UNIVERSITE D'ORLEANS
Faculté des Sciences
LIFO

Laboratoire d'Informatique Fondamentale d'Orléans
4, rue Léonard de Vinci, BP 6759
F-45067 Orléans Cedex 2
FRANCE

Rapport de Recherche

www : http://www.univ-orleans.fr/SCIENCES/LIFO/

Recognizing i-triangulated graphs in $O(mn)$

F.Roussel, I. Rusu
Université d'Orléans, LIFO

Rapport N^o **98-10**

Recognizing i-triangulated graphs in $O(mn)$

F. Roussel, I. Rusu

Université d'Orléans, L.I.F.O., B.P. 6759,
45067 Orléans Cedex 2, France

Abstract

We use lexicographic breadth-first search and contractions of subgraphs to give a new (and better, from a complexity point of view) algorithm to recognize i-triangulated graphs.

Keywords: recognition algorithm, lexicographic breadth-first search, triangulated graph, i-triangulated graph

1 Introduction

Let the *chromatic number* of a graph G be the minimum number of colors necessary to color its vertices in such a way that any two adjacent vertices have different colors; and let the *clique number* of G be the maximum size of a clique of G. In [1], Berge defined a *perfect graph* as being a graph in which the equality between the chromatic number and the clique number holds for any of its induced subgraphs. One of the first classes of perfect graphs has been introduced by Gallai [4]: a graph is called *i-triangulated* if every odd cycle on five vertices or more contains at least two non-crossing chords.

At the date, several polynomial algorithms are known for i-triangulated graph recognition (see [2, 9, 10]), the best of which needs $O(mn^2)$ time. We give here an $O(n(m+n))$ algorithm based on LexBFS (see section 2) and successive contractions of subgraphs.

2 Preliminaries

All along the paper, $G = (V, E)$ will denote the *connected* graph we will be working on, in order to test its belonging to the class of i-triangulated graphs. For an arbitrary vertex $x \in V$, we will denote by $N(x)$ the *neighborhood* of x, i.e. the set of vertices (also called *neighbors* of x) which are adjacent to x in G.

For any other graph F ($F \neq G$), its vertex set will be denoted $V(F)$ and its edge set will be denoted $E(F)$. The *breadth first search* technique (abbreviated BFS) is a classical method to visit the vertices of a graph starting with a given vertex x and considering, at any step, all the neighbors of the current vertex (see [6] for more details). Rose, Tarjan and Leuker [8] proposed a stronger version of this algorithm (named *lexicographic breadth first search*), where the structure of the graph influences much more the order in which the vertices are considered. They used LexBFS to give a linear algorithm for recognizing *triangulated graphs*, i.e. graphs in which no chordless cycle on four vertices or more exists. All along the paper, and without making any other specification, we will use the description of this algorithm given by Golumbic in [5].

Assume that any vertex has a label, a set of numbers listed in decreasing order; the labels may be compared using lexicographic order (this is in fact dictionary order, using numbers instead of letters). Then the algorithm may be described as follows:

Algorithm LexBFS

Input: An arbitrary graph G.
Output: A one-to-one function $\sigma : \{1, 2, \ldots, n\} \rightarrow V$ (an order on V).

begin
  assign the label $\emptyset$ to each vertex;
  for $i := n$ downto 1 do
    pick an unnumbered vertex v with largest label (in lexicographic order);
    $\sigma(i) := v$; {comment: this assigns to v the number i}
    for each unnumbered vertex $w \in N(v)$ do
      add i to the label of w (at the end)
end

Tarjan, Rose and Leuker's algorithm for recognizing triangulated graphs consists in two main steps: firstly, Algorithm LexBFS is applied starting with an arbitrary vertex to obtain an order σ; then, an algorithm called Perfect tests whether, for each $i = 1, 2, \ldots, n$, the neighbors of $\sigma(i)$ in the set $\{\sigma(i+1), \ldots, \sigma(n)\}$ induce a clique (this is a necessary and sufficient condition for a graph to be triangulated). Both the algorithms LexBFS and Perfect need $O(m+n)$ time to be performed. Moreover, when the initial graph is not triangulated, Algorithm Perfect allows us to find three vertices v, u, w such that $\sigma^{-1}(v) < \sigma^{-1}(u) < \sigma^{-1}(w)$ and $vu \in E, vw \in E, uw \notin E$.

Now, we have from [5]:

Lemma 1 *Let v, u, w such that $\sigma^{-1}(v) < \sigma^{-1}(u) < \sigma^{-1}(w)$ and $vw \in E, uw \notin E$. Then there exists a vertex $z \in N(u) - N(v)$ such that $\sigma^{-1}(w) < \sigma^{-1}(z)$.*

Lemma 2 *Let v, u, w such that $\sigma^{-1}(v) < \sigma^{-1}(u) < \sigma^{-1}(w)$ and $vu \in E, vw \in E, uw \notin E$. Then a chordless cycle of G (of length at least four) can be found in $O(m)$ operations.*

Proof of Lemma 2. In [5], the proof that a chordless cycle may always be found is given, but no consideration on the complexity of the construction is given. We recall here the main steps of the proof, in order to show that it needs $O(m)$ operations.

