C of G such that $G - E(C)$ has at most one connected component that is not an isolated vertex.*

Proof. Since G is eulerian and connected, it can be decomposed into edge-disjoint cycles that we can order $C_1, \dots, C_n$ according to the following condition: $\cup_{k=1}^i C_k$ is connected, $\forall i \in \llbracket 1, n \rrbracket$. Then C_n is the cycle we are looking for. $\square$

Now let us define a certain family of graphs which will prove to be exactly the graphs for which the optimal objective value of MAXIM is zero. Intuitively they are the graphs for which every block is an odd cycle.

Definition 3 We define the class of graphs $\mathcal{C}^{odd}$ as follows: a simple graph G is in $\mathcal{C}^{odd}$ if there exists n odd cycles $C_1, \dots, C_n$ ($n \geq 1$) such that:

- $\cup_{i=1}^n C_i = G$,
 - $|V(\cup_{k=1}^{i-1} C_k) \cap V(C_i)| = 1, \forall i \in \llbracket 2, n \rrbracket$.
- (1)

Theorem 4 For any simple graph G , $\text{MAXIM}(G) = 0$ if and only if $G \in \mathcal{C}^{odd}$.

Proof. • $\Leftarrow$ We work by induction on the number n of cycles contained in the graph.

Nothing is required for these cycles except that they must be elementary. If $n = 1$, then our graph is an odd cycle which implies $\text{MAXIM}(G) = 0$. Let $n \geq 2$, we assume that all graphs of $\mathcal{C}^{odd}$ with $k \leq n-1$ cycles verify $\text{MAXIM}(G) = 0$. Let $G \in \mathcal{C}^{odd}$ with n cycles $C_1, \dots, C_n$ as in (1). Suppose there exists $\Lambda \in \vec{\mathcal{O}}(G)$ with strictly positive imbalance. Let us call $G' = \cup_{i=1}^{n-1} C_i$ the graph obtained from G after removing C_n and let us take a look at $\Lambda|_{G'}$: the orientation of the edges of G' obtained from Λ restricted to $E(G')$. As G' is a graph of $n-1$ cycles in $\mathcal{C}^{odd}$, our inductive hypothesis implies that we have a vertex $u \in V(G')$ such that $|d_{\Lambda|_{G'}}^+(u) - d_{\Lambda|_{G'}}^-(u)| = 0$. Necessarily, $u = V(G') \cap V(C_n)$. Thus $|d_{\Lambda}^+(u) - d_{\Lambda}^-(u)| = |d_{\Lambda|_{C_n}}^+(u) - d_{\Lambda|_{C_n}}^-(u)| > 0$ implying that $\text{MAXIM}(C_n) > 0$, a contradiction because C_n is an odd cycle.

- $\Rightarrow$ Since $\text{MAXIM}(G) = 0$, we know that G is eulerian. We work again by induction on the number of elementary cycles n . If $n = 1$, then our graph is eulerian with a unique cycle, hence it is a cycle. Now as $\text{MAXIM}(G) = 0$, necessarily it is an odd cycle and is therefore in $\mathcal{C}^{odd}$. Let $n \geq 2$, we assume that all graphs with $k \leq n-1$ cycles verifying $\text{MAXIM}(G) = 0$ are in $\mathcal{C}^{odd}$. Let G be a graph with n cycles such that $\text{MAXIM}(G) = 0$. Thanks to Lemma 2, there exists a cycle C of G such that $G - E(C)$ has at most one connected component G' that is not an isolated vertex. Suppose that $\text{MAXIM}(G') > 0$, let $\Lambda \in \vec{\mathcal{O}}(G')$ with strictly positive imbalance. Let $u_0 \in V(G') \cap V(C)$, we name the vertices of C as follows: $u_0, u_1, \dots, u_k = u_0$. Without loss of generality, we can assume that $d_{\Lambda}^+(u_0) - d_{\Lambda}^-(u_0) > 0$; if it was not the case, replace Λ by its reverse. We complete Λ in an orientation of G by orienting the edges of C : we orient $u_0 u_1$ from u_0 to u_1 and go on as follows:

$$\forall i \in \llbracket 1, k-1 \rrbracket, \begin{cases} \text{if } u_i \in V(G'), & \text{we orient } u_i u_{i+1} \text{ as } u_{i-1} u_i, \\ \text{otherwise,} & \text{we orient } u_i u_{i+1} \text{ as } u_i u_{i-1}. \end{cases}$$

Where orienting an edge ab as another edge cd means orienting it from a to b if cd was oriented from c to d and from b to a otherwise. Let us have a look at the resulting orientation Λ' (cf Fig. 1): when completing Λ in Λ' , the imbalance of the vertices in $V(G') \setminus \{u_0\}$ was left unchanged, the imbalance of the vertices in $V(C) \setminus V(G')$ equals 2 and the imbalance of u_0 was either left unchanged or augmented by two. Hence Λ' has strictly positive imbalance which contradicts $\text{MAXIM}(G) = 0$, therefore, $\text{MAXIM}(G') = 0$.

Suppose $|V(G') \cap V(C)| \geq 2$ and let u and v be 2 distinct vertices in $V(G') \cap V(C)$

such that $u \neq v$. Thanks to Proposition 1, we know that there exists an orientation $\Lambda \in \vec{\mathcal{O}}(G')$ such that $\{w \in V / |d_\Lambda^+(w) - d_\Lambda^-(w)| = 0\} = \{v\}$ and without loss of generality, $d_\Lambda^+(u) - d_\Lambda^-(u) > 0$. We name the vertices of C as follows: $u = u_0 u_1 \cdots u_k = u_0$, $v = u_l$ and we complete Λ in an orientation of G by orienting the edges of C : we orient $u_0 u_1$ from u_0 and u_1 and go on as follows:

$$\forall i \in \llbracket 1, k-1 \rrbracket \setminus \{l\}, \begin{cases} \text{if } u_i \in V(G'), & \text{we orient } u_i u_{i+1} \text{ as } u_{i-1} u_i, \\ \text{otherwise,} & \text{we orient } u_i u_{i+1} \text{ as } u_i u_{i-1}. \end{cases}$$

And we orient $u_l u_{l+1}$ as $u_l u_{l-1}$. In the resulting orientation Λ' , the imbalance of the vertices in $V(G') \setminus \{u, v\}$ was left unchanged, the imbalance of the vertices in $V(C) \setminus V(G')$ equals 2, the imbalance of v was augmented by two and the imbalance of u was either left unchanged or augmented by two. Hence Λ' contradicts $\text{MAXIM}(G) = 0$, therefore, $|V(G') \cap V(C)| = 1$.

Suppose C is even. We call $u \in V(G')$ such that $V(G') \cap V(C) = \{u\}$, and $\Lambda \in \vec{\mathcal{O}}(G')$ such that $\{v \in V / |d_\Lambda^+(v) - d_\Lambda^-(v)| = 0\} = \{u\}$. We name the vertices of C as follows: $u = u_0 u_1 \cdots u_k = u_0$ and we complete Λ in an orientation of G by orienting the edges of C : we orient $u_0 u_1$ from u_0 to u_1 and $u_i u_{i+1}$ as $u_i u_{i-1}$, $\forall i \in \llbracket 1, k-1 \rrbracket$. In the resulting orientation Λ' , the imbalance of the vertices in $V(G') \setminus \{u\}$ was left unchanged, the imbalance of the vertices in $V(C) \setminus V(G')$ equals 2 and, C being even, the imbalance of u was augmented by two. Hence Λ' contradicts $\text{MAXIM}(G) = 0$, therefore, C is odd.

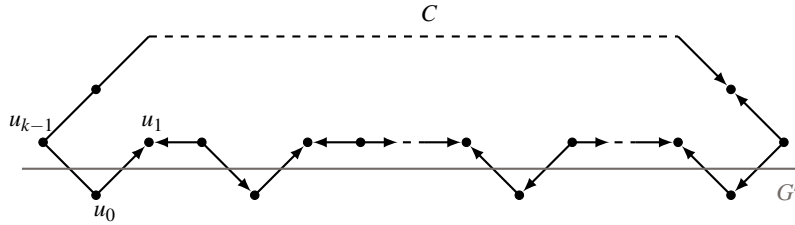
As G' is a graph with at most $n-1$ cycles verifying $\text{MAXIM}(G) = 0$, by the induction hypothesis, there exist $C_1, \dots, C_{n-1}$ odd cycles such that:

- $\cup_{i=1}^{n-1} C_i = G'$,
- $|V(\cup_{k=1}^{i-1} C_k) \cap V(C_i)| = 1, \forall i \in \llbracket 2, n-1 \rrbracket$.

Adding the odd cycle $C_n = C$, we obtain that $G \in \mathcal{C}^{odd}$.

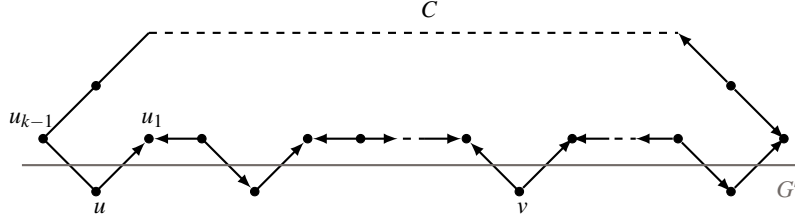
□

Fig. 1: The vertices of C in G' are left unchanged imbalance-wise, the other vertices of C are set to 2 and in the end $|d_{\Lambda'}^+(u_0) - d_{\Lambda'}^-(u_0)| \geq |d_\Lambda^+(u_0) - d_\Lambda^-(u_0)| > 0$



Now in order to widen our perception of those graphs, let us show another characterization.

Fig. 2: The vertices of C in G' are left unchanged imbalance-wise except for v which is set to 2, like the other vertices of C and in the end $|d_{\Lambda'}^+(u_0) - d_{\Lambda'}^-(u_0)| \geq |d_{\Lambda}^+(u_0) - d_{\Lambda}^-(u_0)| > 0$



Theorem 5 For every simple graph G ,

$$G \in \mathcal{C}^{odd} \Leftrightarrow G \text{ is eulerian with no even cycle}$$

Proof. • $\Rightarrow$ By construction, every graph in $\mathcal{C}^{odd}$ is eulerian with no even cycle.

- $\Leftarrow$ We will once again work by induction on the number of cycles n .

If $n = 1$, then our graph is eulerian with a unique odd cycle, hence it is an odd cycle and is therefore in $\mathcal{C}^{odd}$.

Let $n \geq 2$, we assume that all eulerian graphs with no even cycle and $k \leq n - 1$ odd cycles are in $\mathcal{C}^{odd}$. Let G be a graph with no even cycle and n odd cycles. Thanks to Lemma 2, there exists an odd cycle C of G such that $G - E(C)$ has only one connected component G' that is not an isolated vertex. As G' is eulerian and even-cycle-free with $n - 1$ odd cycles, by induction hypothesis, $G' \in \mathcal{C}^{odd}$, hence there exist $C_1, \dots, C_{n-1}$ odd cycles such that:

- $\cup_{i=1}^{n-1} C_i = G'$,
- $|V(\cup_{k=1}^{i-1} C_k) \cap V(C_i)| = 1, \forall i \in \llbracket 2, n-1 \rrbracket$.

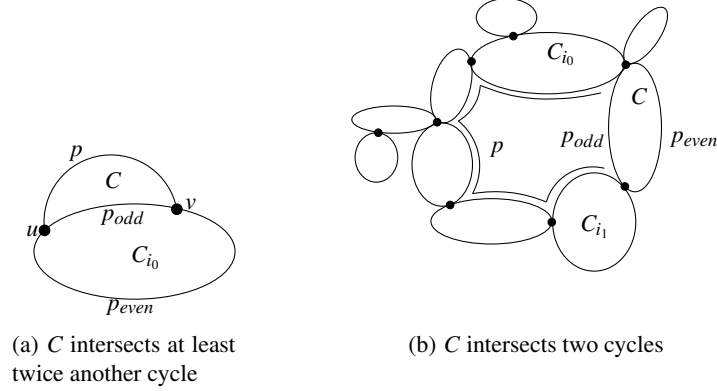
Suppose there exist u and v ($u \neq v$) belonging to $V(\cup_{k=1}^{n-1} C_k) \cap V(C)$. Since G' is connected, let p be an elementary path in G' between u and v . We can assume that u and v are the only vertices of C contained in p , otherwise we could replace v by the first vertex of C encountered when traveling on p from u . C defines two other vertex-disjoint paths between u and v : one even that we will call p_{even} and one odd that we will call p_{odd} . p being vertex disjoint with either p_{even} or p_{odd} , by concatenating it with the one corresponding to its parity, we obtain an even cycle of G , contradicting our hypothesis on G (cf. Fig. 3(a) and 3(b)). This yields that $|V(C) \cap V(G')| = 1$. From that we can conclude

- $\cup_{i=1}^n C_i = G$,
- $|V(\cup_{k=1}^{i-1} C_k) \cap V(C_i)| = 1, \forall i \in \llbracket 2, n \rrbracket$.

Hence $G \in \mathcal{C}^{odd}$.

□

Fig. 3: In both cases, concatenating p with p_{even} or p_{odd} yields an even cycle in G



3 Complexity and (in)approximability

In this section we prove the NP-completeness and inapproximability of our problem and give an approximation algorithm based on the special case of bipartite graphs.

