Discrete Mathematics and Theoretical Computer Science DMTCS vol. **9**, 2007, 193–202

On the complexity of the balanced vertex ordering problem[†]

Jan Kára[1‡], Jan Kratochvíl[1] and David R. Wood[2§]

[1]*Department of Applied Mathematics*
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
{kara,honza}@kam.mff.cuni.cz

[2]*Departament de Matemàtica Aplicada II*
Universitat Politècnica de Catalunya, Barcelona, Spain
david.wood@upc.es

received April 25, 2006, *revised August 27, 2007*, *accepted August 28, 2007*.

We consider the problem of finding a balanced ordering of the vertices of a graph. More precisely, we want to minimise the sum, taken over all vertices v, of the difference between the number of neighbours to the left and right of v. This problem, which has applications in graph drawing, was recently introduced by Biedl et al. [Discrete Applied Math. 148:27–48, 2005]. They proved that the problem is solvable in polynomial time for graphs with maximum degree three, but $\mathcal{NP}$-hard for graphs with maximum degree six. One of our main results is to close the gap in these results, by proving $\mathcal{NP}$-hardness for graphs with maximum degree four. Furthermore, we prove that the problem remains $\mathcal{NP}$-hard for planar graphs with maximum degree four and for 5-regular graphs. On the other hand, we introduce a polynomial time algorithm that determines whether there is a vertex ordering with total imbalance smaller than a fixed constant, and a polynomial time algorithm that determines whether a given multigraph with even degrees has an 'almost balanced' ordering.

Keywords: graph, graph drawing, vertex ordering, complexity

Mathematics Subject Classification: 05C62 (graph representations)

1 Introduction

A number of algorithms for graph drawing use a 'balanced' ordering of the vertices of the graph as a starting point [3, 4, 6, 8, 9]. Here balanced means that the neighbours of each vertex v are evenly distributed to the left and right of v. The problem of determining such an ordering was recently studied by Biedl et al. [1]. We solve a number of open problems from [1] and study a few other related problems.

Let $G = (V, E)$ be a multigraph without loops. An *ordering* of G is a bijection $\sigma : V \to \{1, \dots, |V|\}$. For $u, v \in V$ with $\sigma(u) < \sigma(v)$, we say that u is *to the left* of v and that v is *to the right* of u. The *imbalance* of $v \in V$ in σ, denoted by $B_\sigma(v)$, is

$$\big||\{e \in E : e = \{u, v\}, \sigma(u) < \sigma(v)\}| - |\{e \in E : e = \{u, v\}, \sigma(u) > \sigma(v)\}|\big|.$$

[†] This is a revised and extended version of the authors' conference paper [5].

[‡] Supported by project 1M0021620808 of the Ministry of Education of the Czech Republic

[§] Research supported by a Marie Curie Fellowship from the European Commission under contract MEIF-CT-2006-023865, and by the projects MEC MTM2006-01267 and DURSI 2005SGR00692. Research completed at the Department of Applied Mathematics and the Institute for Theoretical Computer Science, Charles University, Prague, Czech Republic. Supported by project LN00A056 and 1M0021620808 of the Ministry of Education of the Czech Republic, and by the European Union Research Training Network COMBSTRU.

When the ordering σ is clear from the context we simply write $B(v)$ instead of $B_\sigma(v)$. The *imbalance* of ordering σ, denoted by $B_\sigma(G)$, is $\sum_{v \in V} B_\sigma(v)$. The minimum value of $B_\sigma(G)$, taken over all orderings σ of G, is denoted by $M(G)$. An ordering with imbalance $M(G)$ is called *minimum*. The following two facts hold for every ordering:

- Every vertex of odd degree has imbalance at least one.
- The two vertices at the beginning and at the end of the ordering have imbalance equal to their degrees.

These two facts imply the following lower bound on the imbalance of an ordering. Let $\text{odd}(A)$ denote the number of odd degree vertices among the vertices of $A \subseteq V$. Let $(d_1, \dots, d_n)$ be the sequence of vertex degrees of G, where $d_i \le d_{i+1}$ for all $i \in \{1, 2, \dots, n-1\}$. Then

$$B_\sigma(G) \ge \text{odd}(V) - (d_1 \bmod 2) - (d_2 \bmod 2) + d_1 + d_2.$$

An ordering σ is *perfect* if the above inequality holds with equality. PERFECT ORDERING is the decision problem whether a given multigraph G has a perfect ordering. This problem is clearly in $\mathcal{NP}$.

Biedl et al. [1] gave a polynomial time algorithm to compute a minimum ordering of graphs with maximum degree at most three. On the other hand, they proved that it is $\mathcal{NP}$-hard to compute a minimum ordering of a (bipartite) graph with maximum degree six.

One of the main results of this paper is to close the above gap in the complexity of the balanced ordering problem with respect to the maximum degree of the graph. In particular, we prove that the PERFECT ORDERING problem is $\mathcal{NP}$-complete for simple graphs with maximum degree four.

