A note on perfectly orderable graphs

Chinh T. Hoàng

Department of Mathematical Sciences, Lakehead University, Thunder Bay, Ont., Canada P7B 5E1

Received 7 December 1992; revised 20 July 1993

Abstract

We introduce a new class of perfectly orderable graphs that contains complements of chordal bipartite graphs, unions of two threshold graphs, graphs with Dilworth number at most three, and complements of triangulated graphs.

1. Introduction

A natural way to colour the vertices of a graph is:

(i) to impose a linear order < on the vertices, and

(ii) to scan the vertices in this order, assigning to each vertex \( v(j) \) the smallest positive integer assigned to no neighbour \( v(k) \) of \( v(j) \) with \( v(k) < v(j) \).

This heuristic algorithm is called the greedy colouring algorithm, or the sequential colouring algorithm. One may ask the following question: For which ordered graphs does the sequential colouring algorithm deliver an optimal colouring? This question motivated Chvátal [3] to define a "perfect order": an order < is perfect if for each induced subgraph \((H, <)\) of \((G, <)\), the sequential colouring algorithm produces an optimal colouring. Chvátal proved that an order < is perfect if and only if the pair \((G, <)\) does not contain, as induced subgraph, the chordless path on four vertices \(a, b, c, d\) with \(a < b, d < c\) (this ordered subgraph is called an obstruction). A graph is perfectly orderable if it admits a perfect order.

There is a somewhat surprising connection (pointed out first by Chvátal in [4]) between perfectly orderable graphs and a well-known theorem in mathematical programming which we are about to explain. A bipartite graph is chordal if it contains...
no chordless cycle with at least five vertices. A (0,1)-matrix is totally balanced if it does not contain, as a submatrix, the edge-vertex incidence matrix of a cycle of length at least three. A gamma, denoted by $\Gamma$, is the matrix
\[
\begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}.
\]

The following theorem was proved independently by Anstee and Farber [1], Hoffman, Kolen and Sakarovitch [12], and Lubiw [13, 14].

**Theorem 1.** A zero-one matrix $A$ is totally balanced if and only if there is a row and column permutation of $A$ that contains no $\Gamma$ as a submatrix.

Let $B = (X, Y, E)$ be a bipartite graph and let $M_B = (m_{ij})$ be a zero-one matrix whose rows are the vertices of $X$ and columns are the vertices of $Y$ such that $m_{ij} = 1$ if and only if the vertex represented by the $i$th row is adjacent to the vertex represented by the $j$th column. We shall call $M_B$ the bimatrix of $B$. It is easy to verify that (i) there is a row and column permutation of $M_B$ containing no $\Gamma$ if and only if the complement of $B$ is perfectly orderable, and that (ii) $M_B$ is totally balanced if and only if $B$ does not contain any chordless cycle with at least six vertices. Thus, Theorem 1 is equivalent to the following:

**Theorem 2.** The complement of a bipartite graph $B$ is perfectly orderable if and only if $B$ is chordal bipartite.

Note that the “only if” part of the above theorem is trivial, i.e. the complement of any chordless cycle with at least five vertices is not perfectly orderable. The main purpose of this paper is to establish a generalization of Theorem 2. In Section 2, we shall introduce “D-graphs” and show that they are perfectly orderable and contain, as a subclass, all complements of chordal bipartite graphs. In Section 3, we shall show that the class of D-graphs contains unions of two threshold graphs, graphs with Dilworth number at most three, and complements of triangulated graphs (terms not defined here will be defined later). In Section 4, we shall show that a D-graph (respectively, a perfect order on its vertices) can be recognized (respectively, constructed) in $O(nm)$ time.

**2. D-graphs**

Let $X = (x_1, x_2, \ldots, x_k)$ and $Y = (y_1, y_2, \ldots, y_k)$ be two real vectors. We write $X \leq Y$ if $x_i \leq y_i$ for all $i$. Let $M = (m_{ij})$ be a matrix. Then $M_i$ and $M^j$ denote respectively the $i$th row and the $j$th column of $M$. A row $M_i$ is simple if $m_{ij} = m_{ik} = 1$ implies that either $M^j \leq M^i$ or $M^k \leq M^j$. Farber [7] (also see [1, Theorem 3.8]) proved
Theorem 3. Every totally balanced matrix has at least one simple row.