Concerning the complexity of MAXIM, we show that the problem is NP-complete. More precisely, that answering if $\text{MAXIM}(G)$ equals 2 for a graph G such that $\delta_G = 2$ is NP-complete. For that purpose, we introduce a variant of the satisfiability problem that we reduce to a MAXIM instance: the not-all-equal at most 3-SAT(3V).

Not-all-equal at most 3-SAT(3V) is a restriction of not-all-equal at most 3-SAT which is itself a restriction of 3-SAT known to be NP-complete [22], where each clause contains at most three literals and in each clause, not all the literals can be true. Since 2-SAT can be solved in polynomial time, we hereafter deal only with formulas having at least one three-literals clause. The added restriction of not-all-equal at most 3-SAT(3V) is that each variable (not literal) appears at most three times in a formula. The resulting problem is still NP-complete.

Lemma 6 *The not-all-equal at most 3-SAT(3V) problem is NP-complete.*

Proof. See Appendix A.

Now we associate to a not-all-equal at most 3-SAT(3V) instance φ with n variables $\{x_1, \dots, x_n\}$ and m clauses $\{c_1, \dots, c_m\}$ a graph G_φ for which the value w.r.t. MAXIM will give the answer to whether φ is satisfiable or not. If a variable x_i occurs only in positive literals (resp. only in negative literals), it follows that a satisfying assignment of the variables of φ must necessarily give the value TRUE (resp. FALSE) to x_i , therefore x_i can be removed from φ with conservation of the satisfiability. Thus, without loss of generality, we can assume that in any not-all-equal at most 3-SAT(3V) formula, every variable occurs at least once as a positive literal and at least once as a negative literal. G_φ consists of gadgets that mimic the variables and the clauses of φ and additional edges that connect them together:

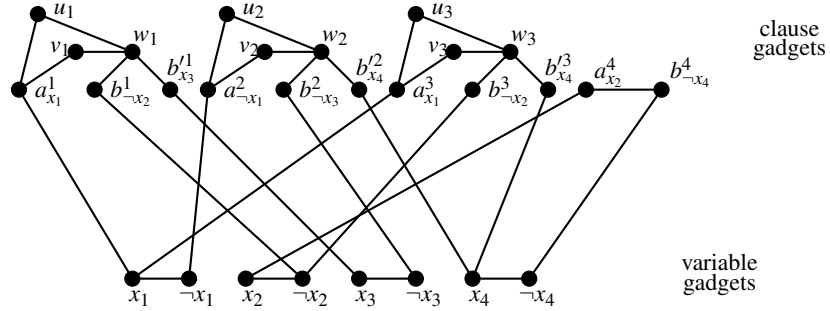
- the gadget corresponding to a variable x_i consists of two vertices labeled x_i and $\neg x_i$ and one edge connecting them;
- the gadget corresponding to a two-literals clause $c_j = (l^1 \vee l^2)$, where l^1 and l^2 are its literals, consists in two vertices labeled $a_{l^1}^j$ and $b_{l^2}^j$ corresponding to l^1 and l^2 respectively (the index " l^k " of the vertices labels stands for the literal it represents, i.e. x_i if l^k is the variable x_i and $\neg x_i$ if l^k is the negation of the variable x_i) and one edge connecting them;
- the gadget corresponding to a three-literals clause gadget consists in six vertices and six edges. For a clause $c_j = (l^1 \vee l^2 \vee l^3)$, where l^1, l^2 and l^3 are its literals (the order is arbitrary), three vertices labeled $a_{l^1}^j, b_{l^2}^j$ and $b_{l^3}^j$ correspond to l^1, l^2 and l^3 respectively. Three additional vertices are labeled u^j, v^j and w^j and the gadgets' edges are $a_{l^1}^j u^j, a_{l^1}^j v^j, u^j w^j, v^j w^j, w^j b_{l^2}^j$ and $w^j b_{l^3}^j$;
- $\forall i \in \llbracket 1, n \rrbracket$, the vertex labeled x_i (resp. $\neg x_i$) is connected to all the vertices labeled $a_{x_i}^j, b_{x_i}^j$ or $b_{\neg x_i}^j$ (resp. $a_{\neg x_i}^j, b_{\neg x_i}^j$ or $b_{x_i}^j$), $\forall j \in \llbracket 1, m \rrbracket$.

As an example, for a formula

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_4), \quad (2)$$

the corresponding graph G_φ is represented in Fig. 4.

Fig. 4: G_φ for $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_4)$



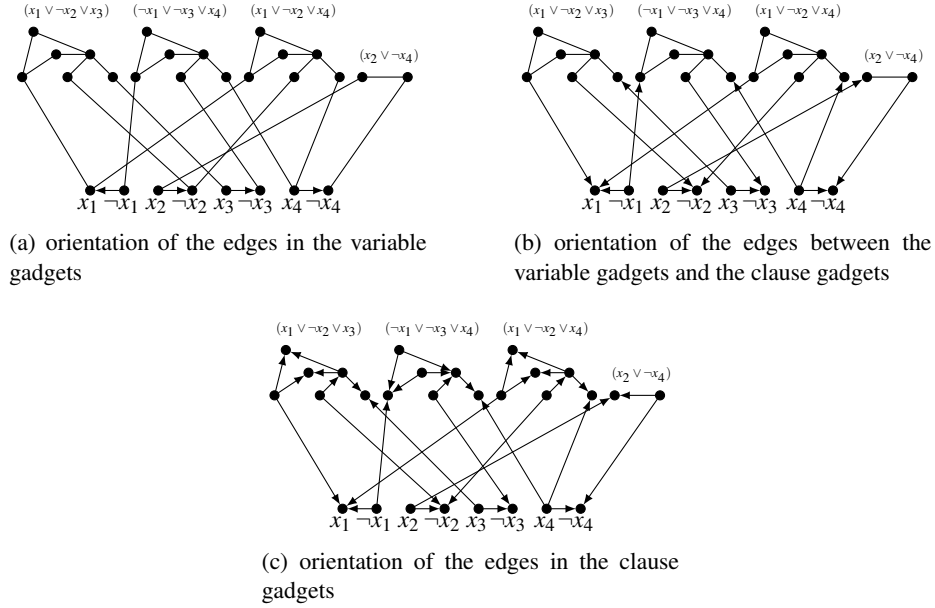
Theorem 7 A not-all-equal at most 3-SAT(3V) formula φ is satisfiable if and only if $\text{MAXIM}(G_\varphi) = 2$.

Proof. • $\Rightarrow$ Suppose φ is satisfiable and let $v : \{x_1, \dots, x_n\} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ be a satisfying assignment of $x_1, \dots, x_n$. We know that $\delta_{G_\varphi} = 2$ which yields $\text{MAXIM}(G_\varphi)$

≤ 2 . So let us build an orientation $\Lambda \in \vec{\mathcal{O}}(G_\varphi)$ for which the imbalance is greater than or equal to 2. First, we assign an orientation to the edges of the variable gadget:

$$\Lambda(x_i \neg x_i) = \begin{cases} \overrightarrow{x_i \neg x_i} & \text{if } v(x_i) = \text{TRUE}; \\ \overrightarrow{\neg x_i x_i} & \text{otherwise.} \end{cases}$$

Fig. 5: G_φ corresponding to $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_4)$ satisfied by $v(x_1, x_2, x_3, x_4) = (\text{FALSE}, \text{TRUE}, \text{TRUE}, \text{TRUE})$.



For example, for the formula $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_4)$ satisfied by the assignment $v(x_1, x_2, x_3, x_4) = (\text{FALSE}, \text{TRUE}, \text{TRUE}, \text{TRUE})$, the edges of the variable gadgets of graph G_φ are oriented as in figure 5(a). Since each variable x_i occurs at least once as a positive literal and at least once as a negative literal, $2 \leq d_{G_\varphi}(x_i) \leq 3$ and $2 \leq d_{G_\varphi}(\neg x_i) \leq 3$, $\forall i \in \llbracket 1, n \rrbracket$. Then to ensure our objective on the imbalance of Λ , the orientation of the edges connecting vertex gadgets and clause gadgets must be such that $\forall i \in \llbracket 1, n \rrbracket$, $|d_\Lambda^+(x_i) - d_\Lambda^-(x_i)| = d_{G_\varphi}(x_i)$ and $|d_\Lambda^+(\neg x_i) - d_\Lambda^-(\neg x_i)| = d_{G_\varphi}(\neg x_i)$. In other words, for $i \in \llbracket 1, n \rrbracket$, if $v(x_i) = \text{TRUE}$ (resp. $v(x_i) = \text{FALSE}$), then the edges adjacent to the vertex x_i are oriented from x_i (resp. to x_i) and the edges adjacent to the vertex $\neg x_i$ are oriented to $\neg x_i$ (resp. from $\neg x_i$), e.g. Fig. 5(b).

So far, all the edges in the variables gadgets and the edges connecting the vertex gadgets and the clause gadgets have been oriented and the vertices in the variables

gadgets have imbalance greater than or equal to 2. In order to complete our orientation Λ we have to orient the edges in the clause gadgets.

Let $c_j = (l^1 \vee l^2)$ be a two-literals clause. Since v satisfies φ , we know that exactly one of the two literals is true w.r.t. v . Which, according to the way we oriented edges so far, means that exactly one of $a_{l^1}^j$ and $b_{l^2}^j$ has one incoming arc from a variable gadget and the other has one outgoing arc to a variable gadget. If $a_{l^1}^j$ is the one with the incoming arc from a variable gadget (meaning that $v(l^1) = \text{TRUE}$), then we assign $\Lambda(a_{l^1}^j b_{l^2}^j) = \overrightarrow{b_{l^2}^j a_{l^1}^j}$, otherwise the opposite. We obtain $|d_\Lambda^+(a_{l^1}^j) - d_\Lambda^-(a_{l^1}^j)| = |d_\Lambda^+(b_{l^2}^j) - d_\Lambda^-(b_{l^2}^j)| = 2$.

Let $c_j = (l^1 \vee l^2 \vee l^3)$ (the order is identical to the one chosen to build the clause gadget, i.e. $d_{G_\varphi}(a_{l^1}^j) = 3$ and $d_{G_\varphi}(b_{l^2}^j) = d_{G_\varphi}(b_{l^3}^{jj}) = 2$) be at three-literals clause. If the edge connecting $a_{l^1}^j$ to a variable gadget is oriented to $a_{l^1}^j$ (meaning that $v(l^1) = \text{TRUE}$), then we assign $\Lambda(a_{l^1}^j u_j) = \overrightarrow{u_j a_{l^1}^j}$, $\Lambda(a_{l^1}^j v_j) = \overrightarrow{v_j a_{l^1}^j}$, $\Lambda(u_j w_j) = \overrightarrow{u_j w_j}$ and $\Lambda(v_j w_j) = \overrightarrow{v_j w_j}$. Since $v(l^1) = \text{TRUE}$, either both $v(l^2)$ and $v(l^3)$ are FALSE or exactly one of $v(l^2)$ and $v(l^3)$ is TRUE and one is FALSE. If both are FALSE then $b_{l^2}^j$ and $b_{l^3}^{jj}$ have an outgoing arc to a variable gadget. In that case, we orient $w_j b_{l^2}^j$ and $w_j b_{l^3}^{jj}$ to w_j and we obtain $|d_\Lambda^+(a_{l^1}^j) - d_\Lambda^-(a_{l^1}^j)| = 3$, $|d_\Lambda^+(b_{l^2}^j) - d_\Lambda^-(b_{l^2}^j)| = |d_\Lambda^+(b_{l^3}^{jj}) - d_\Lambda^-(b_{l^3}^{jj})| = |d_\Lambda^+(u_j) - d_\Lambda^-(u_j)| = |d_\Lambda^+(v_j) - d_\Lambda^-(v_j)| = 2$ and $|d_\Lambda^+(w_j) - d_\Lambda^-(w_j)| = 4$. If exactly one of $v(l^2)$ and $v(l^3)$ is TRUE and one is FALSE, then exactly one of $b_{l^2}^j$ and $b_{l^3}^{jj}$ has an incoming arc from a variable gadget and the other an outgoing arc to a variable gadget. If $b_{l^2}^j$ is the one with the incoming arc from a variable gadget (meanings that $v(l^2) = \text{TRUE}$ and $v(l^3) = \text{FALSE}$), then we assign $\Lambda(w_j b_{l^2}^j) = \overrightarrow{w_j b_{l^2}^j}$ and $\Lambda(w_j b_{l^3}^{jj}) = \overrightarrow{b_{l^3}^{jj} w_j}$, otherwise the opposite. We obtain $|d_\Lambda^+(a_{l^1}^j) - d_\Lambda^-(a_{l^1}^j)| = 3$ and $|d_\Lambda^+(b_{l^2}^j) - d_\Lambda^-(b_{l^2}^j)| = |d_\Lambda^+(b_{l^3}^{jj}) - d_\Lambda^-(b_{l^3}^{jj})| = |d_\Lambda^+(u_j) - d_\Lambda^-(u_j)| = |d_\Lambda^+(v_j) - d_\Lambda^-(v_j)| = |d_\Lambda^+(w_j) - d_\Lambda^-(w_j)| = 2$.