In fact, the NP-completeness result even holds if we additionally restrict the graphs to be planar. This is of particular interest since a number of algorithms for producing orthogonal drawings of planar graphs with maximum degree four start with a balanced ordering of the vertices [2, 6]. We answer this question in the negative by proving that the PERFECT ORDERING problem is $\mathcal{NP}$-complete for planar simple graphs with maximum degree four.

Our third $\mathcal{NP}$-hardness result states that finding an ordering with minimum imbalance is $\mathcal{NP}$-hard for 5-regular simple graphs. All of these $\mathcal{NP}$-hardness results for ordering problems are presented in Section 3. The proofs are based on reductions from various satisfiability problems. Section 2 contains several $\mathcal{NP}$-completeness results for used satisfiability problems. While the complexity of most of these satisfiability problems follows from a general result by Schaefer [7], we believe that our proofs are simpler and the result for PLANAR 2–IN–4SAT is of independent interest.

In Section 4 we present our positive complexity results. In particular, we describe a polynomial time algorithm that determines whether a given graph has an ordering with at most k imbalanced vertices for any constant k. This algorithm has several interesting corollaries. For example, the PERFECT ORDERING problem can be solved in polynomial time for a multigraph in which all the vertices have even degrees (in particular, for 4-regular multigraphs).

2 $\mathcal{NP}$-Hardness of Satisfiability Problems

In this section we prove several $\mathcal{NP}$-hardness results about various satisfiability problems. Note that the results in this section could be also achieved by verifying conditions of a general theorem of Schaefer [7], but we feel that our proofs are simpler. First we introduce several basic definitions about satisfiability. Throughout this paper, formulae are considered to be in a *conjunctive normal form*. That is, each formula φ is a conjunction of some m clauses $c_1 \wedge c_2 \wedge \cdots \wedge c_m$ where each *clause* c_i is a disjunction of n_i literals $l_1^i \vee l_2^i \vee \cdots \vee l_{n_i}^i$ for all $i \in \{1, \dots, m\}$. A *literal* is either a variable or its negation. The *size* of a clause is the number of literals in it. Suppose φ is a formula with variables $x_1, \dots, x_n$. The *incidence graph* of φ is the bipartite graph with vertex set $\{c_1, \dots, c_m, x_1, \dots, x_n\}$, where $\{c_i, x_j\}$ is an edge if and only if the variable x_j occurs in the clause c_i. A *truth assignment* of a formula φ with variables $x_1, \dots, x_n$ is an arbitrary function $t : \{1, \dots, n\} \to \{0, 1\}$. The values 0 and 1 are also sometimes called *false* and *true* respectively. A truth assignment t is *satisfying* if there is at least one true literal in every clause. The formula φ is *satisfiable* if it has at least one satisfying truth assignment.

The decision problem asking whether a given formula φ is satisfiable is called SAT. If we assume that every clause in the given formula φ has size exactly three, then the decision problem asking whether φ is satisfiable is called 3SAT. Two common variants of 3SAT are *Monotone Not–All–Equal 3-Satisfiability* (MON–NAE–3SAT for short) and *1–in–3 Satisfiability* (1–IN–3SAT). Both these problems are defined on formulae in which each clause has size exactly three. Furthermore in MON–NAE–3SAT the formulae are without negations. A truth assignment t is *NAE satisfying* if each clause has at least one true and at least one false literal. t is called *1–in–3 satisfying* if each clause has exactly one true literal. The notions of *NAE satisfiable* and *1–in–3 satisfiable* formulae, and the corresponding decision problems are defined in the obvious way. It is well known that SAT, MON–NAE–3SAT, and 1–IN–3SAT are $\mathcal{NP}$-complete (see [7]).

We say that a formula φ for which all clauses have five literals is *2–or–3–in–5 satisfiable* if there exists a truth assignment such that in each clause either two or three literals are true. Let 2–OR–3–IN–5SAT denote the decision problem asking whether a given formula φ is 2–or–3–in–5 satisfiable.

Lemma 1 *The problem* 2–OR–3–IN-5SAT *is $\mathcal{NP}$-complete for formulae without negations.*

Proof: The problem clearly belongs to $\mathcal{NP}$. We prove $\mathcal{NP}$-completeness by a reduction from MON–NAE–3SAT, which is $\mathcal{NP}$-complete [7]. Suppose we are given a formula φ without negations. Create a formula φ' from φ by adding a new clause $c_0 = x \vee x \vee x \vee x' \vee x'$, and by substituting each clause c with clause $c \vee x \vee x'$ where x and x' are new variables. Given a NAE-satisfying truth assignment t for φ, a 2-or-3-in-5-satisfying truth assignment t' for φ' can be created by setting $t'(x) = 0$ and $t'(x') = 1$. Also, if t' is a 2–or–3–in–5 satisfying truth assignment for φ', then $t'(x) = \neg t'(x')$ by clause c_0. Thus, restricting t' to the variables of φ, we obtain a NAE-satisfying truth assignment for φ. $\square$

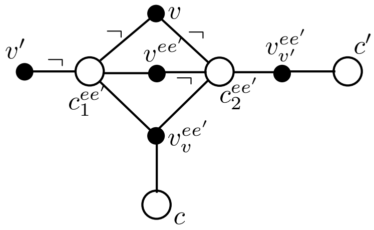
The next lemma uses the following version of the satisfiability problem. Let φ be a formula in which all clauses have four literals. A truth assignment t is *2–in–4 satisfying* if each clause in φ has exactly two true literals. φ is *2–in–4 satisfiable* if there exists a 2–in–4 satisfying truth assignment. 2–IN–4SAT is the decision problem asking whether a given formula φ is 2–in–4 satisfiable.