Consider the vertex w' with the same properties as w, but with largest $\sigma^{-1}(w)$, and denote $x = v, x_1 = u, x_2 = w'$. Then we have a sequence $x, x_1, x_2 \ldots x_k$ (here $k = 2$) of vertices such that, for all $i, j > 0$,

(1) $xx_i \in E$ iff $i \leq 2$

(2) $x_i x_j \in E$ iff $|i - j| = 2$

(3) $\sigma^{-1}(x) < \sigma^{-1}(x_1) < \sigma^{-1}(x_2) < \ldots < \sigma^{-1}(x_k)$

(4) x_{j+1} is the largest vertex (with respect to σ) such that $x_{j-1}x_{j+1} \in E, x_{j-2}x_{j+1} \notin E$ (we assume that $x_0 = x, x_{-1} = x_1$).

As shown in [5], one of the two properties below is true:

a) either the sequence $x, x_1, x_2 \ldots x_k$ above may be extended to a sequence $x, x_1, x_2 \ldots x_k, x_{k+1}$ with the same properties (1)-(4),

b) or a chordless cycle of length four or more may be found.

This is done using Lemma 1 for x_{k-2}, x_{k-1}, x_k: the vertex x_{k+1} obtained in this way (which may be found in $O(|N(x_{k-1})|)$ time) either is nonadjacent to $x, x_1, x_2, \ldots, x_k$ (and then a) holds) or is adjacent to at least one of these vertices (and then b) holds). The adjacency test needs $O(|N((x_{k+1})|)$. So, in $O(|N(x_{k-1})| + |N(x_{k+1})|)$ we test which of a) and b) holds. If a) holds, then we continue to extend the sequence in the same way. As the graph is finite, an index k' exists such that $x_{k'+1}$ does not extend the sequence, so b) holds. The number of operations executed before finding it is in $O(\Sigma_{k=2}^{k'}(|N(x_{k-1})| + |N(x_{k+1})|)) \leq O(\Sigma_{x \in V}|N(x)| + \Sigma_{x \in V}|N(x)|) = O(2\Sigma_{x \in V}|N(x)|) = O(m)$.∎

In the rest of the paper, the $O(m)$ algorithm to find a chordless cycle with four vertices or more will be called Find-Cycle.

Now, once a cycle is found, we can determine a rough structure of a graph (and, in particular, of an i-triangulated graph) by putting together in sets the remaining vertices of the graph, according to their position with respect to the cycle. In fact, this can be done not only for a cycle, but for an arbitrary 2-connected bipartite subgraph of G, as follows:

Lemma 3 *Let $G = (V, E)$ be an i-triangulated graph and let B be a bipartite 2-connected subgraph of G. Then any vertex $x \in V - V(B)$ satisfies exactly one of the following affirmations:*
(i) x is adjacent to no vertex in B;
(ii) x is adjacent to all the vertices in B;
(iii) $\{x\} \cup V(B)$ induces a bipartite graph.

Proof of Lemma 3. By induction on the cardinality of V, we assume the lemma is true for any graph with less than $n = |V|$ vertices (obviously, it is true for graphs with four vertices).

It is easy to see that no vertex can simultaneously satisfy two or three of the affirmations above. If, by contradiction, there exists a vertex x which satisfies none of the affirmations, then $\{x\} \cup V(B)$ induces a connected subgraph which must contain at least one triangle xb_1b_2 (recall that G contains no odd chordless cycle) and at least one vertex $b_3 \in V(B)$ nonadjacent to x. As B is 2-connected, there exists a smallest (with respect to inclusion) cycle $C = [c_1 = b_1, c_2 = b_2, c_3, \ldots, c_i = b_3, \ldots, c_{2p}]$ containing b_1, b_2, b_3 (see [6] for details about 2-connected graphs).

If C is chordless, then it is easy to see that $C + x$ contains an odd cycle which has not two non-crossing chords. If C is not chordless, then a chord c_lc_h exists such that $2 \leq l < i$ and $i < h \leq 2p + 1$ (where $c_{2p+1} = c_1$). The induction hypothesis for x and the 2-connected bipartite subgraph given by $\{c_1, c_2, \ldots, c_l, c_h, \ldots, c_{2p}, c_1\}$ implies that $xc_l, xc_h \in E$, while the induction hypothesis for x and the 2-connected bipartite graph given by $\{c_l, c_{l+1}, \ldots, c_i, \ldots, c_{h-1}, c_h\}$ implies that $xc_i \in E$, a contradiction (since $c_i = b_3$ and $xb_3 \notin E$).■

After this presentation of the main results we need, we approach now, in Sections 3 and 4, the recognition and the coloring of an i-triangulated graph.

3 Recognition algorithm

For an induced subgraph H of G, denote by $T(H)$ the set of vertices in G which are adjacent to *any* vertex in H. If H is an edge xy, then we will write $T(x, y)$ instead of $T(H)$. A graph H is called *uniform* if for any two edges x_1y_1, x_2y_2 of it we have $T(x_1, y_1) = T(x_2, y_2)$ (and so they are both identical to $T(H)$). By Lemma 3, in an i-triangulated graph any induced bipartite 2-connected graph is uniform.

