



Strongly orderable graphs A common generalization of strongly chordal and chordal bipartite graphs[☆]

Feodor F. Dragan

*Universität Rostock, Fachbereich Informatik, Lehrstuhl für Theoretische Informatik,
Albert-Einstein-Str. 21, D-18051 Rostock, Germany*

Received 12 September 1997; revised 11 March 1998; accepted 9 March 1999

Abstract

In this paper those graphs are studied for which a so-called strong ordering of the vertex set exists. This class of graphs, called strongly orderable graphs, generalizes the strongly chordal graphs and the chordal bipartite graphs in a quite natural way. We consider two characteristic elimination orderings for strongly orderable graphs, one on the vertex set and the second on the edge set, and prove that these graphs can be recognized in $O(|V| + |E|)|V|$ time. Moreover, a special strong ordering of a strongly orderable graph can be produced in the same time bound. We present variations of greedy algorithms that compute a minimum coloring, a maximum clique, a minimum clique partition and a maximum independent set of a strongly orderable graph in linear time if such a special strong ordering is given. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Strongly chordal graphs; Chordal bipartite graphs; Strong ordering; Simple elimination ordering; Bisimplicial edge without vertex elimination ordering; Recognition; Maximum clique; Minimum coloring; Greedy algorithms

1. Introduction

Elimination orderings of graphs are an algorithmically powerful tool. Chordal and strongly chordal graphs represent good examples for this. Recall that a graph G is called *chordal* if it has no induced cycles of length greater than three. These graphs have a nice elimination ordering, called simplicial ordering, which makes it possible to solve in linear time a number of optimization problems; among them minimum coloring, maximum independent set, maximum clique and minimum clique partition [13,22]. The

[☆] Research supported by the DFG. On leave from the Universitatea de stat din Moldova, Chişinău.
E-mail addresses: dragan@informatik.uni-rostock.de (F.F. Dragan)

well-known domination problem remains NP-complete even for this restricted class. Strongly chordal graphs were introduced by Farber [11] in order to have a subclass of chordal graphs where the domination problem can be solved efficiently. A graph G is called *strongly chordal* if it admits a *strong simplicial ordering* [11], i.e. an ordering σ of the vertices of G such that

1. if $a < \{b, c\}$ and $ab, ac \in E$ then $bc \in E$,
2. if $ab, ac, bd \in E$, $a < d$ and $b < c$ then $cd \in E$.

(We write $a < b$ whenever in a given ordering σ vertex a has a smaller number than vertex b .)

This ordering allows to solve various domination-like problems as well as the Steiner tree problem and the maximum matching problem in linear time [12,5,24,4,7].

If an ordering σ satisfies only the first condition then σ is called *simplicial ordering*. An ordering satisfying only the second condition we will call *strong ordering*. It is well known that a graph G has a simplicial ordering if and only if G is chordal (cf. [14]). Moreover, a simplicial ordering of a (chordal) graph can be computed in linear time using Lexicographic breadth first search (LexBFS, [22]). Unfortunately, to date the fastest method — doubly lexical ordering of the (closed) neighborhood matrix [20] — producing a strong simplicial ordering of a strongly chordal graph G takes $O(|E|\log|V|)$ [21] or $O(|V|^2)$ [23] time.

It follows from the results in [7,9] that a strong ordering is very useful for computing a maximum matching of a graph. If a graph G has a strong ordering and a strong ordering is given, then a maximum matching of G can be found in linear time. Among the graphs having strong orderings are also *chordal bipartite* graphs, i.e. bipartite graphs having no induced cycles of length greater than four (cf. [14]). It is well known (see [11,1,20]) that chordal bipartite graphs are exactly those bipartite graphs which have a strong ordering, and that this ordering can be computed in $O(\min\{|E|\log|V|, |V|^2\})$ time using doubly lexical ordering of the bipartite adjacency matrix.

In [8] Dahlhaus has shown that a strong ordering is useful also for optimal coloring of graphs. A $O(|E|+|V|)\log|V|$ time algorithm is presented which computes a minimum coloring for a graph with a strong ordering if such an ordering is given. The paper [8] does not give an answer to how one can check whether a given graph has a strong ordering and how to compute such an ordering efficiently if it exists.

In this paper we give two characterizations of the graphs for which a strong ordering exists. We call them *strongly orderable graphs* (note that in [8] they were called *generalized strongly chordal graphs*). We define new elimination orderings, one on the vertex set and the second on the edge set, which generalize the known characteristic elimination orderings of strongly chordal graphs and chordal bipartite graphs. It turns out that both these new orderings are characteristic for strongly orderable graphs. So, this class of graphs generalizes the strongly chordal graphs and the chordal bipartite graphs in quite a natural way. Using those characterizations we give an $O(|V|+|E|)|V|$ time algorithm for recognizing strongly orderable graphs which also produces a special strong ordering of such a graph. In the last section we present variations of greedy algorithms that compute a minimum coloring, a maximum clique, a minimum clique

partition and a maximum independent set of a strongly orderable graph in linear time if such a special strong ordering is given.

Notice that in [9] we characterize strongly orderable graphs for which a strong ordering can be found in linear time by LexBFS. Moreover, there we describe also a subclass of strongly orderable graphs where a strong ordering can be found by doubly lexical ordering of the neighborhood matrix in $O(\min\{|E|\log|V|, |V|^2\})$ time.

2. Strong orderings and related elimination orderings

All graphs $G = (V, E)$ in this note are finite, undirected and simple (i.e. without loops and multiple edges). For $S \subseteq V$ let $G(S)$ be the subgraph of G induced by S . The (*open*) *neighborhood* of a vertex v of G is the set $N_G(v) = \{u \in V: uv \in E\}$ and the *closed neighborhood* is $N_G[v] = N_G(v) \cup \{v\}$. If no confusion can arise we will omit the index G .

Let $\sigma = (v_1, v_2, \dots, v_n)$ be an ordering of the vertex set of a graph G . In what follows, we will write $a < b$ whenever in a given ordering σ vertex a has a smaller number than vertex b . Moreover, $a < \{b_1, \dots, b_k\}$ is an abbreviation for $a < b_i, i = 1, \dots, k$.