Lemma 2 *The problem* 2–IN–4SAT *is $\mathcal{NP}$-complete for formulae without negations.*

Proof: The problem is obviously in $\mathcal{NP}$. We prove its $\mathcal{NP}$-completeness by reduction from 1–IN–3SAT, which is $\mathcal{NP}$-complete for formulae without negations [7]. Let ψ be a formula given as an input for 1–IN–3SAT. Create a formula ψ' with a new variable v by adding v to each clause of ψ. We now show that ψ is 1–in–3 satisfiable if and only if ψ' is 2–in–4 satisfiable. If t is a 1–in–3 satisfying truth assignment for ψ, then by setting $t'(v) = 1$ and $t'(x) = t(x)$ for each variable $x \neq v$ of ψ', we obtain a 2–in–4 satisfying truth assignment for ψ'. Conversely, if t' is a 2–in–4 satisfying truth assignment for ψ', then either t' (in the case that $t'(v) = 1$) or $1 - t'$ (in the case that $t'(v) = 0$) restricted to variables of ψ is a 1–in–3 satisfying truth assignment for ψ. $\square$

Now we strengthen the result from the previous lemma.

Lemma 3 *The problem* 2–IN–4SAT *is $\mathcal{NP}$-complete for planar formulae without negations.*



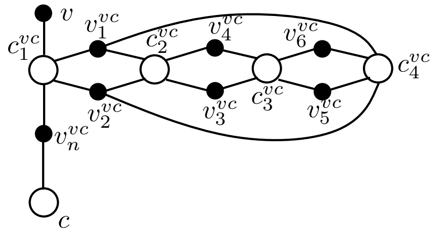


Fig. 1: The crossing gadget for two edges $\{v, c\}$ and $\{v', c'\}$ (left) and the negation gadget for a negative occurrence of variable v in clause c (right) from the proof of Lemma 3. Empty circles represent clauses, and full circles represent variables. The symbol $\neg$ marks a negative occurrence.

Proof: Suppose we have a formula φ with clauses of size four without negations. We now show that if the formula φ is not planar we can alter it in polynomial time so that the resulting formula φ' is planar and φ is 2–in–4 satisfiable if and only if φ' is 2–in–4 satisfiable. The formula φ' will contain some negations but we also define a planar formula ensuring $v = \neg v'$ for two of its variables v, v' and all 2–in–4 satisfying truth assignments. Hence by substitution of each edge representing the negative occurrence by this gadget we prove the lemma.

Let d be a drawing of the incidence graph of φ in the plane, such that any two edges cross at most once. We proceed by induction on the number of crossings in d. If there is no crossing, we are done. Now suppose there is some crossing and for all formulae having a drawing of their incidence graph with less crossings the lemma holds. Consider an edge $e = (v, c)$ and the crossing with some edge $e' = (v', c')$ closest to v on the edge e. Create a new formula ψ by adding three new variables $v^{ee'}, v^{ee'}_v, v^{ee'}_{v'}$ and two clauses $c^{ee'}_1 = \neg v \vee \neg v' \vee v^{ee'} \vee v^{ee'}_v, c^{ee'}_2 = \neg v \vee \neg v^{ee'} \vee v^{ee'}_{v'} \vee v^{ee'}_v$. Then substitute occurrences of v in c by $v^{ee'}_v$, and occurrences of v' in c' by $v^{ee'}_{v'}$. See Figure 1 for an example of a gadget for two crossing edges. Note that only edges $(v', c^{ee'}_1)$, $(v^{ee'}_v, c)$, and $(v^{ee'}_{v'}, c')$ of the gadget can be intersected by other edges of the incidence graph.

After the substitution we clearly obtain a formula with a drawing with one less crossing. It remains to show that ψ is 2–in–4 satisfiable if and only if φ is 2–in–4 satisfiable (we get the rest using the induction). Let t be a 2–in–4 satisfying truth assignment for φ. Setting $t'(x) = t(x)$ for all variables x of φ, $t'(v^{ee'}_v) = t(v)$ and $t'(v^{ee'}_{v'}) = t'(v^{ee'}) = t(v')$, we obtain a 2–in–4 satisfying truth assignment for ψ. The other implication can be seen as follows. Let t' be a 2–in–4 satisfying truth assignment for ψ. We set $t(x) = t'(x)$ for each variable x of φ. If we show that $t'(v) = t'(v^{ee'}_v)$ and $t'(v') = t'(v^{ee'}_{v'})$, we immediately get that t is a 2–in–4 satisfying truth assignment for φ. We analyse two cases (the other two follow by symmetry):

- $t'(v) = t'(v') = 1$: In this case, $c^{ee'}_1$ has already two literals set to zero and so $t'(v^{ee'}_v) = t'(v^{ee'}) = 1$. Now looking at $c^{ee'}_2$ we see that two of its literals are set to zero and one literal is set to one. Thus $t'(v^{ee'}_{v'}) = 1$.

