On Subfamilies of AT-Free Graphs

(Extended Abstract)

Ekkehard Köhler * Derek G. Corneil † Stephan Olariu ‡

Lorna Stewart §

Abstract

We introduce two subfamilies of AT-free graphs, namely, path orderable graphs and strong asteroid free graphs. Path orderable graphs are defined by a linear ordering of the vertices that is a natural generalization of the ordering that characterizes cocomparability graphs. On the other hand, motivation for the definition of strong asteroid free graphs comes from the fundamental work of Gallai on comparability graphs.

We show that cocomparability graphs $\subset$ path orderable graphs $\subset$ strong asteroid free graphs $\subset$ AT-free graphs. In addition, we settle the recognition question for the two new classes by proving that recognizing path orderable graphs is NP-complete whereas the recognition problem for strong asteroid free graphs can be solved in polynomial time.

1 Introduction

We say that a vertex in a graph $G = (V, E)$ *intercepts* a path in G if it is adjacent to at least one vertex of the path, and it *misses* the path otherwise. An *asteroidal triple* (*AT* for short) is an independent set of three vertices such that, between every pair, there is a path that is missed by the third. A graph is *AT-free* if it does not contain an AT.

One of the most compelling motivations for the study of AT-free graphs is the idea that these graphs exhibit a type of linear structure. Indeed, the linear structure exhibited by AT-free graphs is explained, in part, in [1], where it is shown that every connected AT-free graph contains a *dominating pair* (two vertices such that every path connecting them is a dominating set), and a type of linear "shelling sequence" called a *spine*.

The original motivation for the results of the present paper was the idea that AT-free graphs might be further characterized by the existence of a vertex ordering satisfying certain conditions.

Vertex orderings have proven to be useful algorithmic tools for several families of graphs. For example, chordal graphs (respectively, cocomparability graphs) are characterized by the existence of vertex orderings that do not contain the forbidden ordered configuration shown in Figure 1 (a) [2] (respectively, (b) [8]). A graph is an *interval graph* if and only if it has a vertex ordering that contains neither of the configurations of Figure 1 (see for example [10]). Such vertex orderings are referred to as chordal orderings, cocomparability orderings, and interval orderings, respectively.

In other words, in an interval ordering, for every path on two vertices (that is, for every edge), the left endpoint of the path is adjacent to all vertices between the two endpoints of the path. In a cocomparability ordering, each vertex between the two endpoints of a P_2 is adjacent to one or both endpoints of the P_2.

It is well-known that interval graphs are exactly those graphs that are both chordal and cocomparability [5] or, equivalently, both chordal and AT-free [9]. Furthermore, cocomparability graphs form a proper subclass of AT-free graphs [6].

*Technische Universität Berlin

†University of Toronto

‡Old Dominion University

§University of Alberta

Figure 1: Forbidden ordered configurations.

An alternate characterization of the cocomparability ordering is given in Observation 1.1.

Observation 1.1 *A vertex ordering* $v_1, \ldots, v_n$ *of graph* G *is a cocomparability ordering if and only if* $\forall\ v_i, v_j, v_k$ *with* $i < j < k$*, vertex* v_j *intercepts each* v_i, v_k*-path of* G*.*

From this, one can easily see that a cocomparability graph must be AT-free since any independent triple occurs in some order, say $x < y < z$, in a cocomparability ordering, and thus, there cannot exist an x, z-path missed by y.

In an attempt to generalize the cocomparability ordering while retaining the AT-free property, we introduce the following definition.

Definition 1.2 *A graph* $G = (V, E)$ *is* path orderable *if there is an ordering of the vertices* $v_1, \ldots, v_n$ *such that for each triple of vertices* v_i, v_j, v_k *with* $i < j < k$ *and* $v_i v_k \notin E$*, vertex* v_j *intercepts each* v_i, v_k*-path of* G*; such an ordering* $v_1, \ldots, v_n$ *of* V *is called a* path ordering.

It is easily seen by Observation 1.1 and Definition 1.2 that cocomparability graphs are all path orderable. C_5, the chordless cycle on five vertices, is a path orderable graph which is not a cocomparability graph. It is clear that path orderable graphs must be AT-free; however, while it was hoped that Definition 1.2 might characterize AT-free graphs, we shall see later that path orderable graphs form a strict subset of AT-free graphs; as an example see the graph in Figure 2. It is obviously AT-free, however it has no path ordering, as will be shown later.