Definition 1. An ordering σ of the vertex set of a graph G is a strong ordering if for every four vertices a, b, c, d of G such that $ab, ac, bd \in E, a < d$ and $b < c$ we have $cd \in E$.

Definition 2. A graph G is strongly orderable if it admits a strong ordering.

The following lemma indicates how to check whether a given ordering σ is a strong ordering in linear time. The idea is borrowed from [20].

Lemma 3. An ordering σ of the vertex set of a graph G is a strong ordering if and only if for every edge $ab \in E$ we have both

- (a) the smallest neighbor c of a with $c > b$ is adjacent to the smallest neighbor d of b with $d > a$ and $d \neq c$, and
- (b) the smallest neighbor d of b with $d > a$ is adjacent to the smallest neighbor c of a with $c > b$ and $c \neq d$.

Proof. We have to prove the ‘if’ part only. Assume that σ is not a strong ordering of G . Choose four vertices a, b, c and d such that $ab, ac, bd \in E, cd \notin E, a < d$ and $b < c$ which minimize $s(a, b, c, d) = (d - a) + (c - b)$ (in arithmetic operations we use the numbers of vertices in σ). Since $cd \notin E$ neighbor c of a with $c > b$ or neighbor d of b with $d > a$ is not smallest in σ . Without, loss of generality, suppose that there is a neighbor c' of a such that $b < c' < c$. We claim that $c' = d$. If $c'd \in E$ then the existence of vertices a, c', c, d with $s(a, c', c, d) < s(a, b, c, d)$ contradicts to the choice of a, b, c, d . If $c'd \notin E$ and $c' \neq d$ then now the existence of vertices a, b, c', d with $s(a, b, c', d) < s(a, b, c, d)$ contradicts to the choice of a, b, c, d . Hence $c' = d$, and

we conclude that $da \in E$, $c > d$ and c is the smallest in σ neighbor of a such that $c \neq d$ and $c > b$. Repeating the arguments above we obtain that d should be the smallest in σ neighbor of b with $d > a$. Since $cd \notin E$ a contradiction to the requirement of the lemma arises. \square

Now, if we are given a graph G with an ordering σ in the form of an ordered adjacency list, then verification of whether σ is a strong ordering of G can be performed in $O(|V| + |E|)$ time (for details see [20]). Below we will show that a strong ordering of a strongly orderable graph G can be produced in $O(|V| + |E|)|V|$ time.

2.1. Quasi-simple elimination orderings

For a given ordering σ of the vertex set of a graph G , by $G_{\geq v}$ we denote a subgraph of G induced by the set $\{u \in V : u \geq v\}$. Strongly chordal graphs can be characterized also by another elimination scheme [11]. A vertex v of a graph G is called *simple* in G if for all $x, y \in N_G(v)$, $N_G[x] \subseteq N_G[y]$ or $N_G[y] \subseteq N_G[x]$ holds, i.e. $\{N_G[x] : x \in N_G[v]\}$ is linearly ordered by inclusion. A *simple elimination ordering* of a graph G is an ordering σ such that vertex v is simple in $G_{\geq v}$ for all $v \in V$. In [11] it is shown that G is strongly chordal if and only if G admits a simple elimination ordering. Note that a strong simplicial ordering is a simple elimination ordering σ such that for all $v \in V$ in $G_{\geq v}$ we have $N_{G_{\geq v}}[x] \subseteq N_{G_{\geq v}}[y]$ whenever $x < y$ in σ and $x, y \in N_{G_{\geq v}}(v)$. Here we define a new ordering of the vertex set of a graph which has similar relation to strong orderings.

A vertex v is *dominated* by a vertex u (or u *dominates* v) in a graph G if $N_G(v) \setminus \{u\} \subseteq N_G(u) \setminus \{v\}$ holds. If additionally there exists a vertex w adjacent to u but not to v , then we say that v is *strictly dominated* by u (or u *strictly dominates* v) in G . If v dominates u and is dominated by u , i.e. $N_G(v) \setminus \{u\} = N_G(u) \setminus \{v\}$, then we say that the vertices v and u are *twins*. Two vertices v and u of a graph G are *comparable* in G if one of them dominates the other. Otherwise we will say that v and u are *noncomparable*.

Definition 4. A vertex v of a graph G is *quasi-simple* in G if every two vertices of $N_G(v)$ are comparable in G . A *quasi-simple elimination ordering* of a graph G is an ordering σ such that vertex v is quasi-simple in $G_{\geq v}$ for all $v \in V$.

Lemma 5. Let G be a strongly orderable graph. Any strong ordering of G is a quasi-simple elimination ordering of G .

Proof. Let σ be a strong ordering of G . Consider an arbitrary vertex a of G and let b and c be neighbors of a in $G_{\geq a}$. Without loss of generality, assume that $b < c$ in σ . Then, since σ is a strong ordering of G , every neighbor d of b with $a < d$ is adjacent to c , i.e. in $G_{\geq a}$ vertex b is dominated by c . Consequently, each vertex a of G is quasi-simple in $G_{\geq a}$. By definition, σ is a quasi-simple elimination ordering of G . \square

For a vertex $v \in V$ let $deg_G(v)$ be the degree of v in G . The next lemma gives a nice criterion for checking whether a vertex v of G is quasi-simple in G .

Lemma 6. *Let x_1, x_2, \dots, x_k ($k = deg_G(v)$) be an ordering of neighbors of a vertex v such that $deg_G(x_1) \leq deg_G(x_2) \leq \dots \leq deg_G(x_k)$. Then v is quasi-simple in G if and only if both of the following hold:*

- (a) x_j is dominated by x_{j+1} , for every $j = 1, \dots, k - 1$, and
- (b) $x_j x_{j+1} \in E$ if $x_j x_{j-1} \in E$, for every $j = 2, \dots, k - 1$.