- $t'(v) = \neg t'(v') = 1$: If $t'(v^{ee'}) = 1$, then $c^{ee'}_1$ has two literals set one and one literal set to zero. Thus $t'(v^{ee'}_v)$ must be zero. But then $c^{ee'}_2$ has three literals set to zero and we can conclude that this cannot be the case. Hence $t'(v^{ee'}) = 0$ and $t'(v^{ee'}_v) = 1$ to satisfy $c^{ee'}_1$. Moreover $c^{ee'}_2$ has two literals set to one and one literal set to zero showing that $t'(v^{ee'}_{v'}) = 0$.

Now it remains to show how to remove the negative occurrences from φ'. For each negative occurrence of variable v in clause c we add seven new variables $v^{vc}_n, v^{vc}_1, \ldots, v^{vc}_6$ and four new clauses $c^{vc}_0 = v \vee v^{vc}_n \vee v^{vc}_1 \vee v^{vc}_2$, $c^{vc}_1 = v^{vc}_1 \vee v^{vc}_2 \vee v^{vc}_3 \vee v^{vc}_4$, $c^{vc}_2 = v^{vc}_3 \vee v^{vc}_4 \vee v^{vc}_5 \vee v^{vc}_6$, $c^{vc}_3 = v^{vc}_1 \vee v^{vc}_2 \vee v^{vc}_5 \vee v^{vc}_6$. See Figure 1 for an example of a created gadget. We

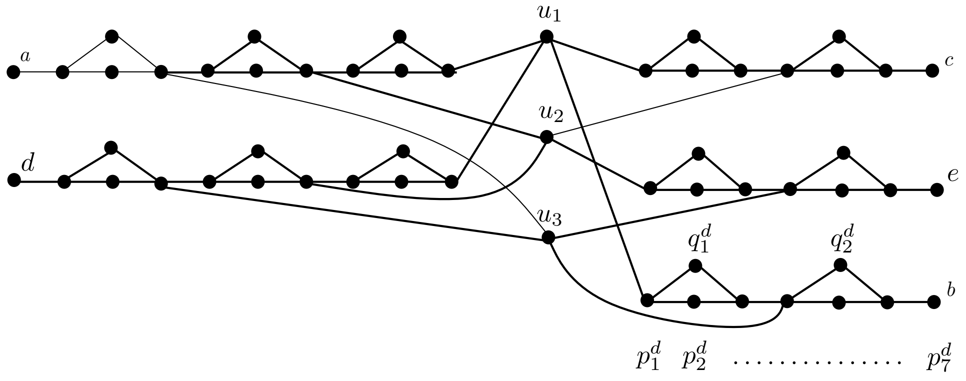


Fig. 2: A graph constructed for formula $(a \vee b \vee c \vee d) \wedge (a \vee c \vee d \vee e) \wedge (a \vee b \vee d \vee e)$ in the proof of Theorem 1. The three clauses have numbers $1, 2, 3$ in the picture.

also substitute the negative occurrence of v in c by a positive occurrence of v_n^{vc}. Let φ'' be the resulting formula. It is straightforward to check that $t''(v) = \neg t''(v_n^{vc})$ in any 2–in–4 satisfying truth assignment t'' of φ'' and by setting $t''(v_1^{vc}) = t''(v_3^{vc}) = t''(v_5^{vc}) = t'(v)$ and $t''(v_2^{vc}) = t_4''^{vc} = t_6''^{vc} = t_n''^{vc} = \neg t'(v)$ we get a 2–in–4 satisfying truth assignment t'' of φ'' from a 2–in–4 satisfying truth assignment t' of φ'. □

Note that if we allowed multiple occurrences of one variable in a clause in the previous lemma the negation gadget would become trivial and for our purposes such a weaker lemma would be sufficient. But we decided to prove the stronger version as we find the lemma of independent interest.

3 $\mathcal{NP}$-Hardness of Balanced Ordering Problems

In this section we prove several $\mathcal{NP}$-hardness results about balanced ordering problems.

Theorem 1 *The* PERFECT ORDERING *problem is $\mathcal{NP}$-complete for planar graphs with maximum degree four.*

Proof: $\mathcal{NP}$-hardness is proved by a reduction from 2–IN–4SAT for planar formulae without negations (see Lemma 3). Given a formula φ, create a graph G_φ with one vertex u_c for each clause c. For each variable v that occurs o_v times in φ, add a path on $3o_v + 1$ new vertices $p_1^v, \ldots, p_{3o_v+1}^v$ to G_φ, add o_v additional vertices $q_1^v, \ldots, q_{o_v}^v$, and connect $q_i^v, i \in \{1, \ldots, o_v\}$ with vertices p_{3i-2}^v and p_{3i}^v. The path with the additional vertices is called a *variable gadget*. Finally for each $i \in \{1, \ldots, o_v\}$, connect vertex p_{3i-2}^v of the path to u_c, where c is the clause corresponding to the i-th occurrence of the variable v. These edges are called *clause edges*. See Figure 2 for an example of this construction.

