# 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.edu

**Abstract.** 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.* [1]. 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 closing 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 six and for 5-regular graphs. On the other hand we present 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.

## 1 Introduction

A number of algorithms for graph drawing use a 'balanced' ordering of the vertices of the graph as a starting point [2–4, 6, 7]. Here balanced means that neighbours of each vertex $v$ are as evenly distributed to the left and right of $v$ as possible (see below for more precise definition). 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 $\sigma : V \to \{1, \ldots, |V|\}$. For $u, v \in V$ with $\sigma(u) < \sigma(v)$, we say that $u$ is *to the left*

* Supported by grant MEC SB2003-0270. Research completed at the Department of Applied Mathematics and the Institute for Theoretical Computer Science, Charles University, Prague, Czech Republic. Supported by projects LN00A056 and 1M0021620808 of the Ministry of Education of the Czech Republic, and by the European Union Research Training Network COMBSTRU (Combinatorial Structure of Intractable Problems)

L. Wang (Ed.): COCOON 2005, LNCS 3595, pp. 849–858, 2005.

of $v$ and that $v$ is *to the right* of $u$. The *imbalance* of $v \in V$ in $\sigma$, denoted by $B_\sigma(v)$, is

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

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

- Every vertex of odd degree has imbalance at least one.
- The two vertices at the beginning and at the end of any 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 $(d_1, \ldots, d_n)$ be the sequence of vertex degrees of $G$, where $d_i \leq d_{i+1}$ for all $1 \leq i \leq n-1$. Then

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

An ordering $\sigma$ 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.

Whether the balanced ordering problem is efficiently solvable for planar graphs is of particular interest since planar graphs are often used in graph drawing applications. 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 six.

Our third $\mathcal{NP}$-hardness result states that finding a minimum ordering is $\mathcal{NP}$-hard for 5-regular simple graphs. All of these $\mathcal{NP}$-hardness results are presented in Section 3. The proofs are based on reductions from various satisfiability problems. Section 2 contains several $\mathcal{NP}$-completeness results for the satisfiability problems that we use.

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 state several $\mathcal{NP}$-hardness results about various satisfiability problems. The results in this section can be achieved by verifying conditions of a general theorem of Schaefer [5]. First we introduce several basic definitions about satisfiability. Throughout this paper, formulae are considered to be in a *conjunctive normal form*. We allow a variable to occur several times in one clause but note that the graphs created in this way are simple (unless stated otherwise). Suppose $\varphi$ is a formula with variables $x_1, \ldots, x_n$. The *incidence graph* of $\varphi$ is the bipartite graph with vertices $c_1, \ldots, c_m$ and $x_1, \ldots, x_n$, where $\{c_i, x_j\}$ is an edge if and only if the variable $x_j$ occurs in the clause $c_i$ (positive or negated). A *truth assignment* of a formula $\varphi$ with variables $x_1, \ldots, x_n$ is an arbitrary function $t : \{1, \ldots, n\} \to \{0, 1\}$. The values 0 and 1 are also sometimes called *false* and *true* respectively. A truth assignment $t$ is *satisfying* $\varphi$ if there is at least one true literal in every clause. The formula $\varphi$ is *satisfiable* if it has at least one satisfying truth assignment.

The decision problem asking whether a given formula $\varphi$ is satisfiable is called SAT. If we assume that every clause in the given formula $\varphi$ has size exactly three, then the decision problem asking whether $\varphi$ is satisfiable is called 3SAT. Two common variants of 3SAT are *Not–All–Equal 3-Satisfiability* (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. A truth assignment $t$ is *NAE satisfying* if each clause has at least one true and at least one false literal, and $t$ 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. SAT is one of the basic $\mathcal{NP}$-complete problems, and it is well known that NAE–3SAT and 1–IN–3SAT are $\mathcal{NP}$-complete even for formulae without negations [5].

We say that a formula $\varphi$ 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 without negations is 2–or–3–in–5 satisfiable. For a formula $\varphi$, in which all clauses have six literals, a truth assignment $t$ is *3–in–6 satisfying* if each clause in $\varphi$ has exactly three true literals. The formula $\varphi$ is *3–in–6 satisfiable* if there exists a 3–in–6 satisfying truth assignment. 3–IN–6SAT is the decision problem asking whether a given formula $\varphi$ is 3–in–6 satisfiable. The fact that 2–OR–3–IN–5SAT is $\mathcal{NP}$-complete and that 3–IN–6SAT is $\mathcal{NP}$-complete for formulae without negations follows from [5].