If, on the other hand, the edge connecting $a_{l^1}^j$ to a variable gadget is oriented from $a_{l^1}^j$ (meanings that $v(l^1) = \text{FALSE}$), then we assign $\Lambda(a_{l^1}^j u_j) = \overrightarrow{a_{l^1}^j u_j}$, $\Lambda(a_{l^1}^j v_j) = \overrightarrow{a_{l^1}^j v_j}$, $\Lambda(u_j w_j) = \overrightarrow{w_j u_j}$ and $\Lambda(v_j w_j) = \overrightarrow{w_j v_j}$. By symmetry, we conclude in the same way that $|d_\Lambda^+(a_{l^1}^j) - d_\Lambda^-(a_{l^1}^j)| = 3$ and $|d_\Lambda^+(b_{l^2}^j) - d_\Lambda^-(b_{l^2}^j)| = |d_\Lambda^+(b_{l^3}^{jj}) - d_\Lambda^-(b_{l^3}^{jj})| = |d_\Lambda^+(u_j) - d_\Lambda^-(u_j)| = |d_\Lambda^+(v_j) - d_\Lambda^-(v_j)| = |d_\Lambda^+(w_j) - d_\Lambda^-(w_j)| = 2$.

Consequently, the imbalance of the resulting orientation Λ is greater than or equal to 2, e.g. Fig. 5(c).

- $\Leftarrow$ Now we assume that $\text{MAXIM}(G_\varphi) = 2$, let $\Lambda \in \vec{\mathcal{O}}(G_\varphi)$ with optimal imbalance. Since all the vertices in the variable gadgets have degree at most 3, each vertex x_i (or $\neg x_i$) is necessarily adjacent to only incoming arcs or only outgoing arcs w.r.t. Λ . We will show that the assignment $v : \{x_1, \dots, x_n\} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ of $x_1, \dots, x_n$ defined by

$$v(x_i) = \begin{cases} \text{TRUE} & \text{if } d_\Lambda^+(x_i) > d_\Lambda^-(x_i); \\ \text{FALSE} & \text{otherwise;} \end{cases}$$

satisfies φ . Suppose φ does not satisfy a clause c_j , $j \in \llbracket 1, m \rrbracket$. If c_j is a two-literals clause $(l^1 \vee l^2)$ then either $v(l^1) = v(l^2) = \text{TRUE}$ or $v(l^1) = v(l^2) = \text{FALSE}$, i.e. either both $a_{l^1}^j$ and $b_{l^2}^j$ have an incoming arc from a variable gadget or both have an outgoing arc to a variable gadget and in both cases, whichever is the orientation assigned to $a_{l^1}^j b_{l^2}^j$ by Λ , either $a_{l^1}^j$ or $b_{l^2}^j$ has a zero imbalance which contradicts our assumption. So c_j is a three-literals clause $(l^1 \vee l^2 \vee l^3)$ (the order is identical to the one chosen to build the clause gadget, i.e. $d_{G_\varphi}(a_{l^1}^j) = 3$ and $d_{G_\varphi}(b_{l^2}^j) = d_{G_\varphi}(b_{l^3}^j) = 2$). Then either $v(l^1) = v(l^2) = v(l^3) = \text{TRUE}$ or $v(l^1) = v(l^2) = v(l^3) = \text{FALSE}$, i.e. either all $a_{l^1}^j$, $b_{l^2}^j$ and $b_{l^3}^j$ have an incoming arc from a variable gadget or they all have an outgoing arc to a variable gadget. In the first case, it implies $\Lambda(a_{l^1}^j u_j) = \overrightarrow{u_j a_{l^1}^j}$, $\Lambda(a_{l^1}^j v_j) = \overrightarrow{v_j a_{l^1}^j}$, $\Lambda(u_j w_j) = \overrightarrow{u_j w_j}$, $\Lambda(v_j w_j) = \overrightarrow{v_j w_j}$, $\Lambda(w_j b_{l^2}^j) = \overrightarrow{w_j b_{l^2}^j}$ and $\Lambda(w_j b_{l^3}^j) = \overrightarrow{w_j b_{l^3}^j}$, and we obtain $|d_\Lambda^+(w_j) - d_\Lambda^-(w_j)| = 0$ which contradicts the optimality of Λ . Similarly, in the second case it implies that the orientations assigned to the edges of the clause gadgets are the opposite from the previous ones and we obtain the same contradiction. So we can conclude that v does satisfy φ . $\square$

Corollary 8 *MAXIM is NP-complete and inapproximable within $\frac{1}{2} + \varepsilon$ where $\varepsilon \in \mathbb{R}_+^*$, unless $P = NP$.*

Proof. Let $\varepsilon \in \mathbb{R}_+^*$, suppose that there exists a polynomial approximation algorithm giving $val \geq (\frac{1}{2} + \varepsilon) \text{MAXIM}(G)$ for an input graph G . Let φ be a not-all-equal at most 3-SAT(3V) formula and G_φ its associated graph. Since G_φ contains at least one three-literals clause gadget, we know that G_φ contains an even cycle and $\delta_{G_\varphi} = 2$. This leads to $\text{MAXIM}(G_\varphi) \in \{1, 2\}$ and since $(\frac{1}{2} + \varepsilon) \text{MAXIM}(G_\varphi) \leq val \leq \text{MAXIM}(G_\varphi)$, if the polynomial approximation algorithm returns a value less than or equal to 1 then

$$(\frac{1}{2} + \varepsilon) \text{MAXIM}(G_\varphi) \leq 1 \Rightarrow \text{MAXIM}(G_\varphi) < 2 \Rightarrow \text{MAXIM}(G_\varphi) = 1;$$

and if it returns a value greater than 1, then $\text{MAXIM}(G_\varphi)$ is greater than 1 hence equal to 2. In other words the polynomial approximation algorithm output answers whether φ is satisfiable or not which implies $P = NP$. $\square$

Let us use $\text{cut}(A)$ to denote the set of edges having only one endpoint in A . Now we consider the case of bipartite graphs: if $G = (V_1 \cup V_2, E)$ is a bipartite graph, the orientation that consists in assigning to each edge in E the orientation from its endpoint in V_1 to its endpoint in V_2 has an imbalance equal to δ_G , i.e. optimal. This simple case permits us to obtain the following lower bound:

Theorem 9 *For every graph G ,*

$$\text{MAXIM}(G) \geq \lceil \frac{\delta_G}{2} \rceil - 1.$$

Proof. Let (V_1, V_2) be a partition of V corresponding to a cut $C = \text{cut}(V_1) \subset E$ such that we have $|\text{cut}(\{v\}) \cap C| \geq \lceil \frac{d(v)}{2} \rceil$, $\forall v \in V$. Such a cut exists: for example a maximum cardinality cut verifies this property, otherwise we could find a higher cardinality cut by switching a vertex $v \in V$ s.t. $|\text{cut}(\{v\}) \cap C| < \lceil \frac{d(v)}{2} \rceil$ from V_1 to V_2 (or the contrary). Moreover, if we iterated this process starting from a random k , we would converge in polynomial time to such a cut. Now we define $\Lambda \in \vec{\mathcal{O}}(G)$ as follows. We begin by orienting all edges in C from V_1 to V_2 . Then for any $i \in \{1, 2\}$, we orient the edges of the induced subgraph $G[V_i]$. We add a new vertex v_0 and an edge between v_0 and each vertex with an odd degree in $G[V_i]$ if it is not eulerian and we consider a decomposition of its edges into edge-disjoint cycles. We orient each of these cycles as a directed cycle. Removing v_0 if necessary, the imbalance of each vertex in $G[V_i]$ is now in $\{-1, 0, 1\}$ which implies that $\forall v \in V$ we have $|d_\Lambda^+(v) - d_\Lambda^-(v)| \geq \lceil \frac{d(v)}{2} \rceil - 1$, hence, $\text{MAXIM}(G) \geq \lceil \frac{\delta_G}{2} \rceil - 1$. $\square$

From the proof proposed above, it is easy to see that when $\delta_G \equiv 0[4]$ then $\text{MAXIM}(G) \geq \frac{\delta_G}{2}$ while $\text{MAXIM}(G) \geq \frac{\delta_G - 1}{2}$ when δ_G is odd and $\text{MAXIM}(G) \geq \frac{\delta_G}{2} - 1$ when $\delta_G \equiv 2[4]$. This leads to an approximation algorithm whose ratio is $\frac{1}{2}$ (resp. $\frac{1}{2} - \frac{1}{2\delta}$, $\frac{1}{2} - \frac{1}{\delta}$) when $\delta_G \equiv 0[4]$, (resp. δ_G is odd, $\delta_G \equiv 2[4]$).

4 Exact algorithm for cacti

The class of graphs defined in the previous section is a special case of cacti. A *cactus* is a connected simple graph for which every edge belongs to at most one simple cycle. Equivalently, a cactus is a connected graph for which every block, or maximal subgraph with no cut-vertex, is a single edge or a cycle. We will start by recalling the structure induced by the blocks of a graph.

Definition 10 Let $G = (V, E)$ be a graph, V_B the set of its blocks and V_C the set of its cutvertices. The block tree of G is the graph $T = (V_B \cup V_C, E')$ where $E' = \{Zu | Z \in V_B, u \in V_C \text{ and } u \in V(Z)\}$.

Observe that the block tree of a connected graph is a tree [11].

In Figure 6 we have an example of a cactus graph G (a), the isolation of its blocks (b) and finally its block tree T (c).

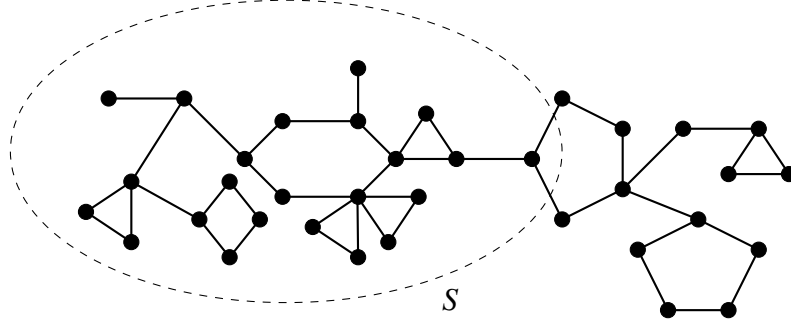
The next lemma is related to the minimum degree of a cactus.

Lemma 11 Let G be a cactus graph,

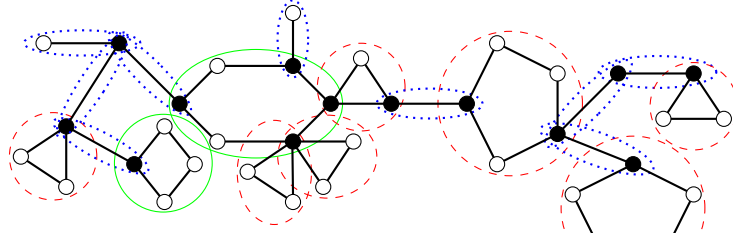
$$\delta_G \leq 2.$$

Proof. Let G be a cactus graph and T its block tree. Let $u \in V(T)$ be a leaf of T . By the definition of a cut vertex, u corresponds necessarily to a block of G . If u corresponds to a bridge of G , then one of its endpoint (the one that is not a cutvertex) is a leaf of G which implies $\delta_G = 1$. If u corresponds to a cycle of G , being a leaf of T , it contains only one cutvertex of G and therefore the degree of its other vertices equals 2. $\square$

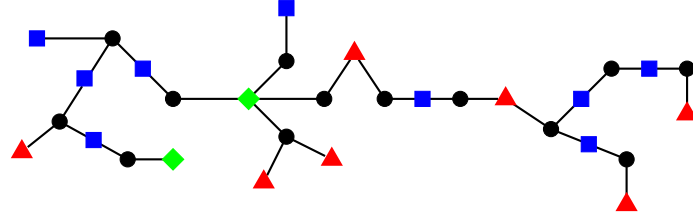
Fig. 6: Building the block tree



(a) G a cactus and S .



(b) Its blocks: even cycles (—), odd cycles (---) and bridges (·····).



(c) And the block tree T : cut vertices (●), even cycles (◆), odd cycles (▲) and bridges (■)

The previous lemma automatically provides an upper bound for $\text{MAXIM}(G)$.

Proposition 12 *Let G be a cactus,*

$$\text{MAXIM}(G) \leq 2.$$

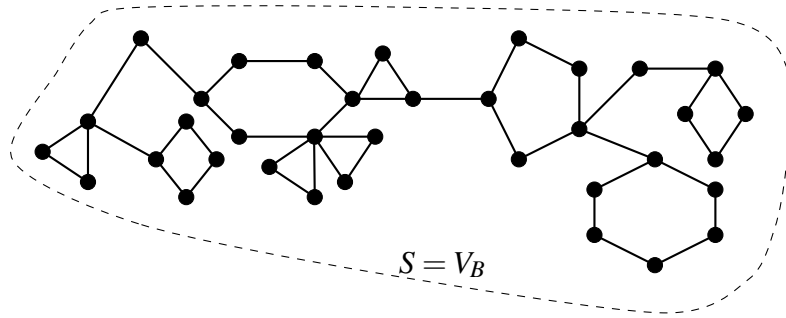
Note that an edge of a cactus is a single edge block if and only if it is a bridge. We have already characterized which of the cacti verify $\text{MAXIM}(G) = 0$, and the previous result states that the only two other possibilities are 1 or 2. The following will permit us to determine which. Let $G = (V, E)$ be a cactus and $T = (V_B \cup V_C, E')$ its block tree,

a subset $A \subseteq V_B$ will hereafter refer to a subset of vertices of T as well as the subgraph consisting in the union of the blocks of G corresponding to these vertices. We define for a cactus G the smallest subset $S \subseteq V_B$ containing all even cycles of G in addition to all vertices of T satisfying one of the following conditions:

- If $Y \in V_B$ corresponds to an odd cycle of G that shares a cut-vertex with a cycle or with two bridges contained in S , then $Y \in S$,
- if $Y \in V_B$ corresponds to a bridge of G that shares a cut-vertex with a block contained in S , then $Y \in S$.