A graph is called *complete multi-partite* if its vertex set can be partitioned into stable sets $S_1, S_2, \ldots, S_l$ such that, for any $i \neq j$, each vertex in S_i is adjacent to each vertex in S_j. Such a stable set will be called *proper* if its cardinality is at least 2. A complete multi-partite graph may be then seen as consisting in several proper stable sets and one clique, such that any two vertices not belonging to the same proper stable set are adjacent. It is easy to see that a graph is complete multi-partite iff it does not contain three vertices a, b, c such that $ab \in E$, but $ca, cb \notin E$.

Now, we can describe the recognition algorithm as follows:

Algorithm Recognize

Input: A graph $G = (V, E)$.
Output: Either "TRUE" or "FALSE".

Step 1. Use algorithm Find-Cycle to find, if possible, a chordless cycle C in G. If C is not found, then goto *Step 7*; otherwise compute $T := T(C)$. If C is not uniform, then return "FALSE".

Step 2. Use breadth-first search to build a largest tree H, which is a partial subgraph of G, such that H contains at least one edge of C and is uniform. Denote by Γ the subgraph *induced* in G by $V(H)$ (then C is an induced subgraph of Γ and $T(\Gamma) = T(C) = T$).

Step 3. Test if Γ is a bipartite graph; if not, then return "FALSE".

Step 4. If Γ is a complete bipartite graph, then test if T induces a complete multi-partite graph. If not, then return "FALSE". Otherwise, for any proper stable set S of T, remove S from T and add S to $V(\Gamma)$. Let $\Gamma :=$ the graph induced by the new $V(\Gamma)$.

If Γ is not a complete bipartite graph, then test if T induces a clique. If not, then return "FALSE".

Step 5. For any x in $V(\Gamma)$, compute $L(x) :=$ the connected component of x in $V-(V(\Gamma)-\{x\})-T$. Test if, for any $x, y \in V(\Gamma)$, $L(x) \cap L(y) = \emptyset$. If not, then return "FALSE".

Step 6. Contract Γ into one vertex γ. Add a new vertex w adjacent exclusively to γ and all the vertices in T.
Denote the new graph by $G_{|\Gamma}$. Put $G := G_{|\Gamma}$ and goto Step 1.

Step 7. Return "TRUE".

The main purpose of Steps 1, 2, 3 above is to find, if possible, a chordless cycle on at least four vertices and to extend it to a maximal bipartite graph Γ around which the rest of the graph would be regularly organised. The input of Step 2 is the cycle C (found in Step 1), for which $T := T(C)$ has been computed in Step 1 (and, by Lemma 3 we know that for any edge xy in C we should have $T(x, y) = T(C)$, if G is i-triangulated). Then the breadth-first search technique (abbreviated BFS) is used to explore the graph and to build H. We start BFS with a vertex c in C (that is, we put $V(H) := \{c\}$) and, at any step of the algorithm BFS, we have a vertex u in H and one of its neighbors, s (at the very beginning, when $u = c$, we take s in C too, in order to insure that at least one edge of C exists in H). If s has already been considered, we do nothing. Otherwise, we add s and us to H iff $T(u, s) = T$. Obviously, all the vertices in C are in H, so C is an induced subgraph of Γ.

In Steps 4 and 5, the remaining vertices in G are partitioned into sets $L(x)$ ($x \in V(\Gamma)$), T and $R = V - V(\Gamma) - T - \bigcup_{x \in V(\Gamma)} L(x)$ (this last set will be never used), which must have the indicated properties if G is an i-triangulated graph. A particular treatment is necessary when Γ is a complete bipartite graph; in this case, Γ is augmented to a complete multi-partite graph.

Step 6 performs the reduction of the graph G into a smaller graph, with the same nature as G (i.e. i-triangulated or not), and indicates that the tests are to be continued on the smaller graph. Contracting Γ into one vertex γ consists in replacing the subgraph Γ with one new vertex γ which is adjacent to any neighbor of any vertex of Γ.

As soon as a graph with no cycle is found in Step 1, the algorithm is stopped and gives the answer "TRUE" (Step 7). The theorem below insures that the algorithm works correctly:

Theorem 1 *The initial graph G is i-triangulated iff the algorithm Recognize returns "TRUE".*

Proof of Theorem 1. We proceed by induction and we notice that for graphs with at most three vertices the algorithm gives the desired answer.

I. *The "only if" part.*

Assume now that the "only if" part holds for i-triangulated graphs with at most $n-1$ vertices, and let us prove it for G, which has n vertices. To this end, we assume that G is i-triangulated and show that in none of the steps 1-6 the answer "FALSE" is given.

Step 1. In the case where G is not only i-triangulated but also triangulated, then in Step 1 no cycle is found, so the algorithm goes to Step 7 and stops giving the answer "TRUE", which is the correct one.

Step 2. Now, if G contains at least one chordless cycle on four vertices or more, in Step 2 we obtain the connected induced subgraph Γ of G, which has H as a spanning tree and C as an induced subgraph. Since H is uniform (by construction) and contains at least one edge in C, for any edge xy in H we have $T(x,y) = T(C) = T$. We have nothing to prove here.

