Discrete Mathematics and Theoretical Computer Science **4**, 2001, 173–178

Linear time recognition of P_4-indifference graphs

Michel Habib[1] and Christophe Paul[2] and Laurent Viennot[3]

[1]*LIRMM, Université Montpellier II, 161 rue Ada, 34392 Montpellier Cedex, France*
Email: `habib@lirmm.fr`

[2]*LaBRI, Université Bordeaux I, 351 cours de la Libération, 33405 Talence Cedex, France.*
Email: `paul@labri.u-bordeaux.fr`

[3]*INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France.*
Email: `Laurent.Viennot@inria.fr`

received Oct 22, 1999, *revised October, 2000*, *accepted Mar 16, 2001*.

A graph is a P_4-indifference graph if it admits an ordering $<$ on its vertices such that every chordless path with vertices a, b, c, d and edges ab, bc, cd has $a < b < c < d$ or $d < c < b < a$. We present a linear time recognition for these graphs.

Keywords: P_4-indifference, algorithm, recognition.

1 Introduction

A P_4 is a chordless path of four vertices. A graph is P_4-indifference if it admits an ordering $<$ on its vertex set such that every P_4 $abcd$ has $a < b < c < d$ or $d < c < b < a$. Such an ordering is called a P_4-indifference ordering. The P_4-indifference graphs were introduced in [Chv84] as a particular class of *perfectly orderable graphs*. A graph is perfectly orderable if there exists an ordering on its vertex set for which the greedy colouring algorithm produces an optimal colouring.

The first recognition algorithm for P_4-indifference graphs is due to Hoàng and Reed and has the complexity of $O(n^6)$ [HR89]. They compute the equivalence classes of some relation on the P_4's of the graph. They then check that these classes do not contain a certain subgraph with 6 vertices. Later, Raschle and Simon, studying more carefully the P_4's relations, proposed an $O(n^2m)$ recognition algorithm [RS97].

Recently, Hoàng, Maffray and Noy gave a characterization by forbidden induced subgraphs [HMN99] and raised the question of the existence of a linear time recognition algorithm. We answer their question in the affirmative way using some of their theorems. Moreover our algorithm computes an adequate ordering of the vertices when it concludes that the input graph is P_4-indifference.

1365–8050 © 2001 Maison de l'Informatique et des Mathématiques Discrètes (MIMD), Paris, France

2 Theoretical basis

We use the following theorems from [HMN99]:

Theorem 1 *[HMN99] Any P_4-indifference graph fulfills the following properties:*

1. *If it contains a C_4 (a chordless cycle of length 4) then it contains an homogeneous set.*
2. *If it contains no C_4 then it is an interval graph.*

These two properties inspire the following recognition algorithm: Compute the modular decomposition tree of the input. For each quotient graph of any node of the tree verify that it is an interval graph. Compute an interval representation of it and use it to test whether it is a P_4-indifference graph and to compute a good ordering of the vertices if there exists one. The existence of linear time algorithms for modular decomposition and interval graph recognition [MS94, MS99, HM91] make this scheme possible for a linear time recognition algorithm.

To justify such an algorithm, we first need some additional theoretical results linking P_4-indifference graphs and modular decomposition.

Theorem 2 *[HMN99] The composition of two graphs is P_4-indifference iff they are both P_4-indifference graphs.*

To make the paper self-contained and because the algorithm is strongly based on theorem 2, we present its proof. This result first appeared in [HMN99].

Proof: Let G be the composition of two graphs G_1 and G_2 where G is obtained from G_1 by replacing a vertex u_2 by G_2 by linking all the vertices of G_2 to all the neighbors of u_2.

First suppose G is P_4-indifference. We prove that G_1 and G_2 are also P_4-indifference. Let $z_1 < \cdots < z_n$ be a P_4-indifference ordering of the vertices of G. The induced order of the vertices of G_2 will obviously fulfill the same condition. G_2 is thus clearly a P_4-indifference graph. Consider the ordering of the vertices of G_1 obtained from $z_1 < \cdots < z_n$ by erasing all the vertices of G_2 but one that is replaced by u_2. It is clearly a P_4-indifference ordering.