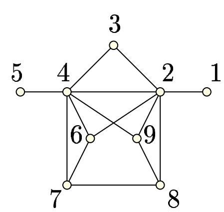

Figure 2: AT-free graph G which is not path orderable.

Nevertheless, since path orderable graphs are interesting in their own right, we attempted to provide a structural characterization of this graph class by identifying a type of forcing relation on the nonadjacent pairs of vertices and the type of structure which would make the vertex ordering of Definition 1.2 impossible.

These investigations follow in Gallai's footsteps [3] in that they involve ideas similar to his forcing relation on the edges of a comparability graph (equivalently, the nonedges of a cocomparability graph) and his definitions of wreaths and asteroids. Specifically, we define strong asteroids and show that path orderable graphs are strong asteroid free. However, it turns out that the strong asteroid concept does not provide a characterization of path orderable graphs; we shall see that path orderable graphs form a proper subclass of strong asteroid free graphs which, in turn, form a proper subclass of AT-free graphs.

Thus, we will identify two distinct subclasses of AT-free graphs, both of which contain cocomparability graphs:

cocomparability $\subset$ path orderable $\subset$ strong asteroid free $\subset$ AT-free

The interest lies, in part, in the natural vertex ordering, in one case, and the relationship with Gallai's work, in the other case. Furthermore, the identification of these graph classes should allow us to narrow the gap between known polynomial and known NP-complete behaviour of problems in the domain of AT-free graphs.

We conclude the paper with a proof that the recognition of path orderable graphs is NP-complete, and a polynomial time recognition algorithm for strong asteroid free graphs. We note that the NP-completeness result settles an open problem stated in [12].

2 Background

In his paper on comparability graphs, Gallai studies the forcing between the edges imposed by a transitive orientation (to avoid misunderstandings, from now on we will refer to the transitive-forcing as t-forcing). Let G be a (not necessarily comparability) graph. Two edges which share a common endpoint and whose other endpoints are nonadjacent t-force each other directly. That is, in any transitive orientation, either both edges are directed away from the common endpoint, or both are directed towards it. The transitive closure of the direct t-forcing relation partitions the edges of a graph into its *t-forcing-classes*. Either there are exactly two different transitive orientations of the edges of a t-forcing-class, or there are none. The latter case occurs when some edge is t-forced in both directions, in which case the graph is not a comparability graph. Edges xy and xz are said to be *knotted* if y and z are connected in $\overline{N(x)}$, the complement of the subgraph of G induced by $\mathrm{N}(x)$, where $\mathrm{N}(x)$, the neighbourhood of x, is defined by $N(x) = \{u | ux \in E\}$.

To capture the t-forcing in a given graph G, Gallai uses the concept of a *knotting graph*: For a graph $G = (V, E)$ the corresponding *knotting graph* is given by $\mathrm{K}[G] = (V_K, E_K)$ where V_K and E_K are defined as follows. For each vertex v of G there are copies $v_1, v_2, \ldots, v_{i_v}$ in V_K, where i_v is the number of connected components of $\overline{\mathrm{N}(v)}$, the complement of the graph induced by $\mathrm{N}(v)$. For each edge vw of E there is an edge $v_i w_j$ in E_K, where w is contained in the i^{th} connected component of $\overline{\mathrm{N}(v)}$ and v is contained in the j^{th} connected component of $\overline{\mathrm{N}(w)}$.

In this graph two edges are incident if they are knotted. The edges of the t-forcing classes of G are given by the connected components of $\mathrm{K}[G]$. Using this structure, Gallai shows that a graph G is a comparability graph if and only if $\mathrm{K}[G]$ is bipartite.

The following definitions describe structures which lead to t-forcing classes which cannot be transitively oriented, and knotting graphs which are not bipartite.