Observe that the maximum degree of G_φ is four. In particular, $\deg(u_c) = 4$, $\deg(q_i^v) = 2$ for all $i \in \{1, \ldots, o_v\}$, $\deg(p_{3i}^v) = 3$ for all $i \in \{1, \ldots, o_v\}$, $\deg(p_{3i-2}^v) = 4$ for all $i \in \{2, \ldots, o_v\}$, $\deg(p_{3i-1}^v) = 2$ for all $i \in \{1, \ldots, o_v\}$, $\deg(p_1^v) = 3$, and $\deg(p_{3o_v+1}^v) = 1$. Also note that the created graph is planar if the incidence graph of φ is planar.

We now prove that G_φ has a perfect ordering if and only if φ is 2–in–4 satisfiable. Suppose G_φ has a perfect linear ordering σ. For each variable v and for each $i \in \{1, \ldots, o_v\}$ the vertex p_{3i-1}^v has one neighbour to the left and one neighbour to the right in σ (since $\deg(p_{3i-1}^v) = 2$). Similarly, q_i^v has one neighbour to the left and one neighbour to the right in σ. Thus they must be placed between p_{3i-2}^v and p_{3i}^v. As p_{3i-1}^v and q_i^v are on one side (e.g., to the left) of vertex p_{3i-2}^v (p_{3i}^v) the other neighbours of the vertex must be on the other side. This implies that in σ, the path in each variable gadget is in the order given by its numbering or inverse numbering, and all the clause edges (the edges with exactly one endpoint in the variable gadget) have a clause vertex on the same end (for example, the left

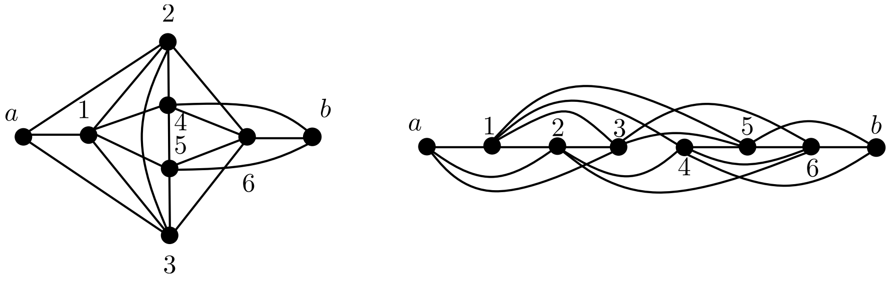


Fig. 3: The triple edge gadget from Lemma 4 (left) and its natural ordering (right).

end of each clause edge is a vertex of a path). If the path in the gadget for variable v is ordered according to its numbering, then set $t(v) := 0$. Otherwise set $t(v) := 1$. This truth assignment is 2–in–4 satisfying because each clause vertex has two neighbours on each side.

For a given truth assignment t we can analogously construct a perfect linear ordering. First place each variable gadget corresponding to a variable with $t(v) = 0$ with the path placed according to the inverse ordering, and put each vertex q_i^v immediately after vertex $p_{3i-1}^v, i \in \{1, \ldots, o_v\}$. Then place vertices u_c in an arbitrary order and finally the variable gadgets corresponding to variables with $t(v) = 1$ with the paths ordered according to the numbering and vertices q_i^v placed immediately after the vertex p_{3i-2}^v. □

The following two technical lemmas will be used later for removing parallel edges from a multigraph without changing an ordering with minimum imbalance.

Lemma 4 *Let G be the multigraph drawn in Figure 3 with two parallel edges added between the vertices a and b. Then there exists a minimum ordering of G such that a is the leftmost and b the rightmost vertex. Such an ordering is called a* natural ordering *of G.*

Proof: The ordering $a, 1, 2, 3, 4, 5, 6, b$ has imbalance 20. We claim that there is no ordering with smaller imbalance. Let $v_1, \ldots, v_8$ be some ordering of the vertices. We distinguish two cases:

1. There are two parallel edges between v_1 and v_2 (or symmetrically between v_7 and v_8). Because there is only one double-edge in our graph, $v_1 = a$ and $v_2 = b$ (the case $v_1 = b$ and $v_2 = a$ is the same) and we also know that there is at most one neighbour to the right of v_7. Since each vertex of G is connected to exactly one of a and b, there is only one neighbour to the left of v_3. Because the other vertices have imbalance at least one (they have odd degrees), the imbalance of the ordering is at least $B(v_1) + B(v_3) + B(v_7) + B(v_8) + 4 = 5 + 3 + 3 + 5 + 4 = 20$.