Second suppose G_1 and G_2 are P_4-indifference graphs. We show that G is also a P_4-indifference graph. Let $y_1 < \cdots < y_p$ be a P_4-indifference ordering of the vertices of G_2 and $x_1 < \cdots < x_m$ be a P_4-indifference ordering of the vertices of G_1 where $u_2 = x_k$. Then $x_1 < \cdots < x_{k-1} < y_1 < \cdots < y_p < x_{k+1} < \cdots < x_m$ is clearly a P_4-indifference ordering of the vertices of G. $\square$

An immediate corollary of Theorem 2 is the following:

Corollary 1 *A graph is a P_4-indifference graph iff all the quotient graphs of its modular decomposition tree are P_4-indifference graphs.*

Notice that the second part of the proof of theorem 2 also gives a simple way to compute a P_4-indifference ordering of the composition of some P_4-indifference graphs from their P_4-indifference orderings.

Corollary 2 *A P_4-indifference ordering can be computed in linear time from the P_4-indifference orderings of the quotient graphs.*

Proof: Assume you are given a P_4-indifference ordering for each quotient graph. Then visit the modular tree decomposition in a post-order fashion, and for each node N do the following:

- if N is a leaf, then it corresponds to a single vertex and the ordering is trivial.
- if N is an internal node, then for each son S_i of N you have a P_4-indifference ordering σ_i. Let σ_H the P_4-indifference ordering of H, the quotient graph associated to N. Each S_i corresponds to a vertex x_i of H. Then σ a P_4-indifference ordering of the graph whose tree decomposition is rooted at N, can be obtained by substituting each σ_i to x_i in σ_H.

The linearity of the above algorithm comes from the linearity of the sum of the sizes of the quotient graphs. $\square$

So now, to complete our recognition algorithm of P_4-indifference graphs, we just need to compute P_4-indifference ordering and to recognize prime P_4-indifference graphs.

3 Recognition of prime interval P_4-indifference graphs

Let G be a prime interval graph. Let $I_1, \ldots, I_n$ be a minimal interval representation of it where each I_k is an integer interval of $[1,N]\cap$ (with N minimal). If u is a vertex, we denote by I_u its associated interval. Recall that by definition, two vertices u and v of G are linked iff I_u intersects I_v. We say that two intervals *overlap* when they intersect without one being included in the other. When two intervals do not intersect, we say that the one with greater (resp. smaller) elements is *greater* (resp. *smaller*) than the other.

We are now going to show how a minimal interval representation is close to a P_4-indifference ordering. The following theorem can easily be deduced from the proofs in [HMN99]. It links minimal interval representations and P_4-indifference orderings.

Theorem 3 *[HMN99] Consider a prime interval graph G and a minimal interval representation of it. Let $\prec$ be the relation satisfying $x \prec y$ for any vertices x,y satisfying one of the three following properties:*

1. *I_x and I_y overlap and the left bound of I_x is smaller than the left bound of I_y.*
2. *I_x is included in I_y and there exists a P_4 x,y,z,t such that I_y and I_z overlap and the left bound of I_y is smaller than the left bound of I_z.*
3. *I_y is included in I_x and there exists a P_4 y,x,z,t such that I_z and I_x overlap and the left bound of I_z is smaller than the left bound of I_x.*

G is a P_4-indifference graph iff $\prec$ is acyclic. Moreover any extension of $\prec$ (i.e. for each x,y $x \prec y$ implies $x < y$ or for each x,y $x \prec y$ implies $x > y$) is a P_4-indifference ordering.