Proof. Assume that v is quasi-simple in G . If x_j is not dominated by x_{j+1} for some $j \in \{1, \dots, k - 1\}$ then there must be a vertex adjacent to x_j and not to x_{j+1} . On the other hand, we have $deg_G(x_{j+1}) \geq deg_G(x_j)$. Hence there exists a vertex which adjacent to x_{j+1} and not to x_j . This means that neighbors x_{j+1} and x_j of v are noncomparable in G , which is impossible. Thus, for every $j = 1, \dots, k - 1$, x_j is dominated by x_{j+1} in G . Now assume that x_j is adjacent to x_{j-1} for some $j = 2, \dots, k - 1$. Since x_j is dominated by x_{j+1} and x_{j-1} is dominated by x_j we immediately conclude that $x_{j-1} x_{j+1} \in E$ and hence $x_j x_{j+1} \in E$.

To prove the converse we will show that for every two neighbors x_i and x_j of v with $i < j$, $N_G(x_i) \subset N_G[x_j]$ holds. Let l be the smallest index such that $i \leq l < j$ and $x_l x_{l+1} \in E$ (if l is not defined we set $l = j$). From the requirements of the lemma we obtain $x_p x_{p+1} \in E$, for all $p \in \{l, \dots, j - 1\}$, and

$$N_G(x_i) \subseteq N_G(x_{i+1}) \subseteq \dots \subseteq N_G(x_l) \subset N_G[x_l] \subseteq N_G[x_{l+1}] \subseteq \dots \subseteq N_G[x_j].$$

That is every two neighbors of v are comparable in G . \square

Now assume that a graph G admits a quasi-simple elimination ordering. In what follows, we will use the following special quasi-simple elimination ordering of G .

Procedure LexQSEO (Lexicographic quasi-simple elimination ordering). *Order the vertices of a graph $G=(V,E)$ by assigning numbers from 1 to $|V|$. Assign the number $k + 1$ to a vertex v (as yet unnumbered) which*

- (1) *is quasi-simple in the graph $G(V \setminus \{v_1, \dots, v_k\})$, where v_1, \dots, v_k are already numbered vertices,*
- (2) *has the minimum number of neighbors in $V \setminus \{v_1, \dots, v_k\}$ among all vertices satisfying (1), and*
- (3) *has lexicographically smallest vector $s(v) = (s_i(v) : i = k, k - 1, \dots, 1)$ among all vertices satisfying (2), where $s_i(v) = 1$ if v is adjacent to v_i , and $s_i(v) = 0$ otherwise.*

The running time of this procedure for a graph $G=(V,E)$ can be estimated as follows. First we compute the matrix $R(G) = (r_{xy})_{x,y \in V}$, where $r_{xy} = |N_G(x) \setminus N_G[y]|$. For a fixed vertex x , the row of $R(G)$ corresponding to x can be completed in $O(deg_G(x)|V|)$ time: put initially $r_{xy} := deg_G(x)$ for all $y \in V$ and then for every $v \in N_G(x)$ decrease the value of r_{xy} by 1 if and only if $v \in N_G[y]$. So, the whole matrix $R(G)$ can be computed in $O(|V||E|)$ time. Evidently, a vertex x is dominated by a vertex y in G if and only

if $r_{xy} = 0$. Hence, having the matrix $R(G)$ one can check in constant time whether a vertex x is dominated by a vertex y in G .

Now assume that we have already numbered k ($k \geq 0$) vertices of the graph G and we want to select a vertex in the graph $H^k = G(V^k)$, where $V^k = V \setminus \{v_1, \dots, v_k\}$ and $V^0 = V$, to number it by $k + 1$. Assume also, that we are given the matrix $R(H^k)$ (below we will show how to get the matrix $R(H^{i+1})$ from the matrix $R(H^i)$ efficiently).

Arrange the vertices of V^k in increasing order with respect to the parameter $deg_{H^k}(\cdot)$. This can be done using bucket sort in $O(|V| + |E|)$ time, obtaining the ordering $\tau = (x_1, x_2, \dots, x_{|V^k|})$. Using this ordering τ and the standard linear time technique for obtaining an ordered adjacency list from a nonordered adjacency list of a graph (see e.g. [14]), for each vertex v , we get an ordered representation $(x_{i_1}, x_{i_2}, \dots, x_{i_p})$ of its neighbors in the graph H^k .

Then, having this structure and the matrix $R(H^k)$, we can check whether a vertex v is quasi-simple in H^k using only $deg_G(v)$ time. Note that v is quasi-simple if and only if x_{i_j} is dominated by $x_{i_{j+1}}$ for every $j = 1, \dots, deg_{H^k}(v) - 1$, and $x_{i_j} x_{i_{j+1}} \in E$ if $x_{i_j} x_{i_{j-1}} \in E$ for every $j = 2, \dots, deg_{H^k}(v) - 1$ (see Lemma 6). Consequently, in $O(|V| + |E|)$ time we can select all quasi-simple vertices of the graph H^k and choose among them all vertices with minimum degree in H^k . A vertex v which has lexicographically smallest vector $s(v)$ can be found in $O(\sum_{i=1}^k deg_G(v_i)) = O(|V| + |E|)$ time. By the procedure, the vertex v will be numbered by $k + 1$, i.e. $v_{k+1} := v$.

It remains to note that the matrix $R(H^{k+1})$ for the graph $H^{k+1} = G(V^k \setminus \{v_{k+1}\})$ can be obtained in $O(deg_G(v_{k+1})|V|)$ time from the matrix $R(H^k)$ in the following way: for all $x, y \in V^k \setminus \{v_{k+1}\}$ with $v_{k+1}x \in E$, decrease the value of r_{xy} by 1 if and only if $v_{k+1}y \notin E$.

Summarizing, the total running time of the Procedure LexQSEO is

$$O(|V||E|) + O(|V| + |E|)|V| + \sum_{v \in V} O(deg_G(v)|V|) = O(|V| + |E|)|V|.$$

We will prove later that any lexicographic quasi-simple elimination ordering of a graph G is a strong ordering of G . But first we present an important auxiliary result.

Lemma 7. *Let σ be a lexicographic quasi-simple elimination ordering of a graph G . If a vertex c is strictly dominated by a vertex b in some graph $G_{\geq a}$, where $a \leq \{b, c\}$, then $c < b$ in σ .*

Proof. Assume that the lemma is false, and select a counterexample in which b is as large as possible; that is

- (i) the vertex b has a smaller number than the vertex c in σ , i.e. $b < c$, but
- (ii) for every $b' > b$, if a vertex c' strictly dominated by b' in $G_{\geq a'}$ with $a' \leq \{b', c'\}$, then $c' < b'$ holds.