Now we strengthen the result about 3–IN–6SAT.

**Proposition 1.** *Problem* 3–IN–6 SAT *is* $\mathcal{NP}$*-complete for planar formulae without negations.*

*Proof.* Suppose we have a formula $\varphi$ with clauses of size six without negations. We now show that if the formula $\varphi$ is not planar we can alter it in polynomial time so that the resulting formula $\varphi'$ is planar and $\varphi$ is 3–in–6 satisfiable if and only if $\varphi'$ is 3–in–6 satisfiable. This will prove the lemma. Let $d$ be a drawing of the incidence graph of $\varphi$ in the plane, such that any two edges cross at most once. For each pair of crossing edges $e = (v, c)$ and $e' = (v', c')$, add four new variables $v_1^{ee'}, \dots, v_4^{ee'}$ and three clauses $c^{ee'} = v \vee v \vee v_1^{ee'} \vee v_1^{ee'} \vee v' \vee v_2^{ee'}$, $c_e^{ee'} = v_1^{ee'} \vee v_1^{ee'} \vee v_1^{ee'} \vee v_3^{ee'} \vee v_3^{ee'} \vee v_3^{ee'}$, $c_{e'}^{ee'} = v_2^{ee'} \vee v_2^{ee'} \vee v_2^{ee'} \vee v_4^{ee'} \vee v_4^{ee'} \vee v_4^{ee'}$. Then substitute occurrences of $v$ in $c$ by $v_3^{ee'}$, and occurrences of $v'$ in $c'$ by $v_4^{ee'}$. See Figure 1 for an example of a gadget for two crossing edges.

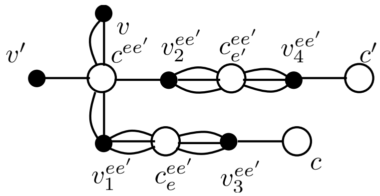


**Fig. 1.** The crossing gadget for two edges $\{v, c\}$ and $\{v', c'\}$. Empty circles represent clauses, and full circles represent variables

After the substitutions we clearly obtain a planar formula. It remains to prove that $\varphi'$ is 3–in–6 satisfiable if and only if $\varphi$ is. To do so, we show that 3–in–6 satisfiability of the formula is unchanged by each substitution. Let $t$ be a 3–in–6 satisfying truth assignment for $\varphi$ and let $\psi$ be the formula obtained from $\varphi$ by the substitution described above. Setting $t'(x) = t(x)$ for all variables $x$ of $\varphi$ and $t'(v_1^{ee'}) = \neg t(v), t'(v_2^{ee'}) = \neg t(v'), t'(v_3^{ee'}) = t(v)$ and $t'(v_4^{ee'}) = t(v')$, we obtain a 3–in–6 satisfying truth assignment for $\psi$. The other implication can be seen as follows. Let $t'$ be a 3–in–6 satisfying truth assignment for $\psi$. Hence it must hold that $t'(v_1^{ee'} = \neg t'(v_3^{ee'})$ and $t'(v_2^{ee'}) = \neg t'(v_4^{ee'})$. It is also clear that $t'(v) = \neg t'(v_1^{ee'}) = t'(v_3^{ee'})$. Thus, regardless of the truth assignment, there are two true and two false literals in the clause $c^{ee'}$. Hence $t'(v') = \neg t'(v_2^{ee'}) = t'(v_4^{ee'})$ and we can conclude (because $t'(v) = t(v_3^{ee'})$ and $t'(v') = t(v_4^{ee'})$) that if $t'$ is restricted to the variables of $\varphi$, then a 3–in–6 satisfying truth assignment is obtained.

## 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 graphs with maximum degree four.*

*Proof.* The construction is a refinement of a construction by Biedl *et al.* [1]; the difference being that we reduce the maximum degree from six to four. $\mathcal{NP}$-hardness is proved by a reduction from NAE–SAT. Given a formula $\varphi$, create a graph $G_\varphi$ with one vertex $u_c$ for each clause $c$. For each variable $v$ that occurs $o_v$ times in $\varphi$, add a path on $3o_v + 1$ new vertices $p_1^v, \dots, p_{3o_v+1}^v$ to $G_\varphi$, 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 $p_{3i}^v$ of the path. 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.