Step 3. Because of Step 2, we know that H is uniform. Let us firstly prove that Γ must be triangle-free. If this is not the case, let xyz be a triangle in Γ. If at least one edge of the triangle is an edge of H (say xy), we have $T(x,y) - T \supseteq \{z\}$, so $T(x,y) \neq T$, a contradiction. So no edge is in H. Since H is a tree, we can see that either the path joining x and y in H does not contain z or the path joining z to y in H does not contain x. Without loss of generality, we can suppose that the path P joining x and y in H does not contain z. Then P and the edge xy of Γ form a chordless cycle on four or more vertices, for which Lemma 3 can be applied (using z) to deduce that z is adjacent to any vertex of the cycle (and thus to the endpoints of at least one edge of H), so z should be in T, a contradiction.

Since G is i-triangulated and Γ is an induced subgraph of G, Γ cannot contain odd cycles on at least five vertices. Therefore Γ must be bipartite.

Step 4. Notice that, since Γ contains the chordless cycle C, Γ cannot be a clique. If Γ is a complete bipartite graph and T is not a complete multi-partite graph, then Γ contains two nonadjacent vertices a, b, while T contains three vertices c, d, e such that $de \in E$, but $cd, ce \notin E$; consequently, a, b, c, d, e induce an odd cycle which has no pair of non-crossing chords, a contradiction. Then, as indicated in the algorithm, any proper stable set S of T is removed from T and added to $V(\Gamma)$ such that the new graph Γ is complete multi-partite, while T is a (possibly empty) clique.

Finally, if Γ is bipartite but not complete bipartite, then T must be a clique, otherwise three vertices $a, b, c \in V(\Gamma)$ such that $bc \in E$, but $ab, ac \notin E$ and two nonadjacent vertices d, e in T induce again an odd cycle which has no pair of non-crossing chords, a contradiction.

Step 5. Notice that Γ is either a complete multi-partite graph with $k \geq 3$ proper stable sets $S_1, S_2, \ldots, S_k$, or a bipartite graph. In both cases we denote by Γ^0 (respectively T^0) the *old* graph Γ (respectively the *old* set T), i.e. the one we had before performing Step 4.

Now, if in Step 5 the graph Γ is a complete multi-partite graph (with $k \geq 3$), then we assume without loss of generality that Γ^0 is induced by $S_1 \cup S_2$, so that $T^0 = S_3 \cup \ldots \cup S_k \cup T$.

On the other hand, if in Step 5 the graph Γ is bipartite, then $\Gamma^0 = \Gamma$ (the two vertex sets of the bipartition are still denoted S_1, S_2) and $T^0 = T$.

In both cases T is a clique and H is a spanning tree of Γ^0. Moreover, every vertex of T is adjacent to all the vertices of Γ.

Claim 1 *In Γ, every edge xy satisfies $T(x,y) - V(\Gamma) = T$ (so that $T(\Gamma) = T$).*

Proof of Claim 1. If this is not true, we must have an edge xy of $E(\Gamma)$ and a vertex z in $V - (T \cup V(\Gamma))$ such that z is adjacent to both x and y.

By treating together the two cases with respect to Γ, the following situations can occur:

• xy is an edge of H; this is impossible, since then z should be in T^0 (i.e. either in $S_3 \cup ... \cup S_k \cup T$ if Γ is a complete multi-partite graph, or in T if Γ is a bipartite graph), a contradiction.

• xy is an edge of $\Gamma^0 - H$; then xy forms with the unique path joining x, y in H a chordless cycle for which Lemma 3 can be applied to deduce that z should have been in T^0, a contradiction. • xy is an edge of Γ, but not of Γ^0; this can happen only in the situation where Γ is complete multi-partite (with $k \geq 3$). Then assume without loss of generality that $x \in S_j, y \in S_{j+1}$ such that $j \geq 2$. By

Lemma 3 for z and the bipartite graph induced by S_j, S_{j+1}, the vertex z is adjacent to every vertex in $S_j \cup S_{j+1}$. In particular, it is adjacent to two nonadjacent vertices y, u in S_{j+1}. If z is adjacent to every vertex in Γ, then we are done (z should be in T, a contradiction). Otherwise, assume there exists a vertex $a \in S_q$ ($1 \leq q \leq k$) nonadjacent to z. For any index $r \neq q, j+1$, take $b \in S_r$ and notice that a, b induce with u, y, z a bad cycle on five vertices, except if z is adjacent to b. Then (since $r \neq q, j+1$ was arbitrarily chosen and $b \in S_r$ was arbitrarily chosen too) z is completely adjacent to any set S_r ($r \neq q$; for $r = j+1$, this was already proved). Moreover, z is adjacent to no vertex of S_q (otherwise, by Lemma 3 for S_q, S_j, it should be adjacent to *every* vertex in S_q, a contradiction).