Since c is strictly dominated by b in $G_{\geq a}$ and $a \leq b$ holds, in the graph $G_{\geq b}$ either c is still strictly dominated by b or these vertices are twins.

Case 1: c is strictly dominated by b in $G_{\geq b}$. Consider in σ the smallest vertex c^* such that $b < c^*$ and c^* is strictly dominated by b in $G_{\geq b}$. Evidently, $c^* \leq c$.

We claim that c^* is quasi-simple in the graph $G_{\geq b}$.

If this is not the case, we will find in $G_{\geq b}$ two noncomparable neighbors u and v of c^* . From $N_{G_{\geq b}}(c^*) \subset N[b]$ we conclude that $u, v \in N[b]$. Moreover, since b is quasi-simple in $G_{\geq b}$, but u and v are noncomparable in this graph, the vertex b must coincide with one of these vertices, say $b = u$.

The vertices $b = u$ and v are noncomparable in the graph $G_{\geq b}$. Hence, there exist two vertices x and z of $G_{\geq b}$ (other than b and v) such that $xb, zv \in E$ and $xv, zb \notin E$ hold. Since b is quasi-simple in $G_{\geq b}$ and $x, c^* \in N(b)$, the vertices x and c^* are comparable in $G_{\geq b}$. From $vc^* \in E$ and $xv \notin E$ we conclude that the vertex x is strictly dominated by c^* (and hence, by b) in $G_{\geq b}$. Moreover, assumption (ii) yields $x < c^*$. Thus, we obtain that the vertex x is strictly dominated by b in $G_{\geq b}$ and fulfills the condition $b < x < c^*$. This contradiction to the choice of the vertex c^* proves that c^* is quasi-simple in $G_{\geq b}$.

Therefore, both vertices b and c^* are quasi-simple in $G_{\geq b}$, but the vertex c^* has less neighbors in $G_{\geq b}$ than the vertex b . Since b was a quasi-simple vertex of the graph $G_{\geq b}$ with the minimum number of neighbors in $G_{\geq b}$, a contradiction arises. (Recall that σ is a lexicographic quasi-simple elimination ordering of G .)

Case 2: c and b are twins in $G_{\geq b}$. Now we claim that the vertex c is quasi-simple in $G_{\geq b}$. Indeed, since b is quasi-simple in $G_{\geq b}$ any two neighbors of c in $G_{\geq b}$, different from b , are comparable in $G_{\geq b}$ as neighbors of b . So, let $bc \in E$ and x be an arbitrary neighbor of c with $x > b$. If x and b are noncomparable in $G_{\geq b}$ then x and c cannot be comparable as well due to $N_{G_{\geq b}}[c] = N_{G_{\geq b}}[b]$.

Hence, both vertices b and c are quasi-simple in $G_{\geq b}$ and both have the same neighborhood in this graph. But recall that c was strictly dominated by b in the graph $G_{\geq a}$. Therefore, the largest vertex v of the set $(N_G(c) \cup N_G(b)) \setminus (N_G(c) \cap N_G(b))$ fulfills $v < b$ and is adjacent to b and not to c . This means that before assigning a number to vertex b the vector $s(c)$ was lexicographically smaller than the vector $s(b)$ (see condition (3) in the Procedure LexQSEO). Since σ is a lexicographic quasi-simple elimination ordering of G , again a contradiction arises.

The contradictions obtained in both cases show that the assumption $b < c$ is wrong. □

Lemma 8. Any lexicographic quasi-simple elimination ordering of a graph G is a strong ordering of G .

Proof. Let σ be a lexicographic quasi-simple elimination ordering of G and assume that it is not a strong ordering. Then we will find in G four vertices a, b, c and d such that $ab, ac, bd \in E$, $a < b < c$ and $a < d$, but $cd \notin E$. Since a is quasi-simple in $G_{\geq a}$ any two neighbors of a in $G_{\geq a}$ are comparable. From $a < \{b, c, d\}$, $b, c \in N(a)$, $db \in E$ and $dc \notin E$ we conclude that the vertex c is strictly dominated by b in this graph $G_{\geq a}$. Hence, by Lemma 7, in σ vertex c must have a smaller number than the vertex b , contradicting $b < c$. □

Since a quasi-simple vertex v of G is quasi-simple in any induced subgraph of G which includes v from Lemmas 5 and 8 we derive

Theorem 9. *A graph G is strongly orderable if and only if it admits a quasi-simple elimination ordering.*

Hence, the Procedure LexQSEO can be used for recognizing strongly orderable graphs as well as for computing a strong ordering of such a graph.

Corollary 10. *It can be tested in $O(|V| + |E|)|V|$ time whether a graph $G = (V, E)$ is strongly orderable.*

Corollary 11. *For a given strongly orderable graph $G = (V, E)$ a strong ordering can be produced in $O(|V| + |E|)|V|$ time.*

2.2. Simplicial-edge-without-vertex elimination orderings

A vertex v of a graph G is called *simplicial* in G if every two neighbors of v are adjacent, i.e. the neighborhood $N_G(v)$ of v induces a complete subgraph of G . Using this notion one can give an alternative definition of simplicial orderings. An ordering σ is a *simplicial ordering* of G if vertex v is simplicial in $G_{\geq v}$ for all $v \in V$. Note also that a vertex is simple if and only if it is quasi-simple and simplicial as well. In [15] the notion of simpliciality was adapted for bipartite graphs. An edge ab of a bipartite graph G is called *bisimplicial* if $N_G(a) \cup N_G(b)$ induces a complete bipartite subgraph of G . Analogously to this we define a simplicial edge of an arbitrary graph.

Definition 12. An edge ab of a graph G is simplicial in G if every two distinct vertices $v \in N_G(a)$ and $u \in N_G(b)$ are adjacent in G .

We will need also the following notion of a *simplicial-edge-without-vertex elimination ordering*. It generalizes the known notion of a *bisimplicial-edge-without-vertex elimination ordering* (see [10,18,3,17]) and refers to an edge elimination ordering such that no vertices are deleted in the process.