Let $G$ be a graph and let $x, y$ be two vertices of $G$. $N_G(x)$ denotes the set of vertices adjacent to $x$ in $G$. When the context is clear, we shall write $N_G(x) = N(x)$. The vertex $x$ is said to dominate $y$ if every neighbour of $y$ (different from $x$) is a neighbour of $x$, i.e. $N(y) - \{x\} \subseteq N(x)$. We say that $x$ strictly dominates $y$ if $x$ dominates $y$ but $y$ does not dominate $x$, and that $x$ and $y$ are comparable if $x$ dominates $y$ or vice versa. A vertex $x$ is a $d$-vertex if for any edge $yz$ with $y, z \notin N(x)$, $y$ and $z$ are comparable. A graph $G$ is a $D$-graph if each of its induced subgraphs contains a $d$-vertex. Let $B$ be a bipartite graph and let $M_B$ be the bimatrix of $B$. Then a simple row of $M_B$ corresponds to a $d$-vertex of the complement of $B$. Thus Theorem 3 is equivalent to

Theorem 4. Complements of chordal bipartite graphs are $D$-graphs.

Actually, Theorem 4 (and hence Theorem 3) was proved first by Brouwer and Kolen [2] in terms of hypergraphs. A direct proof of Theorem 4 can be found in [15].

We are going to prove the following theorem which generalizes Theorem 2.

Theorem 5. Every $D$-graph is perfectly orderable.

Proof. We are going to describe an algorithm which, given a $D$-graph $G$, delivers a perfect order $\pi$ on the vertices of $G$. We assume that initially the vertices are $v_1, v_2, \ldots, v_n$, and that $\pi(v_i) = 0$ for all $v_i$. The algorithm shall construct an acyclic orientation that corresponds to the order $\pi$ in the following way: $a \rightarrow b$ if and only if $\pi(a) < \pi(b)$ ($x \rightarrow y$ denotes the directed edge with head $y$ and tail $x$).

0. Set $num \leftarrow 1$, $G_{num} \leftarrow G$.
1. Choose a $d$-vertex $x$ of $G_{num}$ with $\pi(x) = 0$ and indegree 0. Set $\pi(x) \leftarrow num$.
   For all neighbours $y$ of $x$ in $G_{num}$ do
     if the edge $xy$ is undirected then set $x \rightarrow y$.
   2. For each undirected edge $v_i v_j$ with $v_i, v_j \notin N_{G_{num}}(x)$ do
      set $v_i \rightarrow v_j$ if $v_i$ strictly dominates $v_j$.
   3. Set $G_{num + 1} \leftarrow G_{num} - x$, $num \leftarrow num + 1$.
      If $G_{num} = \emptyset$ then stop else goto Step 1.

Now, we are going to prove that the above algorithm delivers a perfect order on $G$. First, we note the following:

Fact 1. If an edge $ab$ is assigned the direction $a \rightarrow b$ in Step 2 of some iteration $i$ then $a$ strictly dominates $b$ in the graph $G_i$.

Note that if $a$ dominates $b$ and $b$ is a $d$-vertex then so is $a$. Thus in Step 1, the vertex $x$ always exists provided no (directed) cycle is created. Suppose that a cycle $C$ is created for the first time during Step 2 of some $i$th iteration. Enumerate the vertices of $C$ as $c_1, c_2, \ldots, c_k$ in the cyclic order ($c_j \rightarrow c_{j+1}$ with the subscripts taken modulo $k$). It
is clear that each directed edge \( c_j \to c_{j+1} \) is created in Step 2 of some iteration \( k \) with \( k \leq i \). Fact 1 implies that, in \( G_i \), \( c_j \) dominates \( c_{j+1} \) for all \( j \). Since the dominance relation is transitive, it follows that each vertex of \( C \) dominates any other vertex of \( C \), but no vertex of \( C \) strictly dominates another vertex of \( C \). Let \( c_r \to c_{r+1} \) be the directed edge of \( C \) that is created in the \( i \)th iteration. Then Fact 1 implies that \( c_r \) strictly dominates \( c_{r+1} \), a contradiction.