Definition 2.1 *An* odd wreath *of size k in a graph is a cycle of knotted edges, specifically, a sequence of vertices $v_0, v_1, v_2, \ldots, v_k$ where k is odd, $v_1, \ldots, v_k$ are distinct, $v_0 = v_k$, and $\forall 0 \leq i < k$, edges $v_i v_{i+1}$ and $v_{i+1} v_{i+2}$ exist in the graph and are knotted (addition modulo k).*

Definition 2.2 *An* odd asteroid *of size k in a graph is a sequence of vertices $v_0, v_1, v_2, \ldots, v_k$ where k is odd, $v_1, \ldots, v_k$ are distinct, $v_0 = v_k$, and $\forall 0 \leq i < k$, $\exists$ a $v_i v_{i+1}$-path in G which is missed by $v_{(i+\frac{k+1}{2}) \bmod k}$.*

Gallai points out that an asteroid is the complement of a wreath, and proves that a graph is a comparability graph if and only if it contains no odd wreath or, equivalently, a graph is a cocomparability graph if and only if it contains no odd asteroid.

3 Path Orderable Graphs and Strong Asteroid Free Graphs

t-forcing is a fundamental concept for comparability graphs, and thus for cocomparability graphs as well. Given the similarities of the linear ordering characterizations of path orderable graphs and cocomparability graphs, one might expect a similar forcing concept for path orderable graphs. In fact such is the case.

For a graph G and a vertex v of G let $C_1, \ldots, C_k$ be the connected components of $G \setminus (\mathrm{N}(v) \cup \{v\})$ and let $B_i^1, \ldots, B_i^\ell$ be the connected components of the graph induced by the vertices of C_i in $\overline{G}$ ($1 \leq i \leq k$); the B_i^j are called the *blobs* of v in G.

Lemma 3.1 *Let G be a path orderable graph and $v_1, \ldots, v_n$ a corresponding path ordering. For every vertex v of G and every blob B of v, the vertices of B either occur all before v in the path ordering or all after v in the path ordering.*

PROOF. Suppose there is a vertex v and a blob B of v with $u, w \in B$ and $u < v < w$ in the path ordering of G. By the definition of blobs, u and w are in the same connected component C of $G \setminus \mathrm{N}(v)$. Since u and w are also in the same connected component B of C in $\overline{G}$, there has to be a path of non-edges in B between u and w. Thus, there is a pair of vertices u', w' in B with $u'w' \notin E$ and $u' < v < w'$. But $u', w' \in C$; therefore there is a u', w'-path in $G \setminus \mathrm{N}(v)$, contradicting the path ordering. □

By Lemma 3.1, any path ordering has to fulfill the property that if one of the vertices u of a blob B of v precedes v in the ordering, then all of the vertices of B have to be in front of v as well. As an example consider the graph in Figure 2. Following the above definition of blobs, vertex 3 has the three blobs $\{6,7,8,9\}$, $\{5\}$, $\{1\}$; vertex 7 has the blobs $\{3,1,9\}$, $\{2\}$, $\{5\}$; vertex 8 has the blobs $\{3,5,6\}$, $\{4\}$, $\{1\}$; vertex 5 has only the blob $\{1,2,3,6,7,8,9\}$ and vertex 1 has only the blob $\{3,4,5,6,7,8,9\}$. Suppose now there is a path ordering of G. By Lemma 3.1 we can, without loss of generality, assume that 1 is in front of all vertices of its blob and thus 5 appears after all vertices of its blob in the path ordering; in particular vertices 3, 6, 7, 8, 9 are between 1 and 5. Since 7 and 8 are in the same blob of 3, they either appear both before or both after 3 in the path ordering. However, if they both appear before 3, then, again by Lemma 3.1, we have a contradiction because 3 and 1 are in the same blob of 7, but on different sides in the path ordering. On the other hand, if both 7 and 8 appear after 3 in the path ordering we have again a contradiction, since 3 and 5 are in the same blob of 8 but on different sides in the path ordering. Hence there cannot be a path ordering for the graph in Figure 2.

When interpreting the constraints of Lemma 3.1 as orientations of the edges of $\overline{G}$, in the sense that edges from the same blob of a vertex v to v in $\overline{G}$ have to have the same orientation, one can define the following forcing on the edge set of $\overline{G}$.

Let G be a graph (not necessarily path orderable) and let $e_1 = uv$, $e_2 = vw$ be edges of $\overline{G}$ with a common end-vertex v. Then one can define a relation $\approx$ by $e_1 \approx e_2$ (e_1 and e_2 *force each other* or *are knotted* at v) if and only if u and w are in the same blob of v (possibly $u = w$) in G. The consecutive application of this relation defines a class partition of the edges of $\overline{G}$, where two edges e_a, e_b are in the same class (*forcing class*) of $\overline{G}$ if there is a sequence $e_1, e_2, \ldots, e_k$ of edges such that $e_a = e_1 \approx e_2 \approx \ldots \approx e_k = e_b$. Observe that the forcing classes are refinements of the t-forcing classes.