In Figures 6(a) and 7 are examples of two cacti along with their subsets S .

Fig. 7: A graph G such that $\delta_G = 2$ and $S = V_B$.



Theorem 13 *Let G be a cactus,*

$$\text{MAXIM}(G) = 2 \iff S = V_B \text{ and } \delta_G = 2.$$

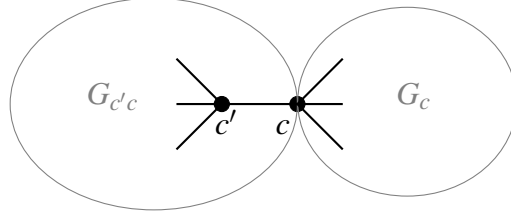
The idea of this result and its proof is that if we have a cactus G , we consider the subgraph S_0 of G consisting in the union of all the even cycles of G and that can be oriented with imbalance at least 2. Then we can extend the orientation of S_0 into an orientation of S , keeping an imbalance of at least two for each vertex with at least two oriented edges. Now if $\delta_G = 2$ and $S = V_B$, we obtain an orientation of G with imbalance at least 2. Moreover, for a cactus G , the subset S can be computed in polynomial time by a basic search on the block tree of G which means that we can derive a polynomial time algorithm solving MAXIM for any input cactus.

In order to prove Theorem 13, we first show two lemmas that will be useful in the proof of our theorem.

Lemma 14 *Let $G = (V, E)$ be a graph, $cc' \in E$ a bridge of G . Let us call G_c the connected component of $G \setminus cc'$ containing c and $G_{c'}$ the graph obtained by adding c and c' to the connected component of $G \setminus cc'$ containing c' (see Fig. 8). Then*

$$\text{MAXIM}(G) \leq \max_{\Lambda \in \vec{\mathcal{O}}(G_c)} \min \left(\min_{v \in V(G_c) \setminus \{c\}} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, |d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| + 1 \right).$$

Fig. 8: A graph G and a bridge cc' .



Proof. Let $\Lambda \in \vec{\mathcal{O}}(G)$ be an optimal orientation of G w.r.t. MAXIM. If $d_{\Lambda|_{G_{c'c}}}^+(c) - d_{\Lambda|_{G_{c'c}}}^-(c)$ and $d_{\Lambda|_{G_c}}^+(c) - d_{\Lambda|_{G_c}}^-(c)$ do not have the same sign, then we switch the assignment of Λ of the edges of $G_{c'c}$. Doing so, the imbalance of all the vertices of G is unchanged except for that of c which got risen by 2 hence Λ is still optimal and $|d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| \leq |d_{\Lambda|_{G_c}}^+(c) - d_{\Lambda|_{G_c}}^-(c)| + 1$. Moreover,

$$\text{MAXIM}(G) = \min_{v \in V} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| \leq \min_{v \in V(G_c) \setminus \{c\}} |d_{\Lambda|_{G_c}}^+(v) - d_{\Lambda|_{G_c}}^-(v)|$$

which yields

$$\begin{aligned} \text{MAXIM}(G) &\leq \min\left(\min_{v \in V(G_c) \setminus \{c\}} |d_{\Lambda|_{G_c}}^+(v) - d_{\Lambda|_{G_c}}^-(v)|, |d_{\Lambda|_{G_c}}^+(c) - d_{\Lambda|_{G_c}}^-(c)| + 1\right) \\ &\leq \max_{\Lambda \in \vec{\mathcal{O}}(G_c)} \min\left(\min_{v \in V(G_c) \setminus \{c\}} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, |d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| + 1\right). \end{aligned}$$

□

Lemma 15 *Let G be a cactus such that $\delta_G = 2$. Then there exists a cycle of G with at most one gate (vertex adjacent to any vertex not belonging to the cycle).*

Proof. Let T be the block tree of G . If T has no leaf, then it is a vertex graph, hence G is a cycle which necessarily has no gate. If T has a leaf l , it necessarily corresponds to a block of G . If l was a bridge, the degree of its endpoint in G which is not a cutvertex would be 1, hence l is a cycle of G . And being a leaf in T , it has at most one gate in G . □

Proof (Theorem 13).

- $\Leftarrow$ We assume that $S = V_B$ and $\delta_G \geq 2$. From Proposition 12, we know that $\text{MAXIM}(G) \leq 2$. We will build an orientation $\Lambda \in \vec{\mathcal{O}}(G)$ such that $\min_{v \in V} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| = 2$. We start by orienting the edges of the subgraph S_0 consisting in the union of all the even cycles of G . S_0 having no odd cycles, it is bipartite and we can therefore choose an orientation of the edges of S_0 such that $|d_{\Lambda|_{S_0}}^+(v) - d_{\Lambda|_{S_0}}^-(v)| = d_{S_0}(v) \geq 2$, $\forall v \in V(S_0)$. We now recursively extend Λ to the rest of the blocks in S ensuring an imbalance of at least 2 for each vertex adjacent to at least two oriented edges. Let $Z \in V_B$ be an unoriented block of G that is either:

- an odd cycle of G sharing a cut-vertex with an oriented cycle or two oriented bridges,
- a bridge sharing a cut vertex with an oriented block.

If there was no such block and the graph G was not totally oriented, the set of oriented blocks of G denoted by $\vec{S}$ would contradict the minimality of S .

If Z is an odd cycle, we choose a cut vertex c of Z adjacent to oriented edges and name the vertices of B $c = v_1 v_2 \cdots v_k = c$. We assign:

$$\circ \Lambda(v_i v_{i+1}) = \begin{cases} \overrightarrow{v_i v_{i+1}} & \text{if } i \text{ is odd;} \\ \overrightarrow{v_{i+1} v_i} & \text{otherwise} \end{cases}, \forall i \in \llbracket 1, k-1 \rrbracket.$$

Let us now consider the imbalance of the vertices of Z . Since c is adjacent to either a cycle or two bridges in $\vec{S}$, it is adjacent to at least two edges in $\vec{S}$ and therefore $|d_{\Lambda|\vec{S}}^+(c) - d_{\Lambda|\vec{S}}^-(c)| \geq 2$ according to our inductive hypothesis. Since $|d_{\Lambda|Z}^+(c) - d_{\Lambda|Z}^-(c)| = 0$, $|d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| = |d_{\Lambda|\vec{S}}^+(c) - d_{\Lambda|\vec{S}}^-(c)| \geq 2$. If there is another cut-vertex c' of Z that is adjacent to a block in $\vec{S}$ such that $d_{\Lambda|\vec{S}}^+(c') - d_{\Lambda|\vec{S}}^-(c')$ and $d_{\Lambda|Z}^+(c') - d_{\Lambda|Z}^-(c')$ do not have the same sign, then we switch the assignment of Λ of the edges of the whole connected component of $\vec{S}$ containing c' for its opposite. Necessarily, c' is the only vertex this connected component shares with Z otherwise Z would be contained in a bigger block of G . Then doing so, the imbalance of all vertices is left unchanged except for that of c' which is now equal to $|d_{\Lambda|\vec{S}}^+(c') - d_{\Lambda|\vec{S}}^-(c')| + |d_{\Lambda|Z}^+(c') - d_{\Lambda|Z}^-(c')| = |d_{\Lambda|\vec{S}}^+(c') - d_{\Lambda|\vec{S}}^-(c')| + 2$.

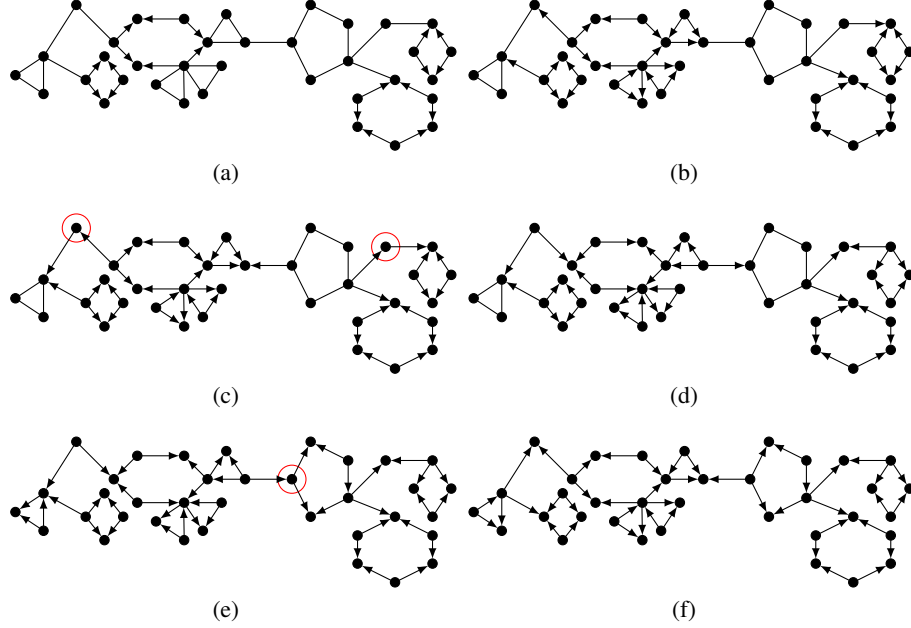
This process is repeated for all the cut-vertices c' of Z different from c adjacent to a block in $\vec{S}$ to be such that $|d_{\Lambda}^+(c') - d_{\Lambda}^-(c')| \geq 2$. If $u \in V(Z)$ is not a cut-vertex, then $|d_{\Lambda}^+(u) - d_{\Lambda}^-(u)| = |d_{\Lambda|Z}^+(u) - d_{\Lambda|Z}^-(u)| = 2$ and Λ becomes an orientation of $\vec{S} \cup \{Z\}$ with imbalance at least two for all the vertices of G adjacent to at least two oriented edges.

If Z is a bridge, we take c one of its endpoints adjacent to an oriented edge, we call u the other one and assign

$$\Lambda(cu) = \begin{cases} \overrightarrow{cu} & \text{if } d_{\Lambda|\vec{S}}^+(c) > d_{\Lambda|\vec{S}}^-(c); \\ \overrightarrow{uc} & \text{otherwise.} \end{cases}$$

Concerning the imbalance of c , by inductive hypothesis, $|d_{\Lambda|\vec{S}}^+(c) - d_{\Lambda|\vec{S}}^-(c)| \geq 1$, then $|d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| = |d_{\Lambda|\vec{S}}^+(c) - d_{\Lambda|\vec{S}}^-(c)| + |d_{\Lambda|Z}^+(c) - d_{\Lambda|Z}^-(c)| \geq 2$. If u is adjacent to a block in $\vec{S}$ as well and $d_{\Lambda|\vec{S}}^+(u) - d_{\Lambda|\vec{S}}^-(u)$ and $d_{\Lambda|Z}^+(u) - d_{\Lambda|Z}^-(u)$ do not have the same sign then we switch the assignment of Λ of the edges of the whole connected component of $\vec{S}$ containing u for its opposite. This connected component necessarily doesn't contain c otherwise z would be contained in a bigger block of G . Then doing so, the imbalance of all vertices is left unchanged except for that of u which is now equal to $|d_{\Lambda|\vec{S}}^+(u) - d_{\Lambda|\vec{S}}^-(u)| + |d_{\Lambda|Z}^+(u) - d_{\Lambda|Z}^-(u)| = |d_{\Lambda|\vec{S}}^+(u) - d_{\Lambda|\vec{S}}^-(u)| + 1 \geq 2$. Λ thus becomes an orientation of $\vec{S} \cup \{Z\}$ with imbalance at least two for all the vertices of G adjacent to at least two oriented edges.

Fig. 9: Orienting the edges of G verifying $\delta_G = 2$ and $S = V_B$ in such a way that $\text{MAXIM}(G) = 2$.



We now add Z to $\vec{S}$ and proceed like this for all blocks Z in $S \setminus \vec{S}$ until $\vec{S} = S$. Now since $V_B = S$ and $\delta_G = 2$, we conclude that Λ is an orientation of all the edges of G with an imbalance equal to 2.

In Fig. 9 can be found the orienting process of the cactus presented in Fig. 7. First, the even cycles are oriented (Fig. 9(a)), then the blocks adjacent to the even cycles (Fig. 9(b)) and then iteratively the unoriented blocks adjacent to oriented blocks (Fig. 9(c), (d), (e) and (f)). In Fig. 9(c) and (e), the vertices with imbalance zero are circled in red, and in the next step, the orientation is reversed on one of the subtree where the circled vertex is the root in order to ensure an orientation imbalance of at least 2.