Now, suppose that the algorithm creates an obstruction for the first time during the \( i \)th iteration. Let the vertices of this obstruction (in \( G_i \)) be \( a, b, c, d \) with edges \( ab, bc, cd \) (and no other edge) and the relations \( \pi(a) < \pi(b), \pi(d) < \pi(c) \). We know that in \( G_i \), \( a \) (respectively, \( d \)) cannot dominate \( b \) (respectively, \( c \)).

Without loss of generality, we may assume that the directed edge \( a \to b \) is created in Step 1 or Step 2 of the \( i \)th iteration. In the former case, we know that the directed edge \( d \to c \) is created in Step 2 of some \( j \)th iteration with \( j < i \). But then Fact 1 implies that in \( G_i \), \( d \) dominates \( c \), a contradiction. In the latter case, we must have \( a \) dominating \( b \), a contradiction. \( \square \)

A graph \( G \) is minimally non-perfectly orderable (MNPO) if \( G \) is not perfectly orderable but every proper induced subgraph of \( G \) is. Properties of MNPO graphs were studied in [9, 10]. In view of Theorem 5 one might ask whether no MNPO graph can contain a \( d \)-vertex. However, we are going to show that

there is a MNPO graph containing a \( d \)-vertex. (1)

Consider the graph \( G \) shown in Fig. 1. It is easy to verify that \( G \) is perfectly orderable but every proper induced subgraph of \( G \) is. Now, let \( F \) be the graph obtained from \( G \) by adding a new vertex \( x \) and joining \( x \) to every vertex of \( G \) except \( a \) and \( b \). Then \( x \) is a \( d \)-vertex of \( F \). Since any proper induced
subgraph – containing $a$ and $b$ – of $G$ admits a perfect order $< \nless a < b$, it follows that every proper induced subgraph of $F$ is perfectly orderable with $x < y$ for all $y$ in $G$. Now, suppose that $F$ admits a perfect order $<$ on its vertices. From the previous paragraph we know that $b < j$ and $b < a$. If $x < d$ then we have an obstruction with vertices $x,d,a,b$ if $d < x$ then we have an obstruction with vertices $d,x,j,b$. Thus $F$ is MNPO and $(1)$ is justified.

3. Subclasses of D-graphs

It turns out that Theorem 5 generalizes two previously known theorems proved by Chvátal, Hoâng, Mahadev and de Werra [5]. In order to describe these two theorems we need to introduce a few definitions. The Dilworth number of a graph $G$ is the largest number of pairwise incomparable vertices of $G$. Threshold graphs are graphs with Dilworth number one. It is easy to prove that $G$ is a threshold graph if and only if $G$ does not contain as induced subgraphs a $P_4$ (the chordless path on four vertices), or a $C_4$ (the chordless cycle on four vertices), or a $2K_2$ (the complement of a $C_4$). It was proved in [5] that unions of two threshold graphs are perfectly orderable. Using the notion of a D-graph, we are going to establish a generalization of this result.

**Theorem 6.** Let $G_1$ be a threshold graph and let $G_2$ be a graph containing no induced $P_4$ and no induced $C_4$. Then the union of $G_1$ and $G_2$ is a D-graph.

**Proof.** Let $G = G_1 \cup G_2$. Let $x$ be the vertex that dominates any other vertex of $G_1$. We may assume that $x$ is not a d-vertex of $G$, for otherwise we are done. Consider any edge $yz$ of $E(G)$ with $y, z \notin N_G(x)$. Then clearly we have $yz \in E(G_2) - E(G_1)$. Suppose that $y$ and $z$ are incomparable, i.e. there are vertices $y', z'$ with $y'y', z'z' \in E(G)$ and $y'z', z'y' \notin E(G)$. Since $x$ dominates any other vertex in $G_1$, we know that $yy', zz' \in E(G_2) - E(G_1)$. But then $G_2$ contains a $P_4$ or $C_4$, a contradiction. 