Definition 13. Let $G = (V, E)$ be a graph, (e_1, \dots, e_m) be an ordering on E and $G_i = (V, E_i)$ be a subgraph of G with vertex set V and edge set $E_i = \{e_j : j \geq i\}$. The ordering (e_1, \dots, e_m) is a simplicial-edge-without-vertex elimination ordering for G if each edge e_i is simplicial in G_i .

It is known from [10,3] (see also [18,17]) that a bipartite graph G is chordal bipartite if and only if G has a bisimplicial-edge-without-vertex elimination ordering. Here we will show that strongly orderable graphs are exactly those graphs for which a simplicial-edge-without-vertex elimination ordering exists.

Let $\sigma = (v_1, \dots, v_n)$ be an ordering on the vertex set V of a graph $G = (V, E)$. For every edge e of G , denote by $left(e)$ ($right(e)$) the number in σ of its smaller (respectively, larger) endvertex. A natural way to construct an ordering $\tau_\sigma = (e_1, \dots, e_m)$ on the edge set E of G from vertex ordering σ is the following. Order edges of G by assigning numbers from 1 to $|E|$. Assign the number k to an edge e (as yet unnumbered) which has lexicographically smallest pair $(left(e), right(e))$. In other words, we number first all edges that are incident to v_1 , in order of their right endpoints, then all unnumbered edges that are incident to v_2 and so on.

Lemma 14. *Let σ be a strong ordering of a strongly orderable graph G and τ_σ be an edge ordering of G obtained from σ . Then τ_σ is a simplicial-edge-without-vertex elimination ordering of G .*

Proof. Let $\sigma = (v_1, \dots, v_n)$ be a strong ordering of a graph G and $\tau_\sigma = (e_1, \dots, e_m)$. It is enough to show that the edge e_1 is simplicial in G and that σ is a strong ordering of the graph $G_1 = (V, E \setminus \{e_1\})$ too.

Let $e_1 = xy$ and assume that $x < y$ in σ . From our construction of τ_σ we obtain that x is the smallest vertex of G with respect to σ such that $N_G(x) \neq \emptyset$, and y is the smallest vertex of $N_G(x)$. Hence, for every $u \in N_G(x) \setminus \{y\}$ and $v \in N_G(y) \setminus \{x, u\}$ we have $y < u$ and $x < v$. Since σ is a strong ordering of G these vertices must be adjacent. So, the edge e_1 is simplicial in G .

Now assume that σ is not a strong ordering of G_1 . Then there exist four vertices a, b, c, d of G such that $ab, ac, bd \in E \setminus \{e_1\}$, $a < d$ and $b < c$ but $cd \notin E \setminus \{e_1\}$. Since σ is a strong ordering of G we conclude $cd = e_1 = xy$. But this is impossible, since by $a < d$ and $b < c$ neither c nor d can be the first vertex in σ which has a neighbor. \square

We call a graph *nontrivial* if it has at least one edge.

Corollary 15. *Every nontrivial induced subgraph H of a strongly orderable graph G has a simplicial edge.*

Proof. The smallest edge of H with respect to a simplicial-edge-without-vertex elimination ordering of G is simplicial in H . \square

Evidently, given a strong ordering of a strongly orderable graph G , a simplicial-edge-without-vertex elimination ordering of G can be produced in $O(|V| + |E|)$ time.

Theorem 16. *A graph G is strongly orderable if and only if it admits a simplicial-edge-without-vertex elimination ordering.*

Proof. In view of Lemma 14, we only need to prove that if G has a simplicial-edge-without-vertex elimination ordering τ then G is a strongly orderable graph. We proceed by induction on the number of edges of G . Let xy be the first edge of τ and

$G_1 = (V, E \setminus \{xy\})$. By induction, the graph G_1 is a strongly orderable graph. Then, by Lemma 5, G_1 has a quasi-simple elimination ordering. Consider a lexicographic quasi-simple elimination ordering σ of G_1 . From Lemma 8 we have that σ is a strong ordering of G_1 .

We claim that σ is a strong ordering of G as well.

Assume that this is not the case. Then there exist four vertices a, b, c, d of G such that $ab, ac, bd \in E$, $a < d$ and $b < c$ in σ , but $cd \notin E$. Since σ is a strong ordering of G_1 and xy is a simplicial edge in G we conclude $xy \in \{ac, bd\}$. Without loss of generality let $a = x$ and $c = y$. Then, we have $bd \in E$, $yd \notin E$ and $b < y$ in σ .

The edge xy is simplicial in G . Consequently, any vertex $z \in N_G(y) \setminus \{b\}$ is adjacent to b , i.e. vertex y is dominated by vertex b in G . Moreover, since d is adjacent to b and not to y we conclude that y is strictly dominated by b in G and, hence, in G_1 . Now we get a contradiction with Lemma 7 — vertex y is strictly dominated by vertex b in G_1 but $b < y$ in the lexicographic quasi-simple elimination ordering σ of G_1 .

Thus, σ is a strong ordering of G , i.e. G is a strongly orderable graph. \square

Let $G = (V, E)$ be a graph. $V' \subseteq V$ is an *independent set* in G if for all $u, v \in V'$, $uv \notin E$ holds. $V' \subseteq V$ is a *clique* in G if for all $u, v \in V'$ with $u \neq v$, $uv \in E$ holds.

A *sun* S_k of size k ($k \geq 3$) is a graph whose vertex set can be partitioned into two sets, $U = \{u_1, \dots, u_k\}$ and $W = \{w_1, \dots, w_k\}$, so that U induces a clique, W is an independent set and w_i is adjacent to u_j if and only if $i = j$ or $i = j + 1 \pmod{k}$. In [5,11] it was proven that a graph G is strongly chordal if and only if it is chordal and does not contain any sun S_k as an induced subgraph. An induced cycle C_k on k vertices with $k \geq 5$ is called a *hole*. An *antihole* is the complement of a hole. A graph is called *weakly triangulated* [16] if it has no hole or antihole as an induced subgraph.

Corollary 17. *Every strongly orderable graph is a weakly triangulated graph that does not contain suns as induced subgraphs.*

Proof. Straightforward verification shows that no edge of a hole is simplicial. By Corollary 15, holes cannot be induced subgraphs of a strongly orderable graph. We will see also that no antihole and no sun has a simplicial edge.