An orientation of the edges of $\overline{G}$ is said to *agree with the forcing* if for any vertex v and any blob B of v all edges between B and v are oriented in the same direction (either towards v or away from v). For a graph G a linear ordering $v_1, \ldots, v_n$ of the vertices of G is said to *agree with the forcing* if the corresponding implied orientation of the edges of $\overline{G}$ (uv is oriented from u to v if $u < v$ in the linear ordering) agrees with the forcing.

Note that when the orientation of one of the edges of a forcing class is fixed, then the orientation of all the edges of its forcing class is determined; hence, either there are exactly two different orientations of the edges of a forcing class that agree with the forcing, or none. In the latter case, some edge is forced to be oriented in both directions, meaning that there is no ordering consistent with the forcing.

Lemma 3.2 *A graph G is path orderable if and only if there is a linear ordering of the vertices of G agreeing with the forcing.*

PROOF. If G is path orderable, then, by Lemma 3.1, the path ordering has to agree with the forcing relation.

Suppose there is a linear ordering of G, that agrees with the forcing relation and suppose there is a triple $u < v < w$ of vertices that violates the path ordering property, i.e. $uw \notin E$ and there is a u,w-path in $G \setminus \mathrm{N}(v)$. Hence, u and w are in the same connected component C of $G \setminus \mathrm{N}(v)$ and, since $uw \notin E$, u and w are also in the same blob B of v. But then this ordering does not agree with the forcing relation; contradiction. □

Corollary 3.3 *A graph G is path orderable if and only if there is an acyclic orientation of $\overline{G}$, agreeing with the forcing relation.*

PROOF. Determine a topological ordering, using the acyclic orientation of $\overline{G}$; then the corollary follows from Lemma 3.2. □

One can define a graph, similar to Gallai's knotting graph, representing the forcing classes of $\overline{G}$. For a graph $G = (V, E)$ the *altered knotting graph* is given by $\mathrm{K}^*[G] = (V_K, E_K)$ where V_K and E_K are defined as follows. For each vertex v of G there are copies $v_1, \ldots, v_{i_v}$ in V_K, where i_v is the number of blobs of v in $\overline{G}$. For each edge vw of E there is an edge $v_i w_j$ in E_K, where w is contained in the i^{th} blob of v in $\overline{G}$ and v is contained in the j^{th} blob of w in $\overline{G}$.

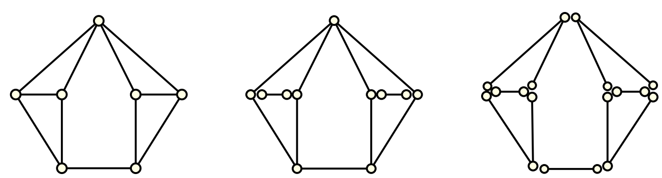

Figure 3: A graph G together with $\mathrm{K}[G]$ and $\mathrm{K}^*[G]$.

As Gallai did for the knotting graph, we draw the altered knotting graph $\mathrm{K}^*[G]$ of a given graph G by putting different copies of the same vertex close together.

Our next task is to examine configurations which cannot occur in path orderable graphs. In a step toward this goal, we define restricted types of wreaths and asteroids.

Definition 3.4 *An* odd strong wreath *of size k in a graph G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where k is odd, $v_1, \ldots, v_k$ are distinct, $v_0 = v_k$, and $\forall 0 \leq i < k$, edges $v_i v_{i+1}$ and $v_{i+1} v_{i+2}$ exist in the graph and are knotted in the altered sense, that is, v_i and v_{i+2} are in the same blob of v_{i+1} in $\overline{G}$ (addition modulo k).*

Definition 3.5 *An* odd strong asteroid *of size k in a graph G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where k is odd, $v_1, \ldots, v_k$ are distinct, $v_0 = v_k$, and $\forall 0 \leq i < k$, v_i and v_{i+1} are in the same blob of $v_{(i+\frac{k+1}{2}) \bmod k}$ in G.*

The two notions are complementary, that is, a graph G has an odd strong wreath if and only if $\overline{G}$ contains an odd strong asteroid. Furthermore, strong asteroids and strong wreaths are restricted types of asteroids and wreaths. We also note that the asteroidal triples are the odd strong asteroids of size three.