Now, if $q \neq 1, 2$, then z is adjacent to every vertex in Γ^0, so it should be in $T^0 = S_3 \cup \ldots \cup S_k \cup T$, a contradiction. If $q = 1$ (and similarly for $q = 2$), then take $c_1, c_2 \in S_2$ and recall that the cycle given by z, c_1, a, c_2 must be uniform. So $T(c_1, z) = T(c_1, a) = T(c_2, z) = T(c_2, a) = T^0$, therefore z should have been put in H in Step 2, a contradiction.∎

Claim 2 *For any $x, y \in V(\Gamma), L(x) \cap L(y) = \emptyset$.*

Proof of Claim 2. By contradiction, assume there exist two vertices x, y in Γ such that $L(x) \cap L(y) \neq \emptyset$, and take x, y as close as possible (i.e. such that the shortest path P joining them in Γ has minimum length). Then there exists a chordless path P' joining x, y in the subgraph induced by $L(x) \cup L(y)$. The cycle induced by $V(P) \cup V(P')$ is a chordless cycle (by the choice of x, y) on three vertices or more. We show that none of these two cases is possible, thus obtaining a contradiction.

If the cycle has three vertices, we must have $xy \in E$. Let z be the internal vertex of P'. Since z is in $V - (V(\Gamma) \cup T)$ and z is adjacent to both x and y, we have a contradiction with Claim 1.

Thus, we can suppose that the cycle has at least four vertices. Let v be the neighbor of x on P', and u its neighbor on P. In both cases with respect to Γ (complete multi-partite or bipartite), we prove that $T(x, v) \neq T(x, u)$.

• if Γ is complete multi-partite (with $k \geq 3$ stable sets), then let $s \in V(\Gamma)$ be a common neighbor of x and u. If $T(x, v) = T(x, u)$, then v is adjacent to both x and s. So $L(x) \cap L(s) \neq \emptyset$ and then x, s are closer than x, y, a contradiction with the choice of x, y as close as possible.

• if Γ is bipartite, then by Claim 1 we have $T(x, u) = T(H)$. Thus, by the construction of H, if $T(x, v) = T(x, u)$ then v should have been put in $V(H)$. Then v should have been in Γ, a contradiction.

So $T(x, v) - T(x, u) \neq \emptyset$, or $T(x, u) - T(x, v) \neq \emptyset$. Assume that the former holds (the reasoning is identical if the later holds) and let $q \in T(x, v) - T(x, u)$. Lemma 3 for q and the cycle induced by $V(P) \cup V(P')$ implies that $qu \in E$, a contradiction.∎

Step 6. In this step, Γ is transformed into a smaller graph $G_{|\Gamma}$, for which, by induction, the "only if" part of Theorem 1 is true. So, if we show that $G_{|\Gamma}$ is i-triangulated, we are sure that the answer returned by the algorithm will be "TRUE". (Notice that $G_{|\Gamma}$ has a smaller number of vertices than G, since it is obtained from G by replacing a graph with at least four vertices, with only two vertices, γ and w).

Recall that a vertex v adjacent to γ in $G_{|\Gamma}$ is either w, or belongs to T, or else is in some $L(x)$ (with $x \in V(\Gamma)$) such that v, x are adjacent in G. Let us firstly prove that:

Claim 3 *If γ is contained in a chordless cycle B of $G_{|\Gamma}$ with at least four vertices, then there exists $x \in V(\Gamma)$ such that in G we have $V(B) - \{\gamma\} \subseteq L(x)$.*

Proof of Claim 3. Let $B = [v_1 = \gamma, v_2, \ldots, v_p]$ be the chordless cycle (which must be even), and notice that each of v_2, v_p belongs either to T or to some set $L(x)$ (where $x \in V(\Gamma)$). Obviously, v_2, v_p cannot be simultaneously in T since they are nonadjacent and T is a clique. Two non-symmetrical cases can occur:

Case 1. $v_2 \in T, v_p \in L(x)$ (where $x \in V(\Gamma)$). Then let a be a neighbor of x in Γ and notice that v_2 is adjacent to both x, a, while v_p is adjacent only to x (since $L(x) \cap L(a) = \emptyset$). Moreover, we can remark that for all $v_i \in \{v_3, ..., v_{p-1}\}$ we have $v_i a \notin E$ and $v_i x \notin E$ (otherwise, $v_i \gamma \in E(G_{|\Gamma})$). Now, the vertices $x, a, v_2, \ldots, v_p$ induce in G an odd cycle with no pair of non-crossing chords.

Case 2. $v_2 \in L(y), v_p \in L(x)$. Then no vertex among $v_2, \ldots, v_p$ is in T (it should be adjacent to γ), so this path is included both in $L(x)$ and in $L(y)$. We deduce that $L(x) \cap L(y) \neq \emptyset$, a contradiction, except if x, y are the same vertex. The proof of the claim is then finished.∎

We can prove now that $G_{|\Gamma}$ is an i-triangulated graph, thus proving the "only if" part of the theorem. If, by contradiction, $G_{|\Gamma}$ is not i-triangulated , then it contains some odd cycle on five vertices or more, with no pair of non-crossing chords. Consider a minimum (with respect to cardinality) such cycle $B = [v_1, v_2, \ldots, v_p]$. Necessarily, γ must be on B, otherwise B is also a bad cycle of G, a contradiction.

Case where B has two crossing chords.

It is easy to see that the crossing chords must be *short* chords, i.e. joining vertices at distance two along the cycle (otherwise a smaller bad cycle can be find). Then we can assume without loss of generality that the two chords are $v_1 v_3, v_2 v_4$.