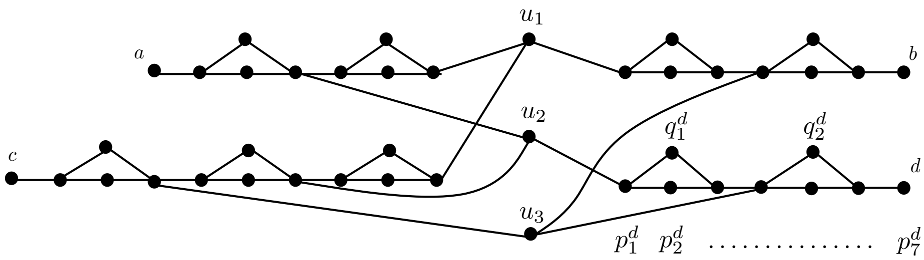


**Fig. 2.** Constructed graph for formula $(a \vee b \vee c) \wedge (c \vee a \vee d) \wedge (d \vee c \vee b)$. The three clauses have numbers $1, 2, 3$ in the picture

Observe that the maximum degree of $G_\varphi$ is four. In particular, $\deg(u_c) = 3$, $\deg(q_i^v) = 2$ 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 $\deg(p_{3o_v+1}^v) = 1$.

We now prove that $G_\varphi$ has a perfect ordering if and only if $\varphi$ is NAE-satisfiable. Suppose $G_\varphi$ has a perfect linear ordering $\sigma$. For each variable $v$, since $\deg(p_{3i-1}^v) = 2$ and $\deg(q_i^v) = 2$, vertices $p_{3i-1}^v, i \in \{1, \ldots, o_v\}$, and $q_i^v, i \in \{1, \ldots, o_v\}$, must have one incident edge to the left and one to the right in $\sigma$. 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 $\sigma$, 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 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 NAE-satisfying because each clause vertex has at least one neighbour 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 $p_{3i-2}^v$. $\square$

Now we present the result about ordering of planar graphs:

**Theorem 2.** *The* PERFECT ORDERING *problem is $\mathcal{NP}$-complete for planar simple graphs with maximum degree six.*

*Proof.* We reduce the problem of 3–IN–6 SAT for planar formulae to the PERFECT ORDERING problem for planar graphs. To do so, use the graph construction from the proof of Theorem 1. Note that multiple occurrences of a variable in a clause do not create any parallel edges in the constructed graph. Clearly the construction creates planar graph of maximum degree six from a planar formula and perfect orderings of the created graph correspond to 3–in–6 satisfying truth assignments, as in the proof of Theorem 1. □

The following two technical lemmas will be used later for removing parallel edges from a multigraph without changing an ordering with minimum imbalance. Their proofs are omitted due to the space limitation.

**Lemma 1.** *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 $G$.*

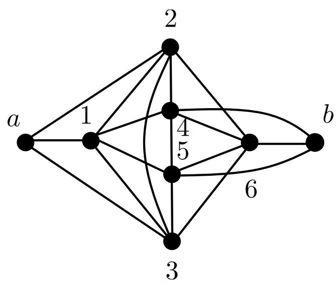


**Fig. 3.** The triple edge gadget

**Lemma 2.** *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 $M(G) = M(G') - 10 \cdot c$.*

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 3.** *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. Suppose that we are given a formula $\varphi$ without negations and with all clauses of size five. Moreover we 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

$\varphi$. For each clause $c$ add a new vertex $v_c$ to $G$. For each variable $x$ that occurs $o_x$ times in $\varphi$, 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 that 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`
  Take three arbitrary vertices $u_i, u_j, u_k \in P$
  $P := P \setminus \{u_i, u_j, u_k\} \cup \{u_{n+1}, u_{n+2}\}$
  Add a complete bipartite graph on $u_i, u_j, u_k$ and $u_{n+1}, u_{n+2}$ to $G$.
  $n := n + 2$
`end`
*Now* $P = \{u_i, u_j\}$
Add to $G$ a complete bipartite graph on $u_i, u_j$ and new vertices $s_1, s_2$.
Add a triple-edge $s_1 s_2$ to $G$.

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 and we show that $G$ has a perfect ordering if and only if $\varphi$ was 2-or-3-in-5 satisfiable. Suppose we have a perfect ordering of $G$. In every ordering of $s_1, s_2$ and their neighbours $u_i, u_j$, $B(s_1) + B(s_2) > 2$. Thus (from the perfectness of the ordering) the ordering begins $s_1, s_2$ without loss of generality. By a similar argument, the ordering ends by vertices $s_2', s_1'$, where $s_1'$ and $s_2'$ are the vertices added in the end of the second run of the procedure on $P$. Because all other vertices must be 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 $\varphi$ by assigning $t(x) = 0$ to the variables whose path is in natural order and $t(x) = 1$ to the variables whose path is reversed. For the other implication, suppose we have a 2-or-3-in-5 satisfying truth assignment $t$ of $\varphi$. We can place vertices $s_1, s_2, u_{n_1}, \ldots, u_{n_0+1}$ added in the first run, then vertices $p_j^x : x$ is a variable with $t(x) = 0, 0 \leq j \leq o_x - 1$, then variable paths for variables $x$ such that $t(x) = 0$ in their natural ordering, then the clause vertices, and then symmetrically the rest of the paths and vertices added in the second run. It is straightforward to check that this ordering is perfect. $\square$

See an example of the above construction in Figure 4.