Definition 3.6 *A graph G is* strong asteroid free *if it does not contain an odd strong asteroid.*

Similar to the t-forcing results, the following holds:

Lemma 3.7 *The forcing-classes of a graph G correspond exactly to the connected components of $\mathrm{K}^*[G]$.*

The next two observations follow from the fact that an odd strong asteroid of size k in G corresponds to an odd cycle of size k in $\mathrm{K}^*[\overline{G}]$.

Observation 3.8 *A graph G is strong asteroid free if and only if $\mathrm{K}^*[\overline{G}]$ is bipartite.*

Observation 3.9 *A graph G is AT-free if and only if $\mathrm{K}^*[\overline{G}]$ is triangle-free.*

Lemma 3.10 *If a graph G is path orderable then $\mathrm{K}^*[\overline{G}]$ is bipartite.*

PROOF. Let $v_1, \ldots, v_n$ be a path ordering of G. Now orient the edges of $\mathrm{K}^*[\overline{G}]$ as follows: $v_i v_j$ is oriented from v_i to v_j if $i < j$. Now, by Lemma 3.1, each vertex of $\mathrm{K}^*[\overline{G}]$ has either only incoming or only outgoing edges. Hence, it is bipartite. □

Not only does the graph in Figure 2 show that path orderable graphs are strictly contained in AT-free graphs, it also establishes that strong asteroid free graphs are strictly contained in AT-free graphs, as shown in the next lemma.

Lemma 3.11 *The class of strong asteroid free graphs is strictly contained in the class of AT-free graphs.*

PROOF. Consider the graph of Figure 2. It is easy to check that 1 3 5 7 8 is an odd strong asteroid in G, and that G is AT-free. □

In the case of comparability graphs, Gallai not only showed that the knotting graph $\mathrm{K}[G]$ of a comparability graph is bipartite, but also proved that a bipartite knotting graph $\mathrm{K}[G]$ is a sufficient condition for G being a comparability graph. The major tool that he used for proving this result is a lemma which shows the following. Given a bipartite knotting graph $\mathrm{K}[G]$ and consider a triangle of G with the property that at least two of the edges of the triangle are in the same t-forcing-class. Then in any orientation of G that agrees with the t-forcing the triangle is not oriented cyclically.

It turns out that a similar lemma holds for strong asteroid free graphs, too. Specifically, for a graph G with a bipartite altered knotting graph $\mathrm{K}^*[\overline{G}]$, any orientation of $\overline{G}$ that agrees with the forcing relation does not contain a cyclically oriented triangle. However, contrary to the t-forcing relation, this lemma is not enough to imply that the orientation is acyclic and, indeed, we shall show that this is not necessarily the case.

Observation 3.12 *Given a vertex v in a graph H and vertices $u, w \in \mathrm{N}(v)$, which are the end-points of a P_4 in $\mathrm{N}(v)$, then the edges uv and wv are forcing each other (see Figure 4).*

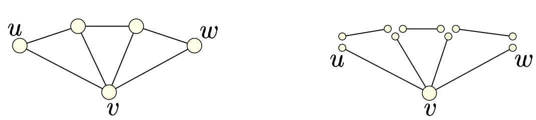

Figure 4: Vertex v with P_4 in $\mathrm{N}(v)$ together with corresponding altered knotting graph $\mathrm{K}^*[H]$.

Remark 3.13 Using this observation one can create a *forcing path*, i.e. a path P, where each consecutive pair of edges of P is knotted at the common end-vertex by the help of an added P_4 as described in Observation 3.12—see Figure 5 (in the following, edges and vertices of the path P itself are called *original edges/vertices*, the added edges and vertices are denoted as *auxiliary edges/vertices*). By the forcing, the orientation of any original edge of P forces the orientation of all other original edges of P. Note that the knotting graph of a forcing path does not contain a triangle or any odd cycle. Furthermore, if P has even length then the end-edges of P are either both oriented towards the inner vertices of P or both oriented outwards from the inner vertices of P. Similarly, if P has odd length the end edges of P have opposite orientations with respect to the inner vertices of P.