Notice that the previous remarks imply that x and y are vertices of some P_4 in each of the three situations of Theorem 3. Moreover any two consecutive vertices of some P_4 are in relation by $\prec$. And all above, any P_4 a,b,c,d either verifies $a \prec b \prec c \prec d$ or $d \prec c \prec b \prec a$. See [HMN99] for the details of the proofs.

In order to use theorem 3 to find a P_4-indifference ordering, in cases *2.* and *3.*, we have to find some P_4 containing x and y. Such a work is the bootleneck of complexity issue. The next lemma is the new tool that makes possible the design of a linear time recognition algorithm.

Lemma 1 *Let b and c be two vertices. The corresponding intervals I_b and I_c in a minimal interval representation overlap iff b and c are the middle vertices of some P_4.*

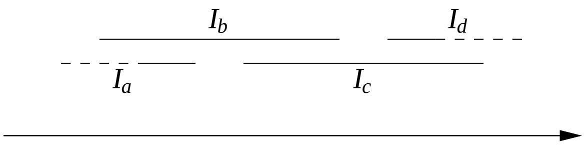

Fig. 1: Any P_4 a', b', c', d' is embedded in the interval representation as shown. Either $a' = a, b' = b, c' = c, d' = d$ or $a' = d, b' = c, c' = b, d' = a$

Proof: Consider a P_4 a, b, c, d with edges ab, bc, cd. We claim that the intervals I_b and I_c associated to b and c must overlap. They intersect since the two vertices are linked. Since I_a intersects I_b but not I_c, I_b cannot be included in I_c. For a similar reason with I_d, I_c cannot be included in I_b. Thus any P_4 is of the form illustrated by Figure 3 in the interval representation.

Conversely, when two intervals I_b and I_c overlap then b and c are the middle vertices of at least one P_4. This is due to the fact that the interval representation has been chosen minimal. Suppose for example that some elements of I_b are smaller than those of I_c (the other case is symmetrical). Let i be the greatest integer of I_b that is not in I_c. There must exist an interval I_a containing i without intersecting I_c otherwise i could be removed yielding a more compact representation (contradicting the minimality of the present one). The same argument allows to conclude that there must exist some interval I_d intersecting I_c but not I_b. a, b, c, d is then a P_4. □

Therefore theorem 3 can be rewriten as follows (in particular cases *2.* and *3.*):

Corollary 3 *Consider a prime interval graph G and a minimal interval representation of it. Let $\prec$ be the relation satisfying $x \prec y$ for any vertices x, y satisfying one of the three following properties (these situations are illustrated by figure 3):*

1. *I_x and I_y overlap and the left bound of I_x is smaller than the left bound of I_y.*
2. *I_x is included in I_y and there exists some interval I_z greater than I_x overlapping I_y.*
3. *I_y is included in I_x and there exists some interval I_z smaller than I_y overlapping I_x.*

G is a P_4-indifference graph iff $\prec$ is acyclic. Moreover any extension of $\prec$ (i.e. for each x, y $x \prec y$ implies $x < y$ or for each x, y $x \prec y$ implies $x > y$) is a P_4-indifference ordering.

Corollary 4 *The recognition of prime P_4-indifference graphs can be done in linear time.*

Proof: Let us briefly describe the recognition algorithm :

- Test if the input graph is an interval graph and if so compute a minimal interval representation. It can be done in linear time by any linear interval graphs recognition algorithm (see [BL76, KM89, HM91, HMPV97, COS98] for example).
- The $\prec$ relation can easily be computed in linear time by storing for each interval the two intervals overlapping it which have the rightmost left bound and the leftmost right bound when they exist. During this computation, you can find all the relations corresponding to overlapping intervals (case (1) of figure 3). Then the relations corresponding to case (2) and (3) of figure 3 can be computed.