If $\gamma \in \{v_1, v_4, v_5, \ldots, v_p\}$, then γ is situated on two chordless cycles on at least four vertices, and Claim 3 implies that there exists $x \in V(\Gamma)$ such that $V(B) - \{\gamma\} \subseteq L(x)$. Then $\{x\} \cup V(B) - \{\gamma\}$ is a bad cycle in G, a contradiction.

If $\gamma = v_2$ (and similarly if $\gamma = v_3$), then by Claim 3 for the cycle $[v_2, v_4 \ldots, v_p, v_1]$ we have that all the vertices $v_4 \ldots, v_p, v_1$ are in some $L(x)$ ($x \in V(\Gamma)$). The vertex v_3 cannot belong to $L(x)$ (a bad cycle would then exist in G), so $v_3 \in T \cup \{w\}$. Now, it is easy to see that we cannot have $v_3 = w$, since w is exclusively adjacent to γ and the vertices in T, while $v_4 \notin \{\gamma\} \cup T$. Moreover, if $v_3 \in T$, then in G we have $v_3 x \in E$ and the cycle $v_1, x, v_3 \ldots v_p$ is a bad cycle in G, a contradiction.

Case where B has only one chord.

Once more, the unique chord must be a short chord (assume it is $v_1 v_3$), otherwise the cycle is not minimum.

If $\gamma = v_1$ (and similarly if $\gamma = v_3$), then we have a chordless cycle which implies that $v_3, \ldots, v_p, v_1$ are in some $L(x)$ and $v_2 \in T \cup \{w\}$. As before, we show that v_2 cannot be identical to w, and it cannot belong to T neither.

If $\gamma = v_2$, then in G the vertices v_1, v_3 must be adjacent to different vertices x, respectively y of Γ (otherwise a bad cycle exists in G). Moreover, since $L(x) \cap L(y) = \emptyset$, at least one of v_1, v_3 must be in T. This is impossible since $v_3 \in T$ would imply $x v_3 \in E$ (thus a bad cycle would exist in G) and similarly if $v_1 \in T$.

If $\gamma = v_i$ ($4 \leq i \leq p$), then all the other vertices must be in the same set $L(x)$ and we have a bad cycle in G.

Case where B has no chord.

Then by Claim 3, the entire cycle is included in some $L(x)$ (where $x \in V(\Gamma)$) and we have a contradiction.

II. *The "if" part.*

We have to prove that if the algorithm answers "TRUE", then G is indeed i-triangulated. By induction, we assume this is true for graphs with smaller number of vertices. Now, since Algorithm Recognize returns "TRUE" and its execution just after computing $G_{|\Gamma}$ is, in fact, the execution of Recognize for this smaller graph $G_{|\Gamma}$, we have, by induction, that $G_{|\Gamma}$ is i-triangulated.

By contradiction, assume G is not i-triangulated and consider an odd cycle $D = [d_1, d_2, \ldots, d_q]$ of G, with at least five vertices, which has no pair of non-crossing chords and is minimum (with respect to cardinality) with this property. Since Γ is bipartite or complete multi-partite, $V(D)$ is not included in $V(\Gamma)$.

Claim 4 *Consider in G two paths P, P', each of them with endpoints x and y, where x, y are two (possible adjacent) distinct vertices in Γ. Assume that $V(P) \cup V(P') \not\subseteq V(\Gamma)$. Then at least one of the paths contains a vertex of T, while the other one either contains a vertex of T or is completely included in Γ.*

Proof of Claim 4. Since $V(P) \cup V(P') \not\subseteq V(\Gamma)$, at least one vertex of $V(P) \cup V(P')$ is in $T \cup \bigcup_{z \in V(\Gamma)} L(z)$. Assume no vertex in T exists in any of the paths. Then $V(P) \cup V(P')$ is included in $V(\Gamma \cup \bigcup_{z \in V(\Gamma)} L(z))$. Moreover, at least one path, say P, has a vertex u in some $L(z)$. As for all $x' \in V(\Gamma)$ we have $L(x') \cap L(z) = \emptyset$, then z must be on the subpath of P situated between x and u. Similarly, z must be on the subpath of P situated between y and u. We have a contradiction.

So at least one vertex in T exists in $V(P) \cup V(P')$ (say it is in $V(P)$). As before for P, we deduce that no vertex in some $L(z)$ can exist on P'. The claim is proved. ∎

Claim 5 *The cycle D must contain at least two vertices of Γ.*

Proof of Claim 5. Assume the contrary and notice that, necessarily, exactly one vertex of Γ exists on D, for otherwise D is a bad cycle of $G_{|\Gamma}$. Say that $d_1 \in V(\Gamma)$ and consider, in $G_{|\Gamma}$, the cycle F given by $V(D) - \{d_1\} \cup \{\gamma\}$. This cycle cannot be a copy of D, therefore at least one supplementary chord must exist (when compared to D). The only possibility is the existence of a vertex d_i ($i = 3, \ldots, q-1$) which is nonadjacent to d_1 in G but is adjacent to a vertex x in Γ (so that the chord γd_i exists in F).