It was proved in [5] (see also [11]) that a graph $G$ is perfectly orderable whenever the Dilworth number of $G$ is at most three. We are going to establish a stronger result.

**Theorem 7.** If the Dilworth number of a graph $G$ is at most three then $G$ is a D-graph.

**Proof.** By the Dilworth Theorem, the vertices of $G$ can be partitioned into three sets $A, B, C$ such that the vertices of $A$ (respectively $B, C$) are pairwise comparable. Enumerate the vertices of $A$ (respectively $B$ and $C$) as $a_1, a_2, \ldots$ (respectively $b_1, b_2, \ldots$ and $c_1, c_2, \ldots$) such that $a_i$ (respectively $b_i$ and $c_i$) dominates $a_j$ (respectively $b_j$ and $c_j$) whenever $i < j$. We may assume that $a_1$ is not a d-vertex of $G$. Thus there are incomparable vertices $b, c$ such that $bc \in E(G)$ and $b, c \notin N_G(a_1)$. By our choice of $a_1$ we have $b, c \in B \cup C$. Without loss of generality, assume $b \in B$. The definition of $B$ implies $c \in C$. Let $h = b_1$ and $e = c_j$. 

Now, consider the vertex $c_1$. We may assume that $c_1$ is not a $d$-vertex. By a similar argument, we see that there are incomparable vertices $b_k, a_p$ such that $b_k \in B, a_p \in A, b_k a_p \notin E(G)$, and $b_k, a_p \notin N(c_1)$. Since $c_1$ dominates $c_j$, we have $b_k c_1 \in E(G)$ (note that we do not assume $c_j \neq c_1$). Since $b_k c_1 \notin E(G)$, we know that $b_k$ dominates $b_k$. Since $b_k a_1 \notin E(G)$, we have $b_k a_1 \notin E(G)$. But then $a_1$ does not dominate $a_p$, a contradiction. \qed

A graph is triangulated (or chordal) if it contains no induced cycle with at least four vertices. Dirac [6] proved that every triangulated graph contains a vertex whose neighbourhood is a clique. Thus, the class of D-graphs contains all complements of triangulated graphs.

4. Recognizing D-graphs

Middendorf and Pfeiffer [16] proved that recognizing perfectly orderable graphs is NP-complete. However, D-graphs can be recognized in polynomial time. A naive algorithm works as follows. Given a graph $G$, determine whether $G$ contains a $d$-vertex $x$, if no such vertex exists then $G$ is not a D-graph, if there is such a vertex $x$, then remove $x$ from $G$ and repeat this process until all vertices are removed from $G$. (This algorithm is correct since any $d$-vertex of $G$ remains a $d$-vertex of any induced subgraph of $G$.) Given an edge $yz$, we can determine in $O(n)$ time whether $y$ and $z$ are comparable. (Here, as usual, $n$ and $m$ denote respectively the number of vertices and the number of edges of $G$.) Thus, a $d$-vertex can be found in $O(nm)$ time. And so the naive algorithm has complexity $O(n^2 m)$. By refining this idea, we obtain an $O(nm)$ algorithm for recognizing D-graphs. We shall briefly describe this algorithm and leave the implementation detail to the reader.

For simplicity, suppose the vertices of $G$ are $1, 2, \ldots, n$. Assume that the graph $G$ is given by its adjacency lists, i.e. for each vertex $x$, there is a linked list $N(x)$ containing the neighbours of $x$. It is easy to see that in $O(m)$ time, we can sort all $n$ lists $N(x)$ in increasing order (for details, see [8]).

For each vertex $x$, we shall maintain a list $B(x)$ of “bad edges”, i.e. the edges $yz$ such that $y, z \notin N(x)$ and $y$ and $z$ are incomparable. It is easy to see that the $n$ sets $B(x)$ can be computed in $O(nm)$ time.