- $\Rightarrow$ If $\delta_G < 2$ then $\text{MAXIM}(G) < 2$. So we assume that $\delta_G = 2$ and $S \subsetneq V_B$. Let $c \in V(\bar{S}) \cap V(S)$. Belonging to both $V(S)$ and $V(\bar{S})$, c must belong to at least two different blocks of G and is therefore a cutvertex of G . Let Z be a block in S adjacent to c , Z is necessarily a bridge of G otherwise all the blocks adjacent to Z would be in S thus contradicting $c \in V(\bar{S})$, and for the same reason, Z is the only bridge in S adjacent to c . Calling c' the endpoint of Z that is not c and G_c the subgraph of G obtained by taking the connected component of $G \setminus cc'$ containing c , we are in the conditions of lemma 14 and thus we obtain

$$\text{MAXIM}(G) \leq \max_{\Lambda \in \vec{\mathcal{O}}(G_c)} \min \left(\min_{v \in V(G_c) \setminus \{c\}} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, |d_{\Lambda}^+(c) - d_{\Lambda}^-(c)| + 1 \right).$$

Proceeding like this for all $c \in V(\bar{S}) \cap V(S)$ yields that $\text{MAXIM}(G)$ is smaller than

$$\max_{\Lambda \in \vec{\mathcal{O}}(\bar{S})} \min \left(\min_{v \in V(\bar{S}) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(\bar{S}) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right).$$

Now if we choose a connected component $\bar{S}_0$ of $\bar{S}$ we get that $\text{MAXIM}(G)$ is smaller than

$$\max_{\Lambda \in \vec{\mathcal{O}}(\bar{S}_0)} \min \left(\min_{v \in V(\bar{S}_0) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(\bar{S}_0) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right).$$

Now we show that the right-hand part of the previous inequality is lower than or equal to 1. Suppose it equals 2 and let $\Lambda \in \vec{\mathcal{O}}(\bar{S}_0)$ satisfying

$$\min \left(\min_{v \in V(\bar{S}_0) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(\bar{S}_0) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right) \geq 2. \quad (3)$$

Thus we have $d_{\bar{S}_0}(v) \geq 2$, $\forall v \in V(\bar{S}_0) \setminus V(S)$ and we know that for any $v \in V(\bar{S}_0) \cap V(S)$, all the blocks in $\bar{S}_0$ adjacent to v are odd cycles, hence $d_{\bar{S}_0}(v) \geq 2$. So $\delta_{\bar{S}_0} \geq 2$ and according to lemma 15, there is a cycle C of $\bar{S}_0$ with at most one gate. If C has no gate then it means that $\bar{S}_0$ consists only of C and since we know that all the cycles of $\bar{S}_0$ are odd, it directly contradicts the existence of Λ . Hence C has exactly one gate g . So for all $v \in V(C) \setminus \{g\}$, $|d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| = |d_{\Lambda_C}^+(v) - d_{\Lambda_C}^-(v)|$ and if we name the vertices of C $g = v_1 v_2 \dots v_k = g$ the assignment of Λ must be

$$\circ \Lambda(v_i v_{i+1}) = \begin{cases} \overrightarrow{v_i v_{i+1}} & \text{if } i \text{ is odd;} \\ \overrightarrow{v_{i+1} v_i} & \text{otherwise} \end{cases}, \forall i \in \llbracket 1, k-1 \rrbracket.$$

or its reverse so that Λ satisfies our assumption. Thus we have

$$\begin{aligned} & \max_{\Lambda \in \vec{\mathcal{O}}(\bar{S}_0)} \min \left(\min_{v \in V(\bar{S}_0) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(\bar{S}_0) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right) \\ &= \max_{\Lambda \in \vec{\mathcal{O}}(\bar{S}_0 \setminus C)} \min \left(\min_{v \in V(\bar{S}_0 \setminus C) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(\bar{S}_0 \setminus C) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right). \end{aligned}$$

If there exists a vertex of degree one in $\bar{S}_0 \setminus C$ then it is adjacent to a bridge in $\bar{S}_0$ and is therefore in $V(\bar{S} \setminus C) \setminus V(S)$, thus contradicting (3). So we assume that $\delta_{\bar{S}_0 \setminus C} = 2$ and we can reiterate the same process with another odd cycle with exactly one gate until we are left with an odd cycle C_{end} with no gates and conclude that

$$\max_{\Lambda \in \vec{\mathcal{O}}(C_{end})} \min \left(\min_{v \in V(C_{end}) \setminus V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)|, \min_{v \in V(C_{end}) \cap V(S)} |d_{\Lambda}^+(v) - d_{\Lambda}^-(v)| + 1 \right) \geq 2;$$

Since this is impossible for an odd cycle, we deduce that (3) is wrong. Hence, $\text{MAXIM}(G) \leq 1$.

□

As already mentioned, Theorem 13 leads to a simple polynomial-time algorithm to compute $\text{MAXIM}(G)$ for cacti. Starting from the set S_0 of even cycles, we add blocks to S using the two operations defined above: bridges sharing a cut-vertex with a block in S , and odd cycles sharing a cut-vertex with either a cycle or two bridges that are already in S , are added to S .

5 Mixed integer linear programming formulations

In this section, we gradually introduce two formulations for the MAXIM problem. For our purposes, we shall consider the original graph $G = (V, E)$ is directed (consider any arbitrary orientation) and let $B \in \{-1, 0, 1\}^{|V| \times |E|}$ stand for its incidence matrix, i.e., the column corresponding to the arc uv (or, equivalently, to the edge uv directed from node u to node v), has only nonzero entries in the rows corresponding to the nodes u and v : $B_{u,uv} = 1$ and $B_{v,uv} = -1$, respectively.

In order to describe an orientation of the graph G , we take orientation variables $x \in \{-1, 1\}^{|E|}$ interpreted as follows. For each edge $uv \in E$ which is originally directed from node u to node v : if $x_{uv} = 1$ then uv is directed from u to v (i.e., the orientation is the same as the original one) and is directed from v to u otherwise (i.e., the edge is “reversed” with respect to the original orientation). Then if we look at the product of B with an orientation vector $x \in \{-1, 1\}^{|E|}$ we obtain $B_v x = d_x^+(v) - d_x^-(v)$, $\forall v \in V$, where $d_x^+(v)$ (resp. $d_x^-(v)$) is the outdegree (resp. indegree) of $v \in V$ in G w.r.t. the orientation described by x and B_v denotes the row of the matrix B which corresponds to node v . Hence the MAXIM problem can be expressed as the following non-linear formulation.

$$\begin{cases} \max h \\ \text{s.t.} \\ h \leq |B_v x|, \forall v \in V \\ x \in \{-1, 1\}^{|E|}. \end{cases}$$

Let us derive from the preceding non-linear formulation an equivalent linear formulation. To that purpose, we remove the absolute value of the constraint by squaring it : $h^2 \leq (B_v x)^2$, $\forall v \in V$. Since it still is not linear, we have to consider variables representing the product of two variables and for that we substitute the x variables by their 0-1 version. So developing, we obtain

$$\begin{aligned} & (B_v(2x - 1))^2 \\ &= \left(\sum_{uv \in E} B_{v,uv}(2x_{uv} - 1) \right)^2 \\ &= \sum_{uv \in E} B_{v,uv}^2 (2x_{uv} - 1)^2 + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (2x_{uv} - 1)(2x_{wv} - 1) \\ &= \sum_{uv \in E} (4x_{uv}^2 - 4x_{uv} + 1) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4x_{uv}x_{wv} - 2x_{uv} - 2x_{wv} + 1) \\ &= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4x_{uv}x_{wv} - 2x_{uv} - 2x_{wv} + 1). \end{aligned}$$

Furthermore, since maximizing h is equivalent to maximizing its square root, substituting h^2 by h , we obtain the following formulation.

$$\begin{cases} \max h \\ \text{s.t.} \\ h \leq d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4x_{uv}x_{wv} - 2x_{uv} - 2x_{wv} + 1), \forall v \in V \\ x \in \{0, 1\}^{|E|}. \end{cases}$$

In order to linearize it, we introduce product variables $z_{uv,wp} \in \{0, 1\}$, $uv, wp \in E$, $uv \neq wp$ representing the variables product $x_{uv}x_{wp}$. For a pair of edges $(uv, wp) \in E^2$, $uv \neq wp$, we add the constraints $z_{uv,wp} \leq x_{uv}$, $z_{uv,wp} \leq x_{wp}$ and $z_{uv,wp} \geq x_{uv} + x_{wp} - 1$ so as to force $z_{uv,wp} = x_{uv}x_{wp}$. We can relax the integer constraint on the z variables and we obtain the following mixed integer linear programming formulation.

$$\text{(MIP1)} \quad \begin{cases} \max h \\ \text{s.t.} \\ h \leq d(v) + 2 \sum_{\substack{uv, wp \in E \\ uv \neq wp}} B_{v,uv} B_{v,wp} (4z_{uv,wp} - 2x_{uv} - 2x_{wp} + 1), \forall v \in V \\ z_{uv,wp} \leq x_{uv} \\ z_{uv,wp} \leq x_{wp} \\ z_{uv,wp} \geq x_{uv} + x_{wp} - 1 \\ x \in \{0, 1\}^{|E|}, z_{uv,wp} \geq 0, uv, wp \in E, uv \neq wp, h \in \mathbb{R}. \end{cases}, \forall uv, wp \in E, uv \neq wp$$

Theorem 16 For any graph G ,

$$\text{MIP1}(G) = \text{MAXIM}(G),$$

Where $\text{MIP1}(G)$ is the square root optimal of the objective value of MIP1 for G .

Proof. $x \in \{0, 1\}^{|E|}$ covers all the possible orientations of G and for every vertex $v \in V$, the first constraint is equivalent to $h \leq (d_x^+(v) - d_x^-(v))^2$. $\square$

Thus we have a first mixed integer linear programming formulation for our problems with $O(m^2)$ variables, $O(m)$ of which are integer variables and $O(m^2)$ constraints.

Let us take a look at the linear program obtained by relaxing the integer constraint of MIP1 on an input graph $G = (V, E)$, and let us consider the triplet (x^{lp}, z^{lp}, h^{lp}) where $x_{uv}^{lp} = \frac{1}{2}$, $\forall uv \in E$; $z_{uv,wp}^{lp} = 0$ for all pairs of edges $uv, wp \in E$ that share no endpoint; $z_{uv,wp}^{lp} = \frac{1+B_{v,uv}B_{v,wp}}{4}$ for all pairs of edges $uv, wp \in E$ (i.e. all pairs of edges in E that share an endpoint), and $h^{lp} = \delta_G^2$.

Lemma 17 (x^{lp}, z^{lp}, h^{lp}) is a feasible solution of the linear relaxation of MIP1 with objective value δ_G^2 .

Proof. Observe that $\forall uv, wp \in E$,

$$0 = x_{uv}^{lp} + x_{wp}^{lp} - 1 \leq z_{uv,wp}^{lp} \leq x_{uv}^{lp} = x_{wp}^{lp} = \frac{1}{2}.$$

Moreover, $\forall v \in V$,

$$\begin{aligned}
& d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4z_{uv,wv}^{lp} - 2x_{uv}^{lp} - 2x_{wv}^{lp} + 1) \\
&= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (1 + B_{v,uv} B_{v,wv} - 1 - 1 + 1) \\
&= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} (B_{v,uv} B_{v,wv})^2 \\
&= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} 1 \\
&= d(v) + (d(v) - 1)d(v) \\
&= d(v)^2 \geq \delta_G^2.
\end{aligned}$$

□

The previous lemma proves that the optimal value of the linear relaxation of MIP1 is at least the minimum degree of the input graph squared, which corresponds to the trivial upper bound of MAXIM(G).

We present a second formulation with reduced number of variables and constraints. This second formulation involves the same orientation variables $x \in \{-1, 1\}^{|E|}$ we described earlier and a second type of variables that are binary: indicator variables y_k^v with $v \in V$ a vertex of G and $k \in \llbracket -d(v), d(v) \rrbracket$. They have the following interpretation: $y_k^v = 1$ if and only if $B_v x = d_x^+(v) - d_x^-(v) = k$, so that the following equation trivially holds

$$\sum_{k \in \llbracket -d(v), d(v) \rrbracket} k y_k^v = B_v x, \forall v \in V.$$

Given the interpretation for the variables y , among those of the form y_k^v , for some fixed node $v \in V$, exactly one of them has value 1. Thus, the following constraints are satisfied

$$\sum_{k \in \llbracket -d(v), d(v) \rrbracket} y_k^v = 1, \forall v \in V.$$

Notice that for a vertex $v \in V$, the difference between its outdegree and indegree w.r.t. any orientation and its degree have the same parity. Thus instead of running k through $\llbracket -d(v), d(v) \rrbracket$, we can limit to $k \in \llbracket -d(v), d(v) \rrbracket$, s.t. $k \equiv d(v)[2]$, i.e. the only possible values of $d_x^+(v) - d_x^-(v)$ when x runs through $\{-1, 1\}^{|E|}$. Then we can show the MAXIM problem may be formulated as the mixed integer program

$$(\text{MIP2}) \begin{cases} \max h \\ \text{s.t.} \\ h \leq \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v)[2]}} \min(|k|, \delta_G) y_k^v, \forall v \in V \\ \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v)[2]}} y_k^v = 1, \forall v \in V \\ \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v)[2]}} k y_k^v = B_v x, \forall v \in V \\ x \in [-1; 1]^{|E|}, y_k^v \in \{0, 1\}, \forall (v, k) \in V \times \llbracket -d(v), d(v) \rrbracket, \text{ s.t. } k \equiv d(v)[2], h \in \mathbb{R}. \end{cases}$$

Theorem 18 For any graph G ,

$$\text{MIP2}(G) = \text{MAXIM}(G),$$

Where $\text{MIP2}(G)$ is the optimal objective value of MIP2 for G .