Consider an antihole $\overline{C}_k = (v_1, \dots, v_k, v_1)$ with the edge set E and an arbitrary edge $v_i v_j$, $i \neq j \pm 1 \pmod{k}$, of it. Vertices v_i and v_j divide the cycle C_k into two induced paths. Let $v_l v_{l+1}$ be a middle edge of a longest path, and assume that v_l is closer than v_{l+1} to v_j in that path. Since $k \geq 5$ in \overline{C}_k we have $v_l v_i, v_i v_j, v_j v_{l+1} \in E$ but $v_l v_{l+1} \notin E$, that is the edge $v_i v_j$ is not simplicial.

Now let S_k be a sun with clique $U = \{u_1, \dots, u_k\}$ and independent set $W = \{w_1, \dots, w_k\}$. Consider an edge $u_i u_j$ of S_k . By definition of suns, vertex w_i cannot be adjacent to both u_{j-1} and u_{j+1} . Assume without loss of generality that $u_{j+1} w_i \notin E$. Then from $w_i u_i \in E$, $u_{j+1} u_j \in E$ and $u_{j+1} w_i \notin E$ it follows that the edge $u_i u_j$ is not simplicial. Now consider the edges $u_i w_i$ and $u_i w_{i-1}$ of S_k . Since $w_i u_{i+1}, w_{i-1} u_{i-1} \in E$ and $w_i u_{i-1}, w_{i-1} u_{i+1} \notin E$ neither of these edges can be simplicial. \square

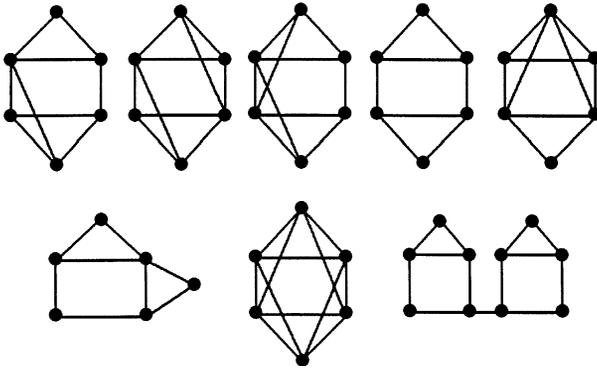


Fig. 1. Some forbidden subgraphs.

The converse of this corollary is not true. Fig. 1 presents some more minimal forbidden subgraphs for strongly orderable graphs.

3. Optimization problems on strongly orderable graphs

Here we show that four classical optimization problems can be solved in linear time on a strongly orderable graph if this graph is given together with a lexicographic quasi-simple elimination ordering.

Denote by

$$\alpha(G) = \max\{|V'| : V' \subseteq V \text{ and } V' \text{ independent in } G\},$$

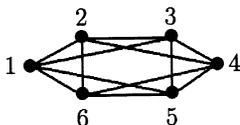
$$\omega(G) = \max\{|V'| : V' \subseteq V \text{ and } V' \text{ clique in } G\},$$

$$\gamma(G) = \min\{k : \text{there is a partition } V_1, \dots, V_k \text{ of } V \text{ such that } V_1, \dots, V_k \text{ are independent in } G\},$$

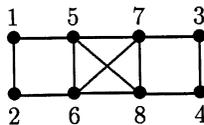
$$\kappa(G) = \min\{k : \text{there is a partition } V_1, \dots, V_k \text{ of } V \text{ such that } V_1, \dots, V_k \text{ are cliques in } G\},$$

respectively, the *independence number*, the *clique number*, the *chromatic number* and the *clique partition number* of a graph G . $\gamma(G)$ is called the chromatic number since a partition V_1, \dots, V_k of V into independent sets $V_i, i = 1, \dots, k$, is nothing else than a coloring of G such that no two adjacent vertices have the same color. The *maximum independent set problem* is to find an independent set S of G such that $|S| = \alpha(G)$. Analogously one can define the *maximum clique problem*, the *minimum coloring problem* and the *minimum clique partition problem*.

Obviously, for every graph G , $\alpha(G) \leq \kappa(G)$ and $\omega(G) \leq \gamma(G)$ hold. One of the most important graph classes is the class of perfect graphs. A graph G is *perfect* [2] if for all $V' \subseteq V$, $\gamma(G(V')) = \omega(G(V'))$ (or, equivalently [19], $\kappa(G(V')) = \alpha(G(V'))$) holds. It is proven in [16] that weakly triangulated graphs are perfect. Hence, by Corollary 17,



A perfectly orderable graph (with a perfect ordering) that is not strongly orderable.



A strongly orderable graph (with a strong ordering) that is not perfectly orderable.

Fig. 2.

strongly orderable graphs are perfect too. Below we will give an algorithmic proof of this fact.

A natural way to color the vertices of a graph G is to put them in order $\sigma=(v_1, \dots, v_n)$ and then assign colors in the following manner. Scan the vertices of G in the order v_1, \dots, v_n . Assign to each v_i the smallest color (positive integer) not used on any neighbor v_j of v_i with $j < i$. We call this *the greedy coloring algorithm*. An ordering σ of a graph is *perfect* if for each induced subgraph H of G , the greedy algorithm applied to H , gives an optimal coloring of H . An *obstruction* in a graph G with ordering σ is a set of four vertices $\{a, b, c, d\}$ with edges ab, bc, cd (and no other edges) and $a < b, d < c$ in σ .

In 1981, Chvátal [6] introduced the notions of perfect ordering and perfectly orderable graphs. A graph is *perfectly orderable* if it admits a perfect ordering. He proved that an ordering is perfect if and only if it contains no obstruction, and that all perfectly orderable graphs are perfect.

Although strongly orderable graphs are in general not perfectly orderable (see Fig. 2), we will see that the greedy coloring algorithm can be still used to compute a minimum coloring (and a maximum clique) of a strongly orderable graph. Moreover, we will show that every strong ordering of a graph G is a perfect ordering of the complement \bar{G} of G . Hence, due to $\kappa(G)=\gamma(\bar{G})$ (and $\alpha(G)=\omega(\bar{G})$), we can solve the minimum clique partition problem (and the maximum independent set problem) on a strongly orderable graph G by applying the greedy coloring algorithm to the graph \bar{G} .