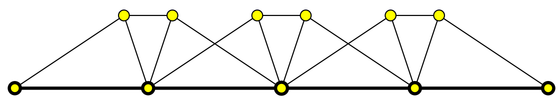

Figure 5: A forcing path of length 4 (original edges and vertices are bold).

Theorem 3.14 *The class of path orderable graphs is strictly contained in the class of strong asteroid free graphs.*

PROOF. Consider the left graph in Figure 6. This graph is the complement of a strong asteroid free graph G. This is proved by constructing the altered knotting graph $\mathrm{K}^*[\overline{G}]$ (see the right graph in Figure 6). By Observation 3.12, the thick edges force each-other, as shown in the altered knotting graph; and, without having a strong asteroid in G, there is a forced oriented cycle on the vertices $x_1, \ldots, x_k$ in $\overline{G}$. Consequently, by Corollary 3.3, G is not path orderable. This construction holds for any $k \geq 4$. □

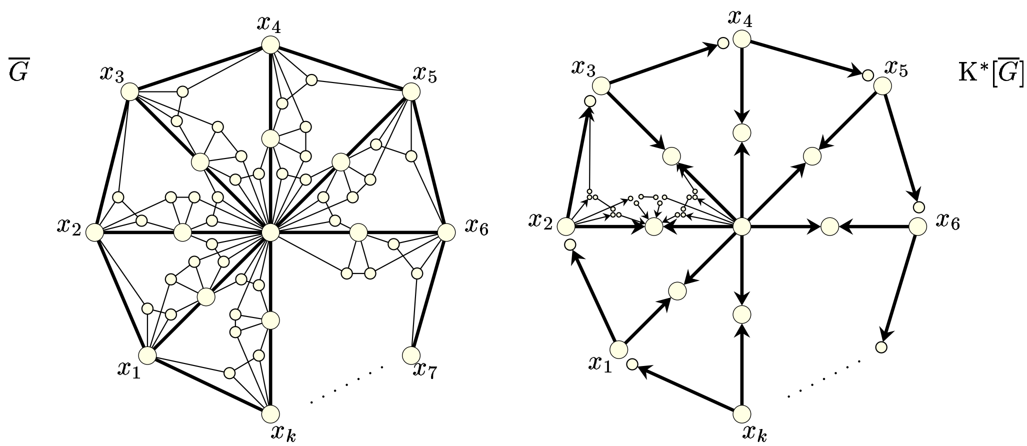

Figure 6: Complement of a strong asteroid free graph, which is not path orderable (left), together with its altered knotting graph (right). To ease understanding of its structure, in the knotting graph only for one of the "arms" of the example the corresponding auxiliary P_4 vertices are drawn in the figure. One of the two possible forced orientations of the main forcing class is given in the right picture.

4 Recognition of Path Orderable and Strong Asteroid Free Graphs

In this section, we show that the recognition of path orderable graphs is NP-complete. This result answers a question which is posed by J. Spinrad in [12]. In contrast, we describe how to recognize strong asteroid free graphs in polynomial time.

If there is only one forcing class for the edge set of $\overline{G}$ one can check in polynomial time whether G is path orderable: Compute $\mathrm{K}^*[\overline{G}]$, Check whether it is bipartite, assign an orientation to $\mathrm{K}^*[\overline{G}]$ by orienting all edges from one of the bipartition classes to the other and check whether this orientation is acyclic on $\overline{G}$.

Similarly one can check whether G is path orderable if the number of forcing classes of $\overline{G}$ is bounded by a constant.

For comparability graphs Gallai's results for the general case, i.e. where no assumption on the number of edge classes is made, leads to a polynomial time recognition algorithm. For this he introduced the (by now well-known) concept of modular decomposition and proved that, using this decomposition scheme, the problem of recognizing comparability graphs reduces to the problem of recognizing prime comparability graphs. But what about the recognition of path orderable graphs? Can one extend the decomposition scheme to this problem?

NOT-ALL-EQUAL 3SAT [4]
INSTANCE: Set U of variables, collection $\mathcal{C}$ of clauses over U such that each clause $c \in \mathcal{C}$ has $|c| = 3$.
QUESTION: Is there a truth assignment A for U such that each clause in C has at least one *true* literal and at least one *false* literal?

Remark 4.1 Without loss of generality one can assume that none of the clauses contains more than one literal of a variable.

Theorem 4.2 *Recognition of path orderable graphs is NP-complete.*

Due to space restrictions in this extended abstract we leave out the proof for this theorem but instead sketch the main ideas of the construction.