Proof. First, if the orientation variables x were constrained to be integers, since $y_k^v = 1$ if and only if $B_v x = d_x^+(v) - d_x^-(v) = k$ and $x \in \{0, 1\}^{|E|}$ covers all the possible orientations of G and for every vertex $v \in V$, the first and third constraints lead to $h \leq |d_x^+(v) - d_x^-(v)|$ and the optimal objective value of the resulting formulation would equal $\text{MAXIM}(G)$.

Now we know that the incidence matrix B is totally unimodular. Hence, there exists an integer optimal solution (x^*, y^*, h^*) of MIP2. If $x^* \in \{-1, 1\}^{|E|}$ then (x^*, y^*, h^*) is solution of the all-integer version of MIP2 mentioned above and therefore optimal, i.e. its objective value is $\text{MAXIM}(G)$. Otherwise x^* describes a partial orientation of G , i.e. for an edge $uv \in E$, if $x_{uv}^* = 1$ then the original orientation of the edge is preserved, if $x_{uv}^* = -1$ then it is reversed and if $x_{uv}^* = 0$ then the edge is left unoriented. We know that for each vertex $v \in V$, $B_v x^* \equiv d(v)[2]$. So the number of edges adjacent to v on which x^* is non-zero must have the same parity as the degree of v . In other words, the number of edges adjacent to v on which x^* is zero must be even. That is to say, let $E' = \{uv \in E | x_{uv}^* = 0\}$ and $G' = (V, E')$, $d_{G'}(v) \equiv 0[2]$ for each vertex $v \in V$. Since G' is eulerian, we can take a cycle of G' , orient it in a way that does not change the imbalance of any vertex, remove it and proceed like this until there are no more edges in G' . The resulting (complete) orientation can be described by an orientation vector $x' \in \{-1, 1\}^{|E|}$ and the triplet (x', y^*, h^*) is a solution of the all-integer version of MIP2 having the same objective value as (x^*, y^*, h^*) , hence it is optimal and therefore equal to $\text{MAXIM}(G)$. $\square$

We now have a second mixed integer linear programming formulation of MAXIM with $O(m+n)$ variables, $O(m)$ of which are integer variable and $O(n)$ constraints.

Let us consider the triplet (x^{lp}, y^{lp}, h^{lp}) where $x_{uv}^{lp} = 0, \forall uv \in E$; $y_k^{v,lp} = 0, \forall (v, k) \in V \times \llbracket -d(v) + 1, d(v) - 1 \rrbracket$, s.t. $k \equiv d(v)[2]$; and $y_{-d(v)}^{v,lp} = y_{d(v)}^{v,lp} = \frac{1}{2}, \forall v \in V$. Observe that (x^{lp}, y^{lp}, h^{lp}) is a feasible solution of the linear relaxation of MIP2 with objective value δ_G . In other words, the linear relaxation MIP2 is generally weak. Let us then try to strengthen it using some valid inequalities.

Remember from Section 2 that it is easy to check whether $\text{MAXIM}(G) = 0$. One can then set variables y_0^v to 0 when $\text{MAXIM}(G) > 0$. We also know from the discussion at the end of Section 3 that $\text{MAXIM}(G)$ can almost not be less than $\frac{\delta_G}{2}$. More precisely, if $\delta_G \equiv 0[4]$ then $\text{MAXIM}(G) \geq \frac{\delta_G}{2}$ while $\text{MAXIM}(G) \geq \frac{\delta_G - 1}{2}$ when δ_G is odd and $\text{MAXIM}(G) \geq \frac{\delta_G}{2} - 1$ when $\delta_G \equiv 2[4]$.

More generally, if l is a known lower bound for $\text{MAXIM}(G)$, then all variables y_k^v for $|k| < l$ can be fixed to 0. To find such a lower bound, we use a standard greedy algorithm to find a locally maximum cut and orient edges as described in the proof of Theorem 9.

Let u be an upper bound for $\text{MAXIM}(G)$. We already know that $\text{MAXIM}(G) \leq \delta_G$, so we can assume that $u \leq \delta_G$. Consider the following inequality

$$h \leq u - \sum_{v=1}^n \sum_{\substack{k \in \llbracket 0, u-1 \rrbracket \\ k \equiv d(v)[2]}} \lambda_k^v (y_k^v + y_{-k}^v), \quad \forall \lambda \in \Lambda_u \quad (4)$$

where the vertices of G are numbered from 1 to $|V| = n$, and

$$\Lambda_u = \left\{ \lambda = (\lambda_k^v)_{(v,k) \in \llbracket 1, n \rrbracket \times \llbracket 0, u-1 \rrbracket} \in \mathbb{N}^{nu} \mid \begin{array}{l} \lambda_{k+1}^v \leq \lambda_k^v, \quad \forall (v, k) \in \llbracket 1, n \rrbracket \times \llbracket 0, u-2 \rrbracket \\ \sum_{v=1}^n \lambda_k^v = u - k, \quad \forall k \in \llbracket 0, u-1 \rrbracket \end{array} \right\}.$$

Observe that coefficients λ_k^v are non-negative integer numbers and are non increasing in k . For each k , there exists only one v such that $\lambda_{k+1}^v = \lambda_k^v - 1$ while $\lambda_{k+1}^w = \lambda_k^w$ for any $w \neq v$. Let us prove that inequalities (4) are valid inequalities for the convex hull of feasible solutions of MIP2 denoted by $\mathcal{P}_2$.

Proposition 19 *Inequalities (4) are valid for $\mathcal{P}_2$.*

Proof. Consider a feasible solution (x, y, h) of $\mathcal{P}_2$. If $y_k^v = 0$ and $y_{-k}^v = 0$ for any $v \in \llbracket 1, n \rrbracket$ and any $k \in \llbracket 0, u-1 \rrbracket$, then $\sum_{v=1}^n \sum_{\substack{k \in \llbracket 0, u-1 \rrbracket \\ k \equiv d(v)[2]}} \lambda_k^v (y_k^v + y_{-k}^v) = 0$ and inequality (4) becomes $h \leq u$. The last is valid since u is an upper bound for $\text{MAXIM}(G)$.

Let us now assume that $h \leq u - 1$, then there exist $v \in \llbracket 1, n \rrbracket$ and $k \leq u - 1$ such that either y_k^v or y_{-k}^v is equal to 1 while $y_{k'}^w = 0$ for any $w \in \llbracket 1, n \rrbracket$ and k' such that $|k'| < k$.

Using the fact that y_k^w is non increasing, we deduce that $\sum_{\substack{k' \in \llbracket 0, u-1 \rrbracket \\ k' \equiv d(w)[2]}} \lambda_{k'}^w (y_{k'}^w + y_{-k'}^w) \leq \lambda_k^w$. Summing up these inequalities for all w and using the fact that $\sum_{v=1}^n \lambda_k^v = u - k$ leads to $h \leq k$ which is valid by k definition. $\square$

Observe that the first family of inequalities included in MIP2 can be seen as a special case of inequalities (4).

Let us now study the separation problem of inequalities (4).

Proposition 20 *Inequalities (4) can be separated in polynomial time.*

Proof. Given a fractional solution (x, y, h) , one can check whether an inequality of type (4) is violated by looking for coefficients $\lambda \in \Lambda_u$ maximizing $\sum_{v=1}^n \sum_{\substack{k \in \llbracket 0, u-1 \rrbracket \\ k \equiv d(v)[2]}} \lambda_k^v (y_k^v + y_{-k}^v)$. Remember that for each k , there exists only one v such that $\lambda_{k+1}^v = \lambda_k^v - 1$ while $\lambda_{k+1}^w = \lambda_k^w$ for any $w \neq v$. Let v_k be such v . Then $\sum_{v=1}^n \sum_{\substack{k \in \llbracket 0, u-1 \rrbracket \\ k \equiv d(v)[2]}} \lambda_k^v (y_k^v + y_{-k}^v)$ can be written as $\sum_{k \in \llbracket 0, u-1 \rrbracket} \sum_{k' \in \llbracket 0, k \rrbracket} (y_{k'}^{v_k} + y_{-k'}^{v_k})$. This immediately leads to the following algorithm. First, all λ_k^v are initially set to 0. Then, we select v_{u-1} maximizing $\sum_{k \in \llbracket 0, u-1 \rrbracket} (y_k^{v_{u-1}} + y_{-k}^{v_{u-1}})$ and we increment by 1 all $\lambda_k^{v_{u-1}}$: $\lambda_k^{v_{u-1}} = \lambda_k^{v_{u-1}} + 1$ for any $k \leq u - 1$. More generally, for each $w \in \llbracket 0, u-1 \rrbracket$, we select v_w maximizing $\sum_{\substack{k \in \llbracket 0, w \rrbracket \\ k \equiv d(v)[2]}} (y_k^{v_w} + y_{-k}^{v_w})$ and we increment by 1 all $\lambda_k^{v_w}$ for $k \leq w$. The algorithm has clearly a polynomial-time complexity. $\square$

A second family of valid inequalities is defined for each vertex v , each integer number $p \in \llbracket 1, d(v) \rrbracket$ and each subset of p edges incident to v .

$$\sum_{\{\{vj_1\}, \dots, \{vj_i\}, \dots, \{vj_p\}\} \subset \text{cut}(\{v\})} B_{v, \{vj_i\}} x_{\{vj_i\}} + \sum_{k \in \llbracket 0, p-1 \rrbracket} 2(p-k) y_{2k-d(v)}^v \leq p. \quad (5)$$

Proposition 21 *Inequalities (5) are valid for $\mathcal{P}_2$.*

Proof. If all variables $y_{2k-d(v)}^v$ are equal to 0 for $k \in \llbracket 0, p-1 \rrbracket$, then inequality (5) is implied by the fact that $x \in [-1; 1]^{|E|}$.

Assume that $y_{2k_0-d(v)}^v = 1$ for some $k_0 \in \llbracket 0, p-1 \rrbracket$. This requires that among all edges of $\text{cut}(\{v\})$ there are exactly k_0 (resp. $d(v) - k_0$) edges $\{vj\}$ such that $B_{v, \{vj\}} x_{\{vj\}} = 1$ (resp. $B_{v, \{vj\}} x_{\{vj\}} = -1$). Consequently $\sum_{\{\{vj_1\}, \dots, \{vj_i\}, \dots, \{vj_p\}\} \subset \text{cut}(\{v\})} B_{v, \{vj_i\}} x_{\{vj_i\}}$ is upper bounded by $k_0 - (p - k_0) = 2k_0 - p$. Moreover, we have $\sum_{k \in \llbracket 0, p-1 \rrbracket} 2(p-k) y_{2k-d(v)}^v = 2(p - k_0)$. Inequality (5) immediately follows. $\square$

The separation problem of inequalities (5) is also easy to solve.

Proposition 22 *Inequalities (5) can be separated in polynomial time.*

Proof. Given a fractional solution (x, y, h) , for each vertex v and any $p \in \llbracket 1, d(v) \rrbracket$, we order the edges of $\text{cut}(\{v\})$ in descending order according to $B_{v, \{vj\}} x_{\{vj\}}$ and we select the first p edges. Then a violated inequality can be detected by comparison of $\sum_{\{\{vj_1\}, \dots, \{vj_i\}, \dots, \{vj_p\}\} \subset \text{cut}(\{v\})} B_{v, \{vj_i\}} x_{\{vj_i\}} + \sum_{k \in \llbracket 0, p-1 \rrbracket} 2(p-k) y_{2k-d(v)}^v$ with p . $\square$

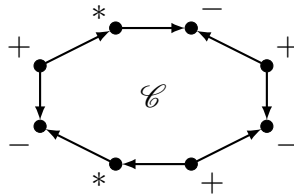
Let us now consider any cycle $\mathcal{C}$ and the two following inequalities.

$$\sum_{v \in \mathcal{C}} (2y_{d(v)}^v + y_{d(v)-2}^v) \leq |\mathcal{C}|, \quad (6)$$

$$\sum_{v \in \mathcal{C}} (2y_{-d(v)}^v + y_{-d(v)+2}^v) \leq |\mathcal{C}|. \quad (7)$$

Proposition 23 *Inequalities (6) and (7) are valid for $\mathcal{P}_2$.*

Fig. 10: A vertex of a cycle $\mathcal{C}$ is respectively in $\mathcal{C}^+$, $\mathcal{C}^-$ or $\mathcal{C}^*$ if it has two, zero or one outgoing incident edge(s) in $\mathcal{C}$.