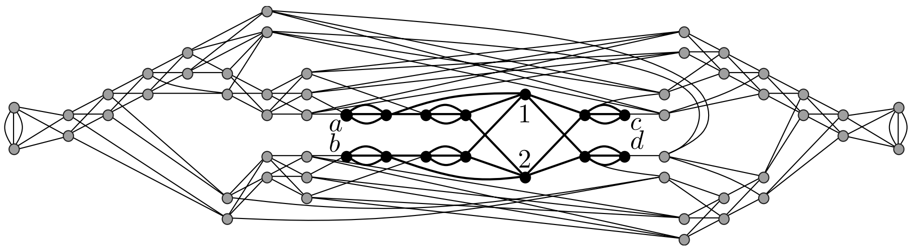


**Fig. 4.** Constructed 5-regular multigraph for formula $(a \vee a \vee b \vee c \vee d) \wedge (a \vee b \vee b \vee c \vee d)$. Clause vertices are marked 1 and 2. Clause vertices and variable paths are drawn in black colour, vertices $p_i^x$ and vertices added by the procedure are in gray colour

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

*Proof.* Construct the multigraph $G$ as in the reduction in the proof of Theorem 3. 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 2 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 and hence the statement of the corollary. □

## 4 Algorithm

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

**Lemma 3.** *There is an $O(n + m)$ time algorithm to test whether a multigraph $G$ with $n$ vertices and $m$ edges has an ordering $\sigma$ in which a given list of vertices imbalanced $= (v_1, \ldots, v_k)$ are the only imbalanced vertices, and $\sigma(v_i) < \sigma(v_{i+1})$ for all $1 \leq i \leq k - 1$.*

*Proof.* 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 $\sigma$ from left to right. We append to $\sigma$ 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. 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 would not get a valid ordering. □

The following theorem is a consequence of Lemma 3.

**Theorem 4.** *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 choice run the procedure from Lemma 3 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 3, we can check in $O(m+n)$ time whether there is an ordering with the chosen vertices imbalanced, and compute the imbalance of the ordering in the case the procedure produced one. □

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

*Proof.* Apply the algorithm from Theorem 4 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). If the algorithm outputs some ordering with imbalance less than $c$, then we are also 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 4 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 (if there is at least one edge). □

## 5 Conclusion and Open Problems

In this paper we have considered the problems of deciding the existence of a perfect ordering for graphs with maximum degree four, planar graphs with maximum degree six and 5-regular multigraphs. All these problems were shown to be $\mathcal{NP}$-complete, thus answering a number of questions raised by Biedl *et al.* [1]. The result for planar graphs still leaves unresolved the complexity of the PERFECT ORDERING problem for planar graphs with maximum degree four or five. We have also established that it is $\mathcal{NP}$-hard to find an ordering with minimum imbalance for 5-regular simple graphs. In the positive direction, we have presented an algorithm for determining an ordering with imbalance smaller than

$k$ running in time $O(n^k(n+m))$. It would be interesting to obtain a fixed-parameter-tractable (FPT) algorithm for this problem (as one cannot hope for a polynomial solution unless $\mathcal{P} = \mathcal{NP}$).

## References


1. Therese Biedl, Timothy Chan, Yashar Ganjali, MohammadTaghi Hajiaghayi, David R. Wood, Balanced vertex-orderings of graphs, *Discrete Applied Mathematics* 148(1), pp. 27–48, 2005.
2. Goos Kant, Drawing planar graphs using the canonical ordering, *Algorithmica* 16, pp. 4–32, 1996.
3. 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.
4. Achilleas Papakostas and Ioannis G. Tollis, Algorithms for area-efficient orthogonal drawings, *Computational Geometry: Theory and Applications* 9, pp. 83–110, 1998.
5. 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.
6. 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.
7. David R. Wood, Optimal three-dimensional orthogonal graph drawing in the general position model, *Theoretical Computer Science* 299 (1–3), pp. 151-178, 2003.