The proof is done using a transformation from NOT-ALL-EQUAL 3SAT. Given an instance I of NAE 3SAT, a graph G is constructed, which is the complement of a path orderable graph if and only if I is NAE 3 SAT-satisfiable. In particular it is shown that I is NAE 3 SAT-satisfiable if and only if there is an acyclic orientation of G that agrees with the forcing. By Corollary 3.3 this is equivalent with $\overline{G}$ being path orderable.

The basic construction of G is done as follows. For every variable x of U an edge e_x is created (in the following called a *variable edge*) and the two possible orientations of e_x are associated with the two possible values *true* and *false* of x.

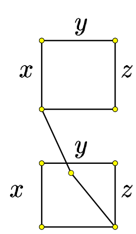

Figure 7: Gadget for clauses.

For each clause $C = x \vee y \vee z$ with literals x, y, z a gadget is constructed, mainly consisting of two C_4s as shown in Figure 7. In each of the C_4s three of the edges (the *base edges*) correspond to the three literals x, y, z of C; a *true* literal of C corresponds to a clockwise orientation of the corresponding base edges in both of the C_4s, whereas a *false* literal corresponds to a counter-clockwise orientation of the corresponding base edges in both C_4s. Furthermore, in each orientation that agrees with the forcing, the fourth edges of the two C_4s will be guaranteed to have opposite orientations in the two C_4s.

The general structure of the connection between variable edges and base edges by forcing paths is shown in Figure 8; for easier understanding, the auxiliary edges and vertices of the forcing paths are omitted in this picture. For a variable edge e_x (see the top of Figure 8) a downwards orientation corresponds to assigning *false* to variable x whereas an upwards orientation corresponds to assigning *true* to x. For each literal x or $\overline{x}$, there is a forcing path of length 4, having e_x and

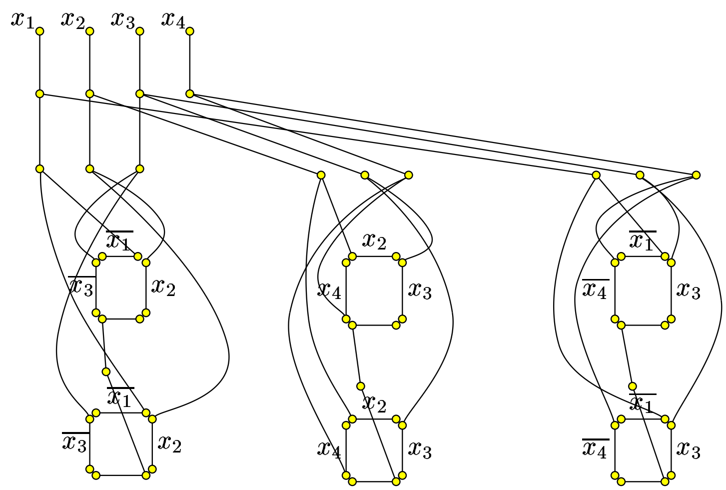

Figure 8: General structure of $\mathrm{K}^*[\overline{G}]$ for the instance $I = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_3 \vee \overline{x_4})$ (auxiliary vertices and edges are omitted).

the corresponding base edge as its end-edges; depending on whether the literal is $\overline{x}$ or x, either the start- or the end-vertex of the base edge (with respect to a clockwise ordering in the C_4) is made the end-vertex of the forcing path.

Now, by Remark 3.13, assigning an upwards orientation to the variable edge e_x results in the desired clockwise orientation of the base edges of each instance of the literal x and a counter clockwise orientation of the base edges of each instance of the literal $\overline{x}$ for any orientation agreeing with the forcing.

All we have to show now is that there is an acyclic orientation of G agreeing with the forcing relation if and only if $\mathcal{C}$ has a NOT-ALL-EQUAL 3SAT satisfying assignment.

By the forcing of the edges and the appropriate knotting of the forcing path from the variable representing edges to the edges representing the literals, each *true* literal in a clause C leads to a clockwise oriented edge and analogously, each *false* literal implies a counter-clockwise oriented edge in the corresponding C_4s. Since every clause has at least one *true* and one *false* literal, each of the C_4s has both an edge that is oriented clockwise and one that is oriented counter-clockwise. Hence, none of the C_4s is cyclically oriented and by the above observations that only the C_4s are possible candidates for oriented cycles, the orientation is acyclic.