Proof. Let us focus on the validity of inequalities (6) (inequalities (7) being provable in the same way). After orientation of the edges of the cycle $\mathcal{C}$, let $\mathcal{C}^+$ be the set of vertices of $\mathcal{C}$ for which their two incident edges are oriented outward from these vertices to their neighbours. We also define $\mathcal{C}^-$ in the same way by considering vertices having two edges of the cycle oriented from their neighbours to them. The remaining set of vertices of the cycle is denoted by $\mathcal{C}^*$ (they have an incoming incident edge and an outgoing incident edge). These definitions are illustrated on Fig. 10. It is now easy to see that we always have $|\mathcal{C}^+| = |\mathcal{C}^-|$.

Suppose that $y_{d(v)}^v = 1$, then $v \in \mathcal{C}^+$. Observe also that $y_{d(v)-2}^v = 1$ requires v to be in $\mathcal{C}^+ \cup \mathcal{C}^*$. Consequently, $\sum_{v \in \mathcal{C}} (y_{d(v)}^v + y_{d(v)-2}^v) \leq |\mathcal{C}^+| + |\mathcal{C}^*|$ and $\sum_{v \in \mathcal{C}} y_{d(v)}^v \leq |\mathcal{C}^+|$. Summing up both inequalities leads to $\sum_{v \in \mathcal{C}} (2y_{d(v)}^v + y_{d(v)-2}^v) \leq 2|\mathcal{C}^+| + |\mathcal{C}^*| = |\mathcal{C}^+| + |\mathcal{C}^-| + |\mathcal{C}^*| = |\mathcal{C}|$. $\square$

Proposition 24 *Inequalities (6) and (7) can be separated in polynomial time.*

Proof. Let us again focus on inequalities (6) since (7) can be separated in a similar way. Given a fractional solution (x, y, h) , we need an algorithm to either compute a cycle $\mathcal{C}$ such that $|\mathcal{C}| - \sum_{v \in \mathcal{C}} (2y_{d(v)}^v + y_{d(v)-2}^v) < 0$ or to certify that such a cycle does not exist. For each edge uv , let us consider a weight $c_{uv} = 1 - (y_{d(v)}^v + \frac{1}{2}y_{d(v)-2}^v) - (y_{d(u)}^u + \frac{1}{2}y_{d(u)-2}^u)$. Then we clearly have

$$|\mathcal{C}| - \sum_{v \in \mathcal{C}} (2y_{d(v)}^v + y_{d(v)-2}^v) = \sum_{\{uv\} \in \mathcal{C}} c_{uv}.$$

We are then looking for an undirected cycle having a negative total weight. This can be done, for example, by computing a minimum weight $\emptyset$ -join, where the last is a set of edge disjoint cycles. This can be done using the algorithm of [16]. Notice that this algorithm can provide many negative weight cycles with some 0 weight cycles. If the total weight of the $\emptyset$ -join is strictly negative, a negative weight cycle can be easily extracted. A different algorithm can also be found in [5] (see section 6 of the reference). $\square$

Given any clique $\mathcal{K}$ and any number $p \in \llbracket 1, |\mathcal{K}| \rrbracket$, we consider the two following inequalities.

$$\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} (p-k)y_{d(v)-2k}^v \leq \frac{p(p+1)}{2}, \quad \forall p \in \llbracket 1, |\mathcal{K}| \rrbracket, \quad (8)$$

$$\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} (p-k)y_{2k-d(v)}^v \leq \frac{p(p+1)}{2}, \quad \forall p \in \llbracket 1, |\mathcal{K}| \rrbracket. \quad (9)$$

Inequalities (8) can be seen as a generalization of the obvious inequalities $\sum_{v \in \mathcal{K}} y_{d(v)}^v \leq 1$ obtained when $p = 1$.

Proposition 25 *Inequalities (8) and (9) are valid for $\mathcal{P}_2$.*

Proof. We can focus on inequalities (8). To maximize the left hand side of (8), we can assume that all edges in $\text{cut}(\mathcal{K})$ are oriented from $\mathcal{K}$ to outside (this is due to the fact that the coefficient $p - k$ increases when k decreases). Then, $y_{d(v)-2k}^v = 1$ is equivalent to say that exactly k edges, inside $\mathcal{K}$ and incident to v , are oriented from v to outside. Observe that inequality (8) is equivalent to

$$p \sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} y_{d(v)-2k}^v \leq \sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} k y_{d(v)-2k}^v + \frac{p(p+1)}{2}. \quad (10)$$

Let q be the number of vertices of $\mathcal{K}$ whose outdegree (in the graph induced by $\mathcal{K}$) is at most $p - 1$ after orientation. The left hand side of (10) is given by pq . The sum of the q lowest outdegrees is exactly given by $\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} k y_{d(v)-2k}^v$. Using Landau's theorem for tournaments [18], we can write that $\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} k y_{d(v)-2k}^v \geq \frac{q(q-1)}{2}$. Adding $\frac{p(p+1)}{2}$ to both sides, we get $\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} k y_{d(v)-2k}^v + \frac{p(p+1)}{2} \geq \frac{p^2 + q^2 + p - q}{2}$. Moreover, from $(q - p)^2 \geq q - p$, we get that $\frac{p^2 + q^2 + p - q}{2} \geq pq$ proving (10). $\square$

To separate inequalities (8), the following heuristic is used. Given a fractional solution (x, y, h) , for each vertex v and any $p \in \llbracket 1, d(v) + 1 \rrbracket$, we use a greedy approach to find a locally maximum-weight clique where the weight of any vertex $u \in V$ is $\sum_{k=0}^{\min(p-1, d(u))} (p - k) y_{d(u)-2k}^u$. We start with $\mathcal{K} := \{v\}$ and then recursively find $u = \arg \max \{ \sum_{k=0}^{\min(p-1, d(w))} (p - k) y_{d(w)-2k}^w \mid w \in \cap_{v' \in \mathcal{K}} N_G(v') \}$, where $N_G(v)$ denotes the set of the neighbours of v in G , and add it to $\mathcal{K}$ until $\cap_{v' \in \mathcal{K}} N_G(v') = \emptyset$. Then if $|\mathcal{K}| \geq p$ we can derive the inequality (8) corresponding to $\mathcal{K}$.

6 Computational results

In order to assess the performance of formulations MIP1 and MIP2, we present some computational results related to a wide variety of graphs. All algorithms were written in C++ calling IBM's ILOG CPLEX optimizer[©] and experiments have been performed using a processor 1.9GHzx4, 15.6GB RAM with four cores.

While the implementation of MIP1 is pretty straightforward: the model is created with all inequalities described in Section 5 and then the mixed-integer-linear-programming solver of CPLEX is run with default parameters, the implementation of MIP2 needs to be detailed.

First of all, an initial solution is determined by computing a locally maximum cut using the standard greedy algorithm described in the proof of Theorem 9. A starting integer solution is then known for both formulations MIP1 and MIP2.

For both formulations and each instance, we set a limited total processing time of 15 minutes (900 seconds). If this limit is reached, then the process is interrupted and returns both the objective value of the current best integer solution L_{MIP1} or L_{MIP2} and the current best upper bound U_{MIP1} or U_{MIP2} .

For formulation MIP2, a cutting-plane algorithm is implemented based on the inequalities of Section 5. Inequalities are generated in the following order:

- We look for a violated inequality of the type (4) according to the proof of Proposition 20.
- We generate cliques using the heuristic described in the end of section 5 and check for a violated inequalities of the types (8) and (9).
- We search for a cycle with minimum weight with a simple flow formulation solved as mixed integer program where each edge $vw \in E$ has weight $\frac{1}{2}(2y_{d(v)}^v + y_{d(v)-2}^v + y_{d(w)}^w + y_{d(w)-2}^w)$. If we find a negative weighted cycle, then corresponding inequality of the type (6) is violated. We do the same with the weight $\frac{1}{2}(2y_{-d(v)}^v + y_{2-d(v)}^v + y_{-d(w)}^w + y_{2-d(w)}^w)$ for an edge vw to find a violated inequality of the type (7). For the sake of simplicity, inequalities (6) and (7) are not separated using the algorithms of [16] and [5] but using a simple integer linear program computing a minimum-weight cycle.

After various experimentations, we chose not to put the inequalities of type (5) in the cutting-plane phase because when included, while the number of generated inequalities increases excessively with the size of the graph, the optimal objective value of the linear program is left unimproved.

After the addition of violated inequalities, the linear relaxation is solved to get a fractional (x, y, h) solution and the process is repeated until no more violated inequalities can be found. The optimal objective value of the last LP is denoted by v_{LP2} . Notice that the cutting-plane phase is intentionally limited to less than 10 minutes (600 seconds). Thus, if either no more valid inequalities can be generated or the time spent in the cutting-plane phase reaches 10 minutes, we switch to branch&bound. The time spent in the cutting-plane phase is denoted by t_{LP2} .

CPLEX's mixed integer programming solver is used with default parameters to solve MAXIM. Notice that some automatic internal cuts are generated by the solver to accelerate the branch&bound. As already mentioned, the total running time is limited to 15 minutes. The time spent in the branch&bound phase is denoted by t_{MIP2} while the best lower and upper bounds are respectively denoted by L_{MIP2} and U_{MIP2} .

For each of the instances we report L_{MIP1} , U_{MIP1} , t_{MIP1} , v_{LP2} , t_{LP2} , L_{MIP2} , U_{MIP2} , and t_{MIP2} , where t_{LP2} , t_{MIP1} and t_{MIP2} are expressed in seconds. We also report $nb_{(4)}$, $nb_{(8,9)}$ and $nb_{(6,7)}$, respectively the number of inequalities of the type (4), (8) & (9) and (6) & (7) generated for each instance $G = (V, E)$ along with $|V|$, $|E|$ and δ_G .

The graph instances used for the computations are denoted as follows:

- K_n : the complete graph with n vertices,
- G_k^n : the n -dimensional grid of length k , i.e. the cartesian product of n paths graphs of length k : $\bullet_{i=1}^n L_k$, where L_k is a path graph of length k ,
- G_{n_1, n_2}^2 : the 2-dimensional grid, i.e. the cartesian product of two path graphs of length n_1 and n_2 : $L_{n_1} \bullet L_{n_2}$,
- tG_k^n : the n -dimensional toroidal grid of length k , i.e. the cartesian product of n cycles of length k : $\bullet_{i=1}^n C_k$, where C_k is a cycle of length k ,
- tG_{n_1, n_2}^2 : the 2-dimensional toroidal grid, i.e. the cartesian product of cycles of length n_1 and n_2 : $C_{n_1} \bullet C_{n_2}$,

- $P_{n,m}$: a randomly generated planar graph with n vertices and m edges,
- $R_{n,m}$: a randomly generated graph with n vertices and m edges.

More about the toroidal grids can be found in [10].

Table 1 shows that formulation MIP2 has a significantly better performance than MIP1. On many instances we can observe that the cutting plane algorithm of MIP2 drastically improved the upper bound δ_G , sometimes so far as to the optimal objective value as can be seen for example on the complete graphs, the planar graphs or most of the n -dimensional toroidal grids. while the number and nature of the generated inequalities varies a lot, it seems to never grow excessively. We can see that as soon as the size of the instance becomes substantial, formulation MIP1 seems to degenerate, needing too much time and/or memory to be processed, while MIP2 shows its ability to handle large size graphs.

7 Further research

While computing the most imbalanced orientation of a graph is generally difficult, the problem turns out to be easy for cacti. It may be the same for other graph classes. Characterizing such graph classes would be interesting.

Two mixed integer linear programming formulations of MAXIM have been presented. Several families of valid inequalities have been presented to strengthen one of the two formulations. Exhibition of other families of valid inequalities might be helpful to solve larger size problems.

Finally, one can also study the weighted version of the most imbalanced orientation problem.

Acknowledgments

The authors wish to thank the associate editor and the anonymous referees for many valuable comments and suggestions that have led to a substantially improved paper.