3.1. Minimum coloring and maximum clique problems

Let $\sigma=(v_1, \dots, v_n)$ be a lexicographic quasi-simple elimination ordering of a strongly orderable graph G . The following algorithm computes simultaneously a minimum coloring and a maximum clique of a graph G . Note that this is backward greedy coloring. The meanings of l and p will be given in the proof of Theorem 19.

Algorithm MC & MC

Input: a strongly orderable graph $G=(V, E)$ with a lexicographic quasi-simple elimination ordering $\sigma=(v_1, \dots, v_n)$.

Output: a minimum coloring $c: V \rightarrow \{1, \dots, \gamma(G)\}$ and a maximum clique K of G .

```

set  $l = 1$ ,  $p = n$  and  $c(v_n) = 1$ 
for  $i = n - 1$  to 1 do
  begin
    find the minimum  $k \in \{1, 2, \dots\}$  such that
     $c(v_j) \neq k$  for all  $v_j \in N_G(v_i)$  with  $j > i$ 
    if  $k \leq l$  then set  $c(v_i) = k$ 
    else set  $p = i$ ,  $l = l + 1$  and  $c(v_i) = l$ 
  end
set  $K = N_G[v_p] \cap \{v_p, v_{p+1}, \dots, v_n\}$ .

```

Our correctness proof of this algorithm is based on the following structural lemma.

Lemma 18. *Let σ be a lexicographic quasi-simple elimination ordering of a graph G . Then any vertex v of G is simplicial in $G_{\geq v}$ or is dominated by a nonneighbor in this graph $G_{\geq v}$.*

Proof. Let x be the smallest vertex of $N_{G_{\geq v}}(v)$ with respect to σ . Since every lexicographic quasi-simple elimination ordering is a strong ordering, by Lemma 14, the edge vx is simplicial in $G_{\geq v}$. Hence, each neighbor u of x with $u > v$ is adjacent to all neighbors of v in $G_{\geq v}$. So, if there is in $G_{\geq v}$ a neighbor u of x which is not adjacent to v , then we are done, since v is dominated by u in $G_{\geq v}$. Otherwise, if all neighbors of x in $G_{\geq v}$ are neighbors of v too, then we obtain that x is dominated by v in the graph $G_{\geq v}$. Since $v < x$ in the lexicographic quasi-simple elimination ordering σ , by Lemma 7, x cannot be strictly dominated by v in $G_{\geq v}$. Thus the vertices v and x are twins in $G_{\geq v}$. So since edge xy is simplicial every two vertices of $N_{G_{\geq v}}(v)$ are adjacent. By definition, v is a simplicial vertex of $G_{\geq v}$. \square

Theorem 19. *Given a strongly orderable graph G and a lexicographic quasi-simple elimination ordering of G , Algorithm MC&MC correctly computes a minimum coloring and a maximum clique of G in linear time.*

Proof. In the algorithm we keep in l the number of currently used colors and in p the position in σ of the last vertex for which a new color was used. By Lemma 18, vertex v is simplicial in $G_{\geq v}$ or is dominated by a nonneighbor in this graph. If v is dominated by a nonneighbor u then the color $c(u)$ is not presented in the neighborhood of v in $G_{\geq v}$. Hence in this case, the smallest color k , that is not used on any neighbor of v in $G_{\geq v}$, will be smaller than or equal to l , and the vertex v will get an old color. We will need a new color only for vertex v which is simplicial in $G_{\geq v}$ and has number of neighbors in this graph equal to l . Thus, the total number of colors used by the algorithm is equal to the cardinality of clique $K = N_G[v_p] \cap \{v_p, v_{p+1}, \dots, v_n\}$. This means that the algorithm computes a minimum coloring and that K is a maximum clique of G .

It is not hard to implement the algorithm such that it runs in $O(|V| + |E|)$ time (see [6]). \square

We have proven that for every strongly orderable graph G , $\gamma(G) = \omega(G)$ holds. Since each induced subgraph of a strongly orderable graph is again strongly orderable, the class of strongly orderable graphs is perfect. Unfortunately, a lexicographic quasi-simple elimination ordering σ of a graph G is not necessarily a lexicographic quasi-simple elimination ordering of every induced subgraph. That is why the reverse ordering $\sigma^R = (v_n, \dots, v_1)$ to a lexicographic quasi-simple elimination ordering $\sigma = (v_1, \dots, v_n)$ is not necessarily a perfect ordering.

3.2. Minimum clique partition and maximum independent set problems

Let $\sigma = (v_1, \dots, v_n)$ be a strong ordering of a strongly orderable graph G . The next algorithm computes a minimum clique partition and a maximum independent set of a graph G in linear time. Its first part is the greedy coloring algorithm which works on the complement \bar{G} of G without constructing \bar{G} . So, we avoid a nonlinear computation of the complement of G . In the algorithm we keep in $t(k)$ the number of vertices of the graph colored with color k .

Algorithm MCP & MIS

Input: a strongly orderable graph $G = (V, E)$ with a strong ordering $\sigma = (v_1, \dots, v_n)$.

Output: a minimum clique partition $c : V \rightarrow \{1, \dots, \kappa(G)\}$ and a maximum independent set S of G .

{ *minimum clique partition* }

for $i = 1$ **to** n **do** set $t(i) = 0$

for $i = 1$ **to** n **do**

begin

find the minimum $k \in \{1, 2, \dots\}$ such that the number of neighbors v_j of v_i with $j < i$ and $c(v_j) = k$ equals $t(k)$;

set $c(v_i) = k$ and $t(k) = t(k) + 1$

end

set $\kappa(G) = \text{maximum } k \text{ such that } t(k) \neq 0$;

{ *maximum independent set* }

set $S = \emptyset$

for $i = \kappa(G)$ **to** 1 **do**

begin

set $v =$ smallest vertex with respect to σ from $K_i = \{x \in V : c(x) = i\}$;

set $u =$ smallest vertex with respect to σ from $K_i \setminus \{v\}$

if v is adjacent to some vertex of S **then** set $S = S \cup \{u\}$

else set $S = S \cup \{v\}$

end

Before we go to the correctness proof of the Algorithm MCP&MIS, we present a preliminary result of independent interest.