2. There is no parallel edge between v_1 and v_2 nor between v_7 and v_8. In this case there is at most one neighbour to the left of v_2, and at most one neighbour to the right of v_7. Hence the total imbalance is at least $B(v_1) + B(v_2) + B(v_7) + B(v_8) + 4 = 5 + 3 + 3 + 5 + 4 = 20$.

□

Lemma 5 *Let G be a 5-regular multigraph and let c be the number of triple-edges in G. Let G' be the graph obtained from G by replacing each triple-edge of G with endpoints a and b by the* triple-edge gadget *in Figure 3. The vertices a and b of the gadget are identified with the original end-vertices of the triple-edge. Then $M(G) = M(G') - 10 \cdot c$.*

Proof: Given an ordering of G with imbalance i, we can create an ordering of G' with imbalance $i + 10 \cdot c$ by inserting, for each triple-edge ab, the vertices $1, \ldots, 6$ from the gadget in Figure 3 between a and b and in this order (the imbalance of a and b does not change by the substitution). Thus $M(G) \geq M(G') - 10 \cdot c$. On the other hand, if we have a minimum ordering of G' with imbalance i', below we show that by changing the given ordering so that each gadget is in its natural ordering, we obtain an ordering with imbalance $\leq i'$. Hence, the new ordering has imbalance i', from the minimality of i'. By substituting each gadget with the triple-edge we obtain an ordering of G with imbalance $i' - 10 \cdot c$, proving that $M(G) \leq M(G') - 10 \cdot c$.

Suppose we have a triple-edge gadget in G' between vertices a and b, with a to the left of b. Each of the vertices has two neighbours u_1, u_2 (u'_1, u'_2 respectively) outside of the gadget. Let u_1 be to the left of u_2, and let u'_1 be to the left of u'_2. If u_1, u_2 are both to the right of a and u'_1, u'_2 are both to the left of b, then we are in the situation described by Lemma 4 (only the two parallel edges from the Lemma are now substituted by the edges to the vertices u_1, u_2, u'_1, u'_2), and we can conclude that after reordering the vertices of the gadget (so that vertices a and b retain their ordering with respect to all other vertices of the graph), we obtain an ordering of G' with less or equal imbalance. If u_1 is to the left and u_2 to the right of a (u'_1 and u'_2 are both still to the left of b), then the imbalance of the gadget in the natural ordering in this situation is 18 and any ordering of the gadget cannot have smaller imbalance, as in the new situation only the ordering of vertices of one of the edges incident with the gadget has changed. Hence if we reorder the vertices of the gadget we obtain an ordering of G' with less or equal imbalance. Now consider the general situation. We have j neighbours of a outside of the gadget and j' neighbours of b outside of the gadget ($0 \leq j, j' \leq 2$) on the other side of a (b respectively). In the original situation the natural ordering of the gadget has imbalance $20 - 2 \cdot (j + j')$. There cannot be an ordering of the gadget with smaller imbalance because in the situation only the ordering of vertices of the $j + j'$ edges incident with the gadget changed. Hence we can conclude that reordering the gadget into the natural ordering cannot increase the imbalance of the graph regardless of the ordering of v, u_1, u_2 and v', u'_1, u'_2. □

For the next reduction we use the 2–OR–3–IN-5SAT problem which we proved to be $\mathcal{NP}$-complete in Section 2.

Theorem 2 *The* PERFECT ORDERING *problem is $\mathcal{NP}$-complete for 5-regular multigraphs.*

Proof: We prove $\mathcal{NP}$-hardness by a reduction from 2–OR–3–IN-5SAT for formulae without negations. Suppose that we are given a formula φ without negations and with all clauses of size five. Moreover, assume that each variable occurs in at least two different clauses in the formula. We can make a formula satisfy this condition by adding satisfied clauses of type $x \vee x \vee x \vee \neg x \vee \neg x$. Now create the following multigraph G from φ. For each clause c add a new vertex v_c to G. For each variable x that occurs o_x times in φ, add a new path (called a *variable path*) with $2o_x - 2$ vertices $v_1^x, \ldots, v_{2o_x-2}^x$, where edges $v_{2i-1}^x v_{2i}^x, 1 \leq i \leq o_x - 1$, are triple-edges. Connect vertex $v_{2i}^x, 1 \leq i \leq o_x - 1$, of the path to the vertex corresponding to the clause with i-th occurrence of x. Furthermore, connect vertex $v_{2o_x-2}^x$ to the vertex corresponding to the clause with the o_x-th occurrence of x (because x was in at least two different clauses we can without loss of generality assume that no parallel edges are created). Connect each vertex $v_{2i-1}^x, 1 \leq i \leq o_x - 1$, to the new vertex p_i^x, and connect each vertex v_1^x to the new vertex p_0^x. Now the only vertices which have degree other than five are in the set $P = \{p_j^x : x \text{ is a variable}, 0 \leq j \leq o_x - 1\}$ and these have degree one. By running the following procedure two times for the set P, all the vertices will have degree five.