References

1. Y. Asahiro, E. Miyano, H. Ono & K. Zenmyo: ‘Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree’, *Proc. of the 3rd Int. Conf. on Algorithmic Aspects in Information and Management (AAIM2007)*, LNCS 4508, 167-177 (2007)
2. Y. Asahiro, E. Miyano & H. Ono: ‘Graph classes and the complexity of the graph orientation minimizing the maximum weighted outdegree’, *Proceedings of the fourteenth Computing: the Australasian Theory Symposium (CATS2008)*, Wollongong, NSW, Australia (2008)
3. Y. Asahiro, J. Jansson, E. Miyano & H. Ono: ‘Degree constrained graph orientation: maximum satisfaction and minimum violation’, *WAOA 2013*, LNCS 8447, 24-36 (2014)
4. J. Bang-Jensen & G. Gutin: Orientations of graphs and digraphs in ‘Digraphs: Theory, Algorithms and applications’, *Springer, 2nd edition*, 417-472 (2009)
5. W. Ben-Ameur & M. Hadji: ‘Designing Steiner Networks with Unicyclic Connected Components: An Easy Problem’, *SIAM J. Discrete Math.* 24 (4), 1541-1557 (2010)

Table 1: Numerical results

graph	L_{MIP1}	U_{MIP1}	t_{MIP1}	ν_{LP2}	t_{LP2}	L_{MIP2}	U_{MIP2}	t_{MIP2}	$nb_{(4)}$	$nb_{(8,9)}$	$nb_{(6,7)}$	$ V $	$ E $	δ_G
K_{10}	5	5	25.17	5	0.01	5	5	0.03	4	2	0	10	45	9
K_{50}	full memory			25	1.56	25	25	3.27	24	12	0	50	1225	49
K_{150}				75	143.37	75	75	288.36	74	52	0	150	11175	149
K_{200}				99	427.48	99	99	859.44	100	75	0	200	19900	199
K_{225}				150	603.84	112	147	296.16	74	46	0	225	25200	224
G_4^4	4	4	360.95	4	0.04	4	4	0.1	0	0	0	256	768	4
G_5^3	2	5	900	5	17.49	5	5	37.16	0	0	22	243	810	5
G_5^4	full memory			4	0.12	4	4	0.38	0	0	0	625	2000	4
$G_{5,10}^2$	2	2	68.34	2	0.28	2	2	0.62	0	0	3	50	85	2
$G_{5,15}^2$	2	2	330.87	2	0.75	2	2	1.64	0	0	6	75	130	2
$G_{10,10}^2$	1	2	900	2	0.77	2	2	1.76	0	0	5	100	180	2
$G_{10,100}^2$	full memory			2	226.76	2	2	455.37	0	0	21	1000	1890	2
$G_{20,50}^2$				2	136.02	2	2	274	0	0	13	1000	1930	2
tG_4^3	6	6	10.5	6	0.05	6	6	0.12	0	0	0	64	192	6
tG_5^4	2	6	900	4	83.75	4	4	204.46	2	0	190	125	375	6
tG_4^4	full memory			8	0.74	8	8	2.28	0	0	0	256	1024	8
tG_4^5				10	17.4	10	10	46.96	0	0	0	1024	5120	10
tG_5^5				9	600.04	4	7	299.97	1	0	0	3125	15625	10
$tG_{5,10}^2$	2	2	44.58	2	0.02	2	2	0.05	2	0	0	50	100	4
$tG_{5,15}^2$	2	2	608.45	2	0.04	2	2	0.08	2	0	0	75	150	4
$tG_{10,10}^2$	4	4	130.5	4	0.09	4	4	0.2	0	0	0	100	200	4
$tG_{25,50}^2$	full memory			2	14.22	2	2	28.69	2	0	0	1250	2500	4
$tG_{25,100}^2$				2	50.12	2	2	100.77	2	0	0	2500	5000	4
$tG_{50,75}^2$				2	215.02	2	2	430.49	2	0	0	3750	7500	4
$P_{24,51}$	2	2	0.38	2	0.39	2	2	0.83	0	7	6	24	51	2
$P_{20,54}$	3	3	2.92	3	0.2	3	3	0.46	0	24	4	20	54	3
$P_{25,69}$	3	3	14.67	3	0.32	3	3	0.72	0	14	3	25	69	3
$R_{10,33}$	3	3	6.91	4	0.22	3	3	0.56	1	11	3	10	33	5
$R_{10,36}$	3	3	277.52	4	0.02	3	3	0.31	0	11	0	10	36	4
$R_{10,40}$	4	4	449.59	5	0.02	4	4	0.25	2	35	0	10	40	7
$R_{15,52}$	4	4	25.57	4	0.11	4	4	0.27	1	7	1	15	52	5
$R_{40,156}$	3	3	374.33	3	0.05	3	3	0.5	0	0	0	40	156	3
$R_{15,78}$	5	7	900	7	0.05	5	5	35.37	1	60	0	15	78	8
$R_{20,95}$	5	6	900	6	0.61	5	5	11.66	1	40	9	20	95	7
$R_{40,234}$	4	7	900	7	0.38	7	7	71.13	0	6	2	40	234	7
$R_{30,217}$	6	10	900	10	0.41	8	9	899.9	0	31	4	30	217	10
$R_{40,390}$	7	14	900	14	1.16	9	13	899.71	0	73	6	40	390	14
$R_{75,693}$	5	9	900	9	0.50	9	9	6.36	0	5	1	75	693	9
$R_{50,918}$	full memory			29	0.49	17	28	899.88	0	588	0	50	918	29
$R_{75,1387}$				26	0.55	16	26	899.86	0	152	0	75	1387	26

6. T. Biedl, T. Chan, Y. Ganjali, M. Hajiaghayi & D. R. Wood: 'Balanced vertex-orderings of graphs', *Discrete Applied Mathematics* 48 (1), 27-48 (2005)
7. M. Chrobak & D. Eppstein: 'Planar orientations with low out-degree and compaction of adjacency matrices', *Theoretical Computer Sciences* 86, 243-266 (1991)
8. F. Chung, M. Garey & R. Tarjan: 'Strongly connected orientations of mixed multigraphs', *Networks* 15, 477-484 (1985)
9. V. Chvátal & C. Thomassen: 'Distances in orientation of graphs', *Journal of Combinatorial Theory, series B* 24, 61-75 (1978)
10. M. Degraaf & A. Schrijver: 'Grid Minors of Graphs on the Torus', *Journal of Combinatorial Theory, series B* 61, 57-62 (1994)
11. R. Diestel: 'Graph Theory, 4th edition', Springer (2010)
12. F. Fomin, M. Matamala & I. Rapaport: 'Complexity of approximating the oriented diameter of chordal graphs', *Journal of Graph Theory* 45 (4), 255-269 (2004)
13. L.R. Ford & D.R. Fulkerson: 'Flows in networks', *Princeton University Press, Princeton, NJ* (1962)
14. A. Frank & A. Gyárfás: 'How to orient the edges of a graph?', *Colloquia Mathematica Societatis János Bolyai* 18, 353-364 (1976)
15. F. Harary, J. Krarup, and A. Schwenk: 'Graphs suppressible to an edge', *Canadian Mathematical Bulletin* 15, 201204 (1971)
16. J. Edmonds & E.L. Johnson: 'Matching, Euler tours and the Chinese postman problem', *Mathematical Programming* 5, 88124 (1973)
17. J. Kára, J. Kratochvíl & D. R. Wood: 'On the complexity of the balanced vertex ordering problem', *Proceedings of COCOON2005, LNCS 3595*, 849-858 (2005)
18. H.G. Landau: 'On dominance relations and the structure of animal societies III. The condition for a score structure' *The Bulletin of Mathematical Biophysics* 15, 143148 (1953)
19. D. Mubayi, T.G. Will & D.B. West: 'Realizing Degree Imbalances in Directed Graphs', *Discrete Mathematics* 239(173), 147-153 (2001)
20. C. Nash-Williams: 'On orientations, connectivity and odd vertex pairings in finite graphs', *Canadian Journal of Mathematics* 12, 555-567 (1960)
21. H. Robbins: 'A theorem on graphs with an application to a problem of traffic control', *American Mathematical Monthly* 46, 281-283 (1939)
22. T. J. Schaefer: 'The complexity of satisfiability problems', *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, 216-226 (1978)

Appendix A Proof of Lemma 6

Let φ be a not-all-equal at most 3-SAT formula with n variables $\{x_1, \dots, x_n\}$ and m clauses $\{c_1, \dots, c_m\}$ and for all $i \in \llbracket 1, n \rrbracket$, let $k_i \in \mathbb{N}$ be the number of occurrences of x_i in φ . We assume that there is at least one variable x_i that has at least 4 occurrences in φ (otherwise φ is already a not-all-equal at most 3-SAT(3V) formula) and we will build from φ a not-all-equal at most 3-SAT(3V) φ' such that φ and φ' are equisatisfiable as follows.

- For all $i \in \llbracket 1, n \rrbracket$, if $k_i \geq 4$ then we introduce k_i new variables $\{x_i^1, \dots, x_i^{k_i}\}$ and for $l \in \llbracket 1, k_i \rrbracket$ we replace the l -th occurrence of x_i in φ with x_i^l .
- For all $i \in \llbracket 1, n \rrbracket$, if $k_i \geq 4$ then we add k_i new clauses $\{c_{x_i}^1, \dots, c_{x_i}^{k_i}\}$ where for $l \in \llbracket 1, k_i - 1 \rrbracket$, $c_{x_i}^l = (x_i^l \vee \neg x_i^{l+1})$ and $c_{x_i}^{k_i} = (x_i^{k_i} \vee \neg x_i^1)$.

Suppose there exists an assignment $v : \{x_1, \dots, x_n\} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ of $x_1, \dots, x_n$ satisfying φ . Then

$$v' : \begin{aligned} &x_i \mapsto v(x_i) \quad \forall i \in \llbracket 1, n \rrbracket \text{ s.t. } k_i \leq 3; \\ &x_i^l \mapsto v(x_i) \quad \forall i \in \llbracket 1, n \rrbracket \text{ s.t. } k_i \geq 4 \text{ and } \forall l \in \llbracket 1, k_i \rrbracket; \end{aligned}$$

is an assignment of the variables x_i and x_i^l satisfying φ' for

- $\forall j \in \llbracket 1, m \rrbracket$, the values of the literals of c_j w.r.t. v and v' are piecewise equal so $v'(c_j) = v(c_j) = \text{TRUE}$ and v' is not-all-equal for c_j as well as v is;
- $\forall i \in \llbracket 1, n \rrbracket$ s.t. $k_i \geq 4$, $\forall l \in \llbracket 1, k_i - 1 \rrbracket$, $v'(x_i^l) = v'(x_i^{l+1}) = v(x_i)$ and $v'(x_i^{k_i}) = v'(x_i^1) = v(x_i)$ so we directly have $\forall l \in \llbracket 1, k_i - 1 \rrbracket$, $v'(c_{x_i}^l) = \text{TRUE}$ and $v'(c_{x_i}^{k_i}) = \text{TRUE}$ and v' is not-all-equal for each of these clauses since they all consist of two literals having opposite values w.r.t. v' .

As an example, for a formula

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee x_3),$$

where x_1 occurs five times and x_3 four so we add nine new variables $x_1^1, x_1^2, x_1^3, x_1^4, x_1^5, x_3^1, x_3^2, x_3^3$ and x_3^4 and nine new clauses:

$$\begin{aligned} \varphi' = &(x_1^1 \vee \neg x_2 \vee x_3^1) \wedge (\neg x_1^2 \vee \neg x_3^2 \vee x_4) \wedge (x_1^3 \vee \neg x_2) \wedge (\neg x_1^4 \vee \neg x_3^3 \vee \neg x_4) \wedge (x_1^5 \vee x_3^4) \\ &\wedge (x_1^1 \vee \neg x_1^2) \wedge (x_1^2 \vee \neg x_1^3) \wedge (x_1^3 \vee \neg x_1^4) \wedge (x_1^4 \vee \neg x_1^5) \wedge (x_1^5 \vee \neg x_1^1) \\ &\wedge (x_3^1 \vee \neg x_3^2) \wedge (x_3^2 \vee \neg x_3^3) \wedge (x_3^3 \vee \neg x_3^4) \wedge (x_3^4 \vee \neg x_3^1). \end{aligned}$$

Now suppose there exists an assignment v' of the x_i and x_i^l satisfying φ' and let $i \in \llbracket 1, n \rrbracket$ such that $k_i \geq 4$. If we take a look at the clauses $c_{x_i}^1, \dots, c_{x_i}^{k_i}$, we notice that if $v'(x_i^1) = \text{FALSE}$ then for $c_{x_i}^1$ to be satisfied, $v'(\neg x_i^2) = \text{TRUE}$, i.e. $v'(x_i^2) = \text{FALSE}$, then for $c_{x_i}^2$ to be satisfied, $v'(\neg x_i^3) = \text{TRUE}$...etc. Repeating this argument, we obtain that if $v'(x_i^1) = \text{FALSE}$ then $v'(x_i^1) = v'(x_i^2) = \dots = v'(x_i^{k_i}) = \text{FALSE}$. Similarly, if $v'(x_i^{k_i}) = \text{TRUE}$ then

for $c_{x_i}^{k_i}$ to be satisfied, $v'(\neg x_i^{k_i-1}) = \text{FALSE}$, i.e. $v'(x_i^{k_i-1}) = \text{TRUE}$, then for $c_{x_i}^{k_i-1}$ to be satisfied, $v'(\neg x_i^{k_i-2}) = \text{FALSE}$...etc. Hence if $v'(x_i^{k_i}) = \text{TRUE}$ then $v'(x_i^{k_i}) = v'(x_i^{k_i-1}) = \dots = v'(x_i^1) = \text{TRUE}$. This yields that

$$\forall i \in \llbracket 1, n \rrbracket \text{ s.t. } k_i \geq 4, v'(x_i^1) = v'(x_i^2) = \dots = v'(x_i^{k_i}).$$

Hence for all $i \in \llbracket 1, n \rrbracket$ such that $k_i \geq 4$, we can replace $x_i^1, \dots, x_i^{k_i}$ by a unique variable x_i and doing so the clauses $c_{x_i}^1, \dots, c_{x_i}^{k_i}$ become trivial and can be removed and only φ remains. So the following assignment of $x_1, \dots, x_n$:

$$v : \begin{array}{ll} x_i \mapsto v'(x_i) & \forall i \in \llbracket 1, n \rrbracket \text{ s.t. } k_i \leq 3; \\ x_i \mapsto v'(x_i^1) & \forall i \in \llbracket 1, n \rrbracket \text{ s.t. } k_i \geq 4; \end{array}$$

satisfies φ . We have just shown that φ and φ' are equisatisfiable. □