Consider in G the paths $[d_1, d_2, \ldots, d_i, x]$ and $[d_1, d_q, d_{q-1}, \ldots, d_i, x]$, both of them with endpoints d_1, x in Γ. By Claim 4, we can assume that the path $[d_1, d_2, \ldots, d_i, x]$ contains a vertex d_j ($j = 2, 3 \ldots, i$) belonging to T. So $d_1 d_j \in E$. Furthermore, the other path is not completely included in Γ (since no vertex of D but d_1 belongs to Γ) so, again by Claim 4, it must contain some d_h ($h = q, q-1, \ldots, i$) which belongs to T. Then $d_1 d_h \in E$, $d_j d_h \in E$ (since T is a clique; notice that $j \neq i, h \neq i$ since $d_1 d_i \notin E$). We deduce that $d_1 d_j, d_1 d_h, d_j d_h$ are three non-crossing chords of D in G (a contradiction), except if some of these edges are identical to some edge of D. The only possibility is $j = 2$ and $h = q$ (so that the unique chord remains $d_j d_h$).

Now, let us show that the cycle given by $V(D) - \{d_1\} \cup \{w\}$ in $G_{|\Gamma}$ is isomorphic to D (and we have a bad cycle in $G_{|\Gamma}$, a contradiction). It is sufficient to show that $d_1 d_p \in E(D)$ if and only if $w d_p \in E(D)$ ($p = 2, 3, \ldots, q$). This is true for $p = 2$ and $p = q$ since w is adjacent to every vertex in T. Furthermore, if $d_1 d_p \in E$ ($2 < p < q$), then $d_p \in L(d_1)$ (since $d_p \notin V(\Gamma)$ by hypothesis and $d_p \in T$ would imply that either $d_j d_p$ or $d_h d_p$ is another chord of D, a contradiction). Now, the path joining d_i and d_p along D, not containing d_1, contains a vertex (namely d_p) of $L(d_1)$ and a vertex (namely d_i) of $L(x)$. As $L(d_1) \cap L(x) = \emptyset$, at least one vertex $t \in T$ must exist on this path. But then another chord (either $d_j t$ or $d_h t$) non-crossing $d_j d_h$ exists in D, a contradiction. Thus for all $p \in \{3, ..., q-1\}$, we have $d_1 d_p \notin E$. As no d_p, $p \in \{3, ..., q-1\}$, belongs to T (otherwise two non-crossing chords exist in D, $d_j d_h$ and either $d_p d_j$ or $d_p d_h$), then w is not adjacent to d_p. ∎

So, at least two vertices of Γ exist on D. Once more we have three cases for D:

Case where D has two crossing chords.

Again, the two crossing chords must be short, so assume they are $d_1 d_3, d_2 d_4$.

If D contains two vertices x, y of Γ which are not consecutive on D, then these are joined along D by two paths. By Claim 4 (using the first possibility for the second path, otherwise D has non-crossing chords), each of these paths contains at least one vertex of T (denoted t_1, respectively t_2). Since T is a clique, t_1, t_2 must be adjacent. Then the edge $t_1 t_2$ is either $d_1 d_3$ or $d_2 d_4$ (say it is $d_1 d_3$) and then we have $x = d_2, y = d_i$ ($i \geq 4$), so we deduce $t_2 y \notin E$ or $t_1 y \notin E$, a contradiction.

If D contains two vertices x, y of Γ which are consecutive on D, then the path of length at least 2 joining them along D must contain a vertex t in T (by Claim 4) Then t is adjacent to both x, y and at least one of these edges must be an edge along D (otherwise D contains two non-crossing chords). We deduce that the edge xy is one of the edges $d_1 d_2, d_2 d_3, d_3 d_4$. For $d_1 d_2$ (and similarly for $d_3 d_4$), we must have $t = d_3$ and we can apply Claim 4 for the paths $[d_2, d_1]$ and $[d_2, d_4, d_5, \ldots, d_p, d_1]$ to find some $d_j \in T$ $(4 \leq j \leq q)$, a contradiction. We deduce that the edge xy must be the edge d_2d_3, and that d_1, d_4 should be in T. This gives a contradiction, since T is a clique.

Case where D has a unique chord.

Assume the unique chord is d_1d_3. As before, at least two vertices x, y of Γ must exist on D and, if they are non-consecutive on D, each of the two paths joining x, y along D must contain a vertex in T. As T is a clique, the two vertices in T must be d_1 and d_3, therefore $x = d_2$ and y cannot be adjacent to both d_1, d_3, unless D has only four vertices, a contradiction. So x, y are consecutive on D; then they must be adjacent to a vertex $t \in T$ which must exist on D. So they must be $x = d_1, y = d_2$ and $t = d_3$. No other vertex in $T \cup V(\Gamma)$ exists on D (we *necessarily* obtained that a pair of vertices in Γ must be the one we found), so a copy of D can be realized in $G_{|\Gamma}$ by taking γ instead of d_1 and w instead of d_2. Therefore $G_{|\Gamma}$ is not i-triangulated , a contradiction.

Case where D has no chord.

Then two vertices in Γ situated on D imply, if they are nonadjacent, the existence of two vertices in T which are nonadjacent (a contradiction) and, if they are adjacent, the existence of a vertex $t \in T$ which should be adjacent to both x, y, a contradiction.∎

4 Complexity aspects

The number of iterations of Steps 1 to 6 in the algorithm is of $O(n)$, since the number of vertices of the graph reduces at each iteration. Steps 1 to 6 may be done in $O(m+n)$ time (where m is the number of edges in G), if the graph is represented both using adjacency lists and the adjacency matrix, as shown below.