For any two adjacent vertices $i, j$, we introduce a pointer $p(i, j)$ which points to the smallest vertex (number) $k$ in the list $N(i)$ such that $k \notin N(j) \cup \{j\}$. By $p(i, j).e$ we denote the element (vertex) of $N(i)$ pointed to by $p(i, j)$. The pointer $p(j, i)$ is defined similarly. As usual, if a pointer cannot be defined then we assigned the value $nil$ to it. Thus, $i$ dominates $j$ (the edge $ij$ is “good”) if and only if $p(j, i) = nil$.

In order to be able to maintain $p(i, j)$ properly, we shall need a “shadow” pointer $p'(i, j)$ that points to the smallest element $l$ of $N(j)$ such that $l \neq i$ and $l > p(i, j).e$. The reader may note that we can initialize $p(i, j)$ and $p'(i, j)$ by first setting $p(i, j)$ ($p'(i, j)$) to point to the first element of $N(i)$ ($N(j)$) and then moving $p(i, j)$ through $N(i)$, $p'(i, j)$ through $N(j)$, while comparing $p(i, j).e$ with $p'(i, j).e$ (and stopping when appropriate).
Assume that the pointers \( p(i, j) \), \( p'(i, j) \) and the sets \( B(x) \) are properly initialized. The algorithm is as follows:

0. Choose a vertex \( x \) of \( G \) with \( B(x) = \emptyset \). If no such vertex \( x \) exists then return "\( G \) is not a D-graph".
1. For each neighbour \( y \) of \( x \) such that for some \( z \), \( p(y, z).e = x \), do the following:
   - Remove \( x \) from \( N(y) \) and update \( p(y, z) \) and \( p'(y, z) \) (set \( p(y, z) \) to point to the immediate successor of \( x \) in the list \( N(y) \), and move \( p(y, z) \) and \( p'(y, z) \) further down the lists \( N(y), N(z) \) if necessary). If \( p(y, z) = \text{nil} \) (the edge \( yz \) becomes "good") then remove the edge \( yz \) from any set \( B(r) \) that contains \( yz \).
2. Remove all occurrences of \( x \) from all adjacency lists and update those pointers \( p'(y, z) \) with \( p'(y, z).e = x \). Remove \( x \) from \( G \). If \( G \) becomes empty then return "\( G \) is a D-graph", else goto Step 0.

If for each vertex \( x \) we maintain a linked list \( P(x) \) containing the pointers \( p(y, z) \) that point to \( x \) (\( p(y, z).e = x \)), then in Step 1 the vertices \( y, z \) can be located easily. (For each \( x \), \( P(x) \) contains at most \(|N_G(x)| \) elements.) The total work involving each pointer \( p(i, j) \) or \( p'(i, j) \) is \( O(\Delta) \), where \( \Delta \) denotes the maximum degree of \( G \), since over the whole history of the algorithm each \( p(i, j) \) (\( p'(i, j) \)) may move from the beginning of \( N(i) \) (\( N(j) \)) to the end of it. Since there are \( O(m) \) pointers, the complexity of the algorithm is \( O(\Delta m) \).

Finally, we note that given a D-graph \( G \), a perfect order on the vertices of \( G \) can be constructed in \( O(\Delta m) \) time by implementing the algorithm described in Theorem 5 in a similar way.

5. An open problem

Let \( F \) be a graph. A graph \( G \) is said to be \( F \)-free if \( G \) does not contain \( F \) as an induced subgraph. In view of Theorem 6, one might consider the following three possible generalizations: To prove that the union of two graphs \( G_1 \) and \( G_2 \) is perfectly orderable whenever \( G_1 \) and \( G_2 \) are (i) \( P_4 \)-free and \( C_4 \)-free, or (ii) \( P_4 \)-free and \( 2K_2 \)-free, or (iii) \( C_4 \)-free and \( 2K_2 \)-free.

The complement of the chordless cycle on six vertices shows that (i) and (iii) are false. We conjecture that (ii) is true:

**Conjecture 1.** Let \( G_1 \) and \( G_2 \) be two graphs that are \( P_4 \)-free and \( 2K_2 \)-free. Then the union of \( G_1 \) and \( G_2 \) is perfectly orderable.

**References**