Suppose now that there is an acyclic orientation of G that agrees with the forcing relation. We assign to a variable x of U the value *true*, if the edge representing variable x (edges on top of Figure 8) is oriented upwards and *false* otherwise. Since the orientation agrees with the forcing relation, all we have to show is that all of the clauses have at least one *true* and one *false* literal. Suppose there is a clause C, which has only *true* (*false*) literals. By the definition of G and the forcing relation, three edges in each of the C_4s in C's gadget are oriented counter-clockwise (clockwise). Since the "fourth-edges" have opposite orientations in the two C_4s of C, exactly one of the C_4s is oriented cyclically, contradicting the orientation of G being acyclic.

A polynomial time recognition algorithm for strong asteroid free graphs follows from Observation 3.8. Given graph G, the altered knotting graph of $\overline{G}$, $\mathrm{K}^*[\overline{G}]$, can be computed in polynomial time: for each vertex v of G, the blobs of v in G can be computed in $O(n^2)$ time; each vertex has fewer than n blobs. Thus, $\mathrm{K}^*[\overline{G}]$ has $O(n^2)$ vertices and $O(n^2)$ edges (since each edge of $\overline{G}$ corresponds to exactly one edge of $\mathrm{K}^*[\overline{G}]$), and can be constructed in $O(n^3)$ time. To test whether $\mathrm{K}^*[\overline{G}]$ is bipartite can be done in $O(n^2)$ time. Overall, the recognition algorithm requires $O(n^3)$ time.

5 Concluding Remarks

We have defined two graph classes and shown that cocomparability graphs $\subset$ path orderable graphs $\subset$ strong asteroid free graphs $\subset$ AT-free graphs. Furthermore, we have shown that the recognition problem for path orderable graphs is NP-complete, and the recognition of strong asteroid free graphs can be solved in polynomial time. We note that AT-free graph recognition is also in P [1, 7].

Although it is somewhat disappointing that no two of these families are equivalent, these classes may give insight into some open problem complexities on AT-free graphs. By adding graph classes in the hierarchy between cocomparability graphs and AT-free graphs, we may be able to identify more precisely the boundary between polynomial and NP-complete behaviour of some of the problems which are known to be polynomially solvable on cocomparability graphs but either NP-complete or unresolved on AT-free graphs. Examples of such problems include graph colouring, clique cover, clique, and the Hamiltonian path and cycle problems. One step in this direction is the observation that the clique problem is NP-complete for path orderable graphs. This follows from the facts that the complements of triangle-free graphs are contained in path orderable graphs, and the independent set problem is known to be NP-complete on triangle-free graphs [11].

Acknowledgements

The authors wish to thank the Natural Science and Engineering Research Council of Canada for their financial support.

References

[1] D. G. Corneil, S. Olariu, and L. Stewart, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.

[2] R. R. Fulkerson and O. A. Gross, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.

[3] T. Gallai, *Transitiv orientierbare Graphen*, Acta Math. Acd. Sci. Hungar., 18 (1967), pp. 25–66. see the english translation by Maffray and Preissmann.

[4] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, 1979, ch. A9, p. 259.

[5] P. C. Gilmore and A. J. Hoffman, *A characterization of comparability graphs and of interval graphs*, Canad. J. Math., 16 (1964), pp. 539–548.

[6] M. C. Golumbic, C. L. Monma, and W. T. Trotter, *Tolerance graphs*, Discrete Appl. Math., 9 (1984), pp. 157–170.

[7] E. Köhler, *Graphs without asteroidal triples*, PhD thesis, Technische Universität Berlin, 1999.

[8] D. Kratsch and L. Stewart, *Domination on cocomparability graphs*, SIAM J. Discrete Math., 6 (1993), pp. 400–417.

[9] C. G. Lekkerkerker and J. C. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fund. Math., 51 (1962), pp. 45–64.

[10] S. Olariu, *An optimal greedy heuristic to color interval graphs*, Inform. Process. Lett., 37 (1991), pp. 65–80.

[11] S. Poljak, *A note on stable sets and colorings of graphs*, Comment. Math. Univ. Carolinae, 15 (1974), pp. 307–309.

[12] J. P. Spinrad, *Representations of graphs.* book manuscript, 2000.