As explained in Section 2, the algorithm Find-Cycle needs no more than $O(m+n)$ operations. To compute $T(C)$ and test the uniformity of C, it is sufficient to compute $T := T(a, b)$ for a fixed edge ab of C, and then to test whether each edge xy has $T(x, y) = T$. This only needs to visit all the neighbors of any vertex, so the complexity is in $O(m)$.

In Step 2, the traversal of the graph needs $O(m+n)$ operations, since this is the complexity of the algorithm BFS. The test whether each edge uw (where u is already in H and w is a neighbor of u not yet considered) has or not $T(u, w) = T$ needs only to visit the neighbors of w (using the adjacency list) and to verify that $|T|$ of them are in T, while the ones which are not in T are nonadjacent to u (the adjacency matrix is used in this case). The complexity is still in $O(m+n)$.

The test in Step 3 needs a classical $O(m+n)$ algorithm.

In Step 4, testing if Γ is a complete multi-partite graph is easily done using the $O(m+n)$ algorithm of Corneil, Perl and Stewart [3] for building the cotree representation of a graph, if it has no induced P_4 (or indicate a P_4, in the contrary case). As the cotree representation of a complete multi-partite graph is very simple (only two levels in the cotree), it is very simple to compare it with the cotree of Γ.

In Step 5, the only difficulty is to verify whether the sets $L(x)$ are disjoint. To do this, Algorithm BFS (using only vertices not in $V(\Gamma) \cup T$) must be performed starting with any vertex in Γ. If one of these traversals encounters another vertex of Γ, then the condition is not verified. Otherwise, it is verified.

In Step 6, all the operations are easy.

Therefore, the complexity of the whole algorithm is in $O(n(m+n))$.

References

[1] C. Berge, Les problèmes de coloration en théorie des graphes, *Publ. Inst. Stat. Univ. Paris* 9 (1960), 123-160.

[2] M. Burlet, J. Fonlupt, Polynomial algorithm to recognize a Meyniel graph, *Annals of Discrete Math.* 21 (1984), 225-252.

[3] D. G. Corneil, Y. Perl, L. K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* vol. 14, no. 4 (1985), 926–934.

[4] T. Gallai, Graphes mit triangulierbaren ungeraden Vielecken, *Magyar Tud. Akad. Mat. Kutato Int. Közl.* 7 (1962) 3-36.

[5] M. C. Golumbic, "Algorithmic graph theory and perfect graphs", Academic Press (1980).

[6] U. Manber, "Introduction to algorithms. A creative approach", Addison-Wesley (1989).

[7] R. M. McConnel, J. P. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, *Proceedings of the fifth annual ACM-SIAM Symposium on Discrete Algorithms, SODA* (1994), 536-545.

[8] D. J. Rose, R. E. Tarjan, G. S. Leuker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* 5 (1976), 266-283.

[9] F. Roussel, I. Rusu, Holes and dominoes in Meyniel graphs, submitted.

[10] S. H. Whitesides, A method for solving certain graph recognition and optimization problems, with applications to perfect graphs, *Annals of Discrete Math.* 21 (1984) 281-297.

Publications récentes du LIFO
LIFO's recent publications

97-7 E. MELIN, B. RAFFIN, X. REBEUF, B. VIROT
A Unifying Approach of Data and Task Parallelism Based on Structural Clock Communication Paradigm (extended version)

97-8 S. ANANTHARAMAN, Editor
Proceedings of UNIF'97, International Workshop on Unification

97-9 F. BENTAYEB, D. LAURENT
Inversion de l'algrèbre relationnelle et mises à jour

97-10 I. PARFENOFF
An efficient parallel algorithm for maximum matching in P_4-tidy graphs

97-11 F. ROUSSEL, I. RUSU
An O(m2+mn) algorithm to recognize Meyniel graphs

97-12 M. BAHMA, G. HAINS
Prevision des performances de la θ-jointure et de la multi-jointure pour des machines SN

97-13 F. LOULERGUE, G. HAINS
Design of an event-oriented functionnal parallel language : the finite case

97-14 S. LIMET, F. SAUBION
A proof method for logic programming

97-15 E. MELIN, B. RAFFIN, X. REBEUF, B. VIROT
An Automatic Distribution of Sequential Code Fitting Irregular Data Accesses

Pour récupérer les rapports de recherche du LIFO / To get LIFO's research reports :

- Par WWW : http://www.univ-orleans.fr/SCIENCES/LIFO/Recherche/Publis/RR/index.html
- Par FTP : ftp://ftp-lifo.univ-orleans.fr/pub/RR
- Par courrier papier adressé à / By surface mail adressed to :
  Arnaud Lallouet, LIFO, Université d'Orléans
  4, rue Léonard de Vinci, BP 6759
  F-45067 Orléans cedex 2, FRANCE
  en mentionnant votre adresse.
- Par courrier électronique à / By e-mail to : Arnaud.Lallouet@lifo.univ-orleans.fr