$n := |P|$
Arbitrarily number the vertices in P by $1, \ldots, n$.
`while` $|P| \geq 3$ `do`
  Let u_i, u_j, u_k be three vertices in P.
  $P := P \setminus \{u_i, u_j, u_k\} \cup \{u_{n+1}, u_{n+2}\}$
  Add a complete bipartite graph on u_i, u_j, u_k and u_{n+1}, u_{n+2} to G.

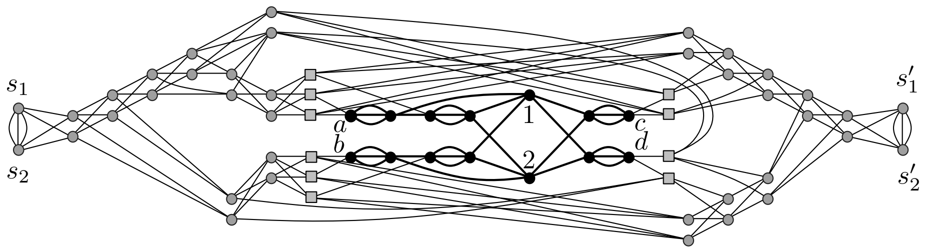


Fig. 4: A graph constructed for formula $(a \vee a \vee b \vee c \vee d) \wedge (a \vee b \vee b \vee c \vee d)$ in the proof of Theorem 2. Clause vertices are marked 1 and 2. Clause vertices and variable paths are drawn in black colour, vertices p_i^x are drawn as squares in gray colour and vertices added by the procedure are drawn as circles in gray colour (vertices added in the first run are drawn in the left and vertices added in the second run are drawn in the right).

  $n := n + 2$
`end`
`//` *Now* $P = \{u_i, u_j\}$
Add to G a complete bipartite graph on u_i, u_j and two new vertices s_1, s_2.
Add a triple-edge $s_1 s_2$ to G.

See Figure 4 for an example of the construction. Let n_0 denote the value of n at the beginning of the procedure and n_1 the value of n at the end of the procedure. It is easy to check that G is 5-regular. We now show that G has a perfect ordering if and only if φ was 2-or-3-in-5 satisfiable. Suppose we have a perfect ordering of G. It holds that $B(s_1) + B(s_2) > 2$ for every ordering. Since the ordering is perfect, we can move s_1, s_2 to the beginning without increasing the imbalance. By a similar argument, the ordering ends with vertices s'_2, s'_1, where s'_1 and s'_2 are the vertices added at the end of the second run of the procedure on P. Because all other vertices are balanced we know that every variable path is either in its natural ordering or reversed. Moreover all edges between the variable path and clauses have clause vertices to the right (or to the left in the reversed case). Because all clause vertices are balanced we get a 2-or-3-in-5 satisfying truth assignment of φ by assigning $t(x) = 0$ to the variables whose path is naturally ordered and $t(x) = 1$ to the variables whose path is reversed. For the converse, suppose we have a 2-or-3-in-5 satisfying truth assignment t of φ. First we place vertices $s_1, s_2, u_{n_1}, \ldots, u_{n_0+1}$ added in the first run. We continue by placing vertices p_j^x where x is a variable with $t(x) = 0$ and $0 \leq j \leq o_x - 1$. Then we place variable paths for variables x such that $t(x) = 0$ in their natural ordering and after them the clause vertices. We finish by symmetrically placing the rest of paths and vertices added in the second run. It is straightforward to check that this ordering is perfect. □

Corollary 1 *It is $\mathcal{NP}$-hard to find a minimum ordering for 5-regular graphs.*

Proof: Construct the multigraph G as in the reduction in the proof of Theorem 2. Say G has c triple edges. Construct G' from G by substituting each triple-edge by a triple-edge gadget. Observe that G' remains 5-regular and is a simple graph. From Lemma 5 we know that orderings of G' with imbalance $|V| + 10 \cdot c$ correspond to perfect orderings of G. This proves $\mathcal{NP}$-hardness of finding the ordering with such imbalance. Hence finding a minimum ordering for 5-regular graphs is $\mathcal{NP}$-hard. □

4 Algorithm

In this section we present an algorithm that determines in polynomial time whether a given multigraph has an ordering with constant imbalance. First we introduce a key lemma.

Lemma 6 *Let vertices $v_1, \ldots, v_k$ be given. There is an $O(n+m)$ time algorithm to test whether a multigraph G with n vertices and m edges has an ordering in which $v_1, \ldots, v_k$ are the only imbalanced vertices and appear in this order. Moreover all orderings with this property have the same imbalance.*