Lemma 20. *Each strong ordering of a graph G is a perfect ordering of the complement \bar{G} of G .*

Proof. Let σ be a strong ordering of G and assume that it is not a perfect ordering of \bar{G} . Then we can find in \bar{G} an obstruction, i.e. four vertices a, b, c, d which induce a path with edges ab, bc, cd of \bar{G} and fulfill $a < b$ and $d < c$ in σ . In G these vertices induce a path with edges ca, ad, db of G . Since σ was a strong ordering of G but we get that $a < b$, $d < c$ in σ and bc is not an edge of G , a contradiction arises. \square

Theorem 21. *Given a strongly orderable graph G and a strong ordering of it, Algorithm MCP&MIS correctly computes a minimum clique partition and a maximum independent set of G in linear time.*

Proof. In the first part of the algorithm we simply apply the greedy coloring algorithm to the complement \bar{G} of G without constructing this graph. Evidently, we need to assign to vertex v_i the smallest color not used on any nonneighbor v_j of v_i with $j < i$, i.e. the smallest color k such that all $t(k)$ vertices of G , colored with color k to this moment, are contained in $N_G(v_i)$. So, the correctness of this part follows from Lemma 20.

In the second part of the algorithm we construct a set S with $|S| = \kappa(G)$. We need to prove only that vertices of S are pairwise nonadjacent. We proceed by induction. Assume that set S constructed before step i ($i = \kappa(G), \kappa(G) - 1, \dots$) is independent and let v and u be defined as in the algorithm. Furthermore, assume that the vertex v has a neighbor b in S . We have $c(b) > c(v) = i$ and $v < b$ in σ . By the first part of the algorithm, this means that there must be a vertex c in clique $K_i = \{x \in V : c(x) = i\}$ with $c < b$ and which is not adjacent to b . (Otherwise, the vertex b must be colored with color i .) Then b is not adjacent to u too. Indeed, if $ub \in E$ then from $uv, vc \in E$, $u < c$, $v < b$ in strong ordering σ we will get $cb \in E$. Analogously, we can show that the vertex u is not adjacent to any vertex $d \in S$. If $ud \in E$ then from $uv, vb \in E$, $u \leq c < b$ and $v < d$ we obtain $db \in E$, which is impossible since S is independent.

Thus, $S \cup \{v\}$ or $S \cup \{u\}$ is an independent set. By induction, the final set S computed by the algorithm is independent.

Again, it is not hard to see that the algorithm works in $O(|V| + |E|)$ time. \square

Acknowledgements

The author is grateful to Andreas Brandstädt and Jeremy Spinrad for many useful discussions.

References

- [1] R.P. Anstee, M. Farber, Characterizations of totally balanced matrices, *J. Algorithms* 5 (1984) 215–230.
- [2] C. Berge, Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind, *Wiss. Zeitschr. Martin-Luther-Univ. Halle-Wittenberg*, 1961, p. 114.

- [3] A. Brandstädt, Special graph classes — a survey, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg, 1991.
- [4] G.J. Chang, Labeling algorithms for domination problems in sun-free chordal graphs, *Discrete Appl. Math.* 22 (1988/89) 21–34.
- [5] G.J. Chang, G.L. Nemhauser, The k -domination and k -stability problems on sun-free chordal graphs, *SIAM J. Algebra Discrete Methods* 5 (1984) 332–345.
- [6] V. Chvátal, Perfectly orderable graphs, *Ann. Discrete Math.* 21 (1984) 63–65.
- [7] E. Dahlhaus, M. Karpinski, On the computational complexity of matching and multidimensional matching in chordal and strongly chordal graphs, Technical Report, International Computer Science Institute, Number TR-94-043, August 1994.
- [8] E. Dahlhaus, Generalized strongly chordal graphs, Technical Report, University of Sydney, Number TR-93-458, 1993.
- [9] F.F. Dragan, On greedy matching ordering and greedy matchable graphs, in: R.H. Möhring (Ed.), 23rd International Workshop “Graph-Theoretic Concepts in Computer Science” (WG’97), Berlin, Germany, 1997, Lecture Notes in Computer Science, vol. 1335, Springer, Berlin, 1997, pp. 184–198.
- [10] U. Faigle, O. Goecke, R. Schrader, Church-Rosser decomposition in combinatorial structures, Preprint, Institut für Operation Research, Universität Bonn, 1986.
- [11] M. Farber, Characterizations of strongly chordal graphs, *Discrete Math.* 43 (1983) 173–189.
- [12] M. Farber, Domination, independent domination and duality in strongly chordal graphs, *Discrete Appl. Math.* 7 (1984) 115–130.
- [13] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph, *SIAM J. Comput.* 1 (1972) 180–187.
- [14] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
- [15] M.C. Golumbic, C.F. Goss, Perfect elimination and chordal bipartite graphs, *J. Graph Theory* 2 (1978) 155–163.
- [16] R. Hayward, Weakly triangulated graphs, *J. Combin. Theory B* 39 (1985) 200–209.
- [17] T. Kloks, D. Kratsch, Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph, *Inform. Process. Lett.* 55 (1995) 11–16.
- [18] B. Korte, L. Lovász, R. Schrader, *Greedoids*, Springer, Berlin 1991.
- [19] L. Lovász, A characterization of perfect graphs, *J. Combin. Theory B* 13 (1972) 95–98.
- [20] A. Lubiw, Doubly lexical orderings of matrices, *SIAM J. Comput.* 16 (1987) 854–879.
- [21] R. Paige, R.E. Tarjan, Three partition refinement algorithms, *SIAM J. Comput.* 16 (1987) 973–989.
- [22] D. Rose, R.E. Tarjan, G. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* 5 (1976) 266–283.
- [23] J.P. Spinrad, Doubly lexical ordering of dense 0–1 matrices, *Inform. Process. Lett.* 45 (1993) 229–235.
- [24] K. White, M. Farber, W. Pulleyblank, Steiner trees, connected domination and strongly chordal graphs, *Networks* 15 (1985) 109–124.