Proof: Let $imbalanced$ denote the ordered list of vertices $(v_1, \ldots, v_k)$. The vertices not in the list $imbalanced$ are called *balanced*. The algorithm works as follows: First we check that all odd-degree vertices are in the $imbalanced$ list. If not, then we can reject since every odd-degree vertex must be imbalanced. Now assume that all balanced vertices have even degrees. Then start building an ordering σ from left to right. We append to σ those vertices that have not been placed yet and have half of their neighbours already placed. Such vertices are called *saturated* and are stored in the set $saturated$. Because saturated vertices are balanced each saturated vertex must be placed before any of its unplaced neighbours. In particular saturated vertices must form an independent set. Hence we cannot make a mistake when placing any saturated vertices. If there is no saturated vertex, the vertex which is placed next will be imbalanced and hence it must be the first unused vertex from the $imbalanced$ list. The described algorithm can be clearly implemented in $O(m+n)$ in a way analogous to the standard implementation of topological sorting. It remains to prove that it is not better to place some vertices from the $imbalanced$ list while there are still some saturated vertices. If the order of vertices of any edge does not change then we have an equivalent ordering. Otherwise it does change, in which case some balanced vertex becomes imbalanced (as the order of vertices in an edge can change only for the edges which contain at least one balanced vertex) and we must reject. Thus we also see that all orderings satisfying the conditions from the statement of the lemma have the same imbalance. □

The following theorem is a consequence of Lemma 6.

Theorem 3 *There is an algorithm that, given an n-vertex m-edge multigraph G, computes a minimum ordering of G with at most k imbalanced vertices (or answers that there is no such ordering) in time $O(n^k \cdot (m+n))$.*

Proof: The algorithm is simple: just try all the possible choices of k imbalanced vertices and their orderings. For each such choice run the procedure from Lemma 6 and select the ordering with minimum imbalance from those orderings. There are $O(n^k)$ k-tuples of imbalanced vertices, and for each such k-tuple, by Lemma 6, we can check in $O(m+n)$ time whether there is an ordering with the chosen vertices imbalanced, and if so, compute the imbalance of the ordering. □

Corollary 2 *There is a polynomial time algorithm to determine whether a given multigraph G has an ordering with imbalance less than a fixed constant c.*

Proof: Apply the algorithm from Theorem 3 with $k = c - 1$. If the algorithm rejects the multigraph or produces an ordering with imbalance greater than c, then the graph does not have an ordering with imbalance less than c (because any ordering with imbalance less than c must have at most $c-1$ imbalanced vertices). Otherwise the algorithm outputs some ordering with imbalance less than c, and we are done. □

Corollary 3 *The* PERFECT ORDERING *problem is solvable in time $O(n^2(n+m))$ for any n-vertex m-edge multigraph with all vertices of even degree.*

Proof: Apply the algorithm from Theorem 3 with $k = 2$, and then check whether the achieved imbalance is equal to that required by the PERFECT ORDERING problem. A perfect ordering of a multigraph with even degrees must have exactly two imbalanced vertices (assuming there is at least one edge). □

5 Conclusion and Open Problems

In this paper we have considered the problems of checking the existence of a perfect ordering for planar graphs with maximum degree four and 5-regular multigraphs. Both these problems were shown to be $\mathcal{NP}$-complete, thus answering a number of questions raised by Biedl et al. [1]. We have also established that it is $\mathcal{NP}$-hard to find an ordering with minimum imbalance for 5-regular simple graphs. We have also introduced an algorithm for determining an ordering with imbalance smaller than k with the time complexity $O(n^k(n+m))$. It would be interesting to obtain a fixed-parameter-tractable (FPT) algorithm for this problem — i.e., an algorithm with the time complexity $O(f(k) \cdot poly(n,m))$ — as one cannot hope for a polynomial solution with k in the input unless $\mathcal{P} = \mathcal{NP}$.

Acknowledgements

The authors would like to thank the anonymous referees for useful comments.

References

[1] Therese Biedl, Timothy Chan, Yashar Ganjali, Mohammad Taghi Hajiaghayi and David R. Wood, Balanced vertex-orderings of graphs, *Discrete Applied Mathematics* 148(1), pp. 27–48, 2005.

[2] Therese Biedl and Goos Kant, A better heuristic for orthogonal graph drawings, *Computational Geometry: Theory and Applications* 9(3), pp. 159–180, 1998.

[3] Goos Kant, Drawing planar graphs using the canonical ordering, *Algorithmica* 16, pp. 4–32, 1996.

[4] Goos Kant and Xin He, Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems, *Theoretical Computer Science* 172(1–2), pp. 175–193, 1997.

[5] Jan Kára, Jan Kratochvíl and David R. Wood, On the complexity of the balanced vertex ordering problem, *Proc. of 11th Annual International Computing and Combinatorics Conference* (COCOON 2005), *Lecture Notes in Computer Science* 3595, pp. 849–858, Springer, 2005.

[6] Achilleas Papakostas and Ioannis G. Tollis, Algorithms for area-efficient orthogonal drawings, *Computational Geometry: Theory and Applications* 9, pp. 83–110, 1998.

[7] Thomas J. Schaefer, The complexity of satisfiability problems, *Proceedings of 10th Annual ACM Symposium on Theory of Computing* (STOC'78), pp. 216–226, ACM, 1978.

[8] David R. Wood, Minimizing the number of bends and volume in three-dimensional orthogonal graph drawings with a diagonal vertex layout, *Algorithmica* 39, pp. 235–253, 2004.

[9] David R. Wood, Optimal three-dimensional orthogonal graph drawing in the general position model, *Theoretical Computer Science* 299 (1–3), pp. 151-178, 2003.