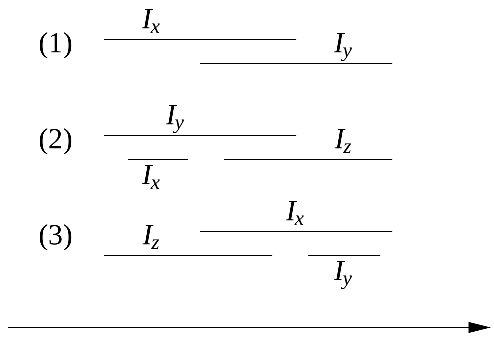

Fig. 2: The three situations of Theorem 3 implying $x < y$ for a P_4-indifference ordering.

- Using a Depth First Search, the acyclicity of $\prec$ can be tested and a linear extension of it can be computed (it there exists one). It can be done in linear time.

$\square$

4 Conclusions

This paper shows how a linear time algorithm for the P_4-indifference graphs recognition can be designed. This algorithm strongly relies on modular decomposition as a preprocessing. But linear time modular decomposition algorithms are still complicated to program. So the natural question is: can this preprocessing step be avoided ?

So it has been shown that prime P_4-indifference graphs are interval graphs. It is well known that Lexicographic Breadth First Search (Lex-BFS) [RTL76] plays an important role on interval graphs [HM91, COS98, HMPV97]. The order Lex-BFS visits the vertices of the input graph can be seen as the output of Lex-BFS: a Lex-BFS ordering. For example in [COS98], 4 sweeps of particular Lex-BFS are used to compute a characteristic ordering of interval graphs (the i-th sweep starts on the last visited vertex of the previous sweep). One can wonder if Lex-BFS can be used to compute a P_4-indifference ordering. As illustrated by the graph of figure 4, the answer is no.

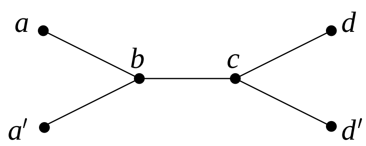

Fig. 3: A graph such that no Lex-BFS ordering is a P_4-indifference ordering

On the above graph, no Lex-BFS ordering is a P_4-indifference ordering. So there is no hope for some special Lex-BFS as those defined in [COS98]. We can remark that this graph contains modules ($\{a,a'\}$ and $\{d,d'\}$). It seems that restricted to prime P_4-indifference graphs, 2 sweeps of Lex-BFS computes a P_4-indifference ordering (it can be a simplification of the presented algorithm). But up to now, we do not know how to avoid the modular decomposition.

The presented algorithm relies on some properties of prime graphs and also on some P_4 relations. Can these structural results be adapted to other classes of perfectly orderable graphs like for example P_4-comparability graphs, P_4-simplicial graphs ... in order to design efficient recognition algorithms ?

We thank the referees for their fruitful remarks on the presentation of the result.

References

[BL76] K.S. Booth and G.S. Leuker. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithm. *J. Comput. Syst. Sci.*, 13:335–379, 1976.

[Chv84] V. Chvàtal. Perfectly ordered graphs. *Annals of Discrete Mathematics*, 21:63–66, 1984.

[COS98] D.G. Corneil, S. Olariu, and L. Stewart. The ultimate interval graph recognition algorithm ? In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, pages 175–180, 1998.

[HM91] W.L. Hsu and T.H. Ma. Substitution decomposition on chordal graphs and applications. In *Proceedings of the 2nd ACM-SIGSAM Internationnal Symposium on Symbolic and Algebraic Computation*, number 557 in LNCS, 1991.

[HMN99] C.T. Hoàng, F. Maffray, and M. Noy. A characterization of p4-indifference graphs. *Journal of Graph Theory*, 31(3):155–162, 1999.

[HMPV97] M. Habib, R.M. McConnell, C. Paul, and L. Viennot. Lex-BFS a partition refining technique, application to transitive orientation and consecutive 1's testing. *Theoretical Computer Science*, 1997. To appear.

[HR89] C.T. Hoàng and B.A. Reed. Some classes of perfectly orderable graphs. *Journal of Graph Theory*, 13(4):445–463, 1989.

[KM89] N. Korte and R. Möhring. An incremental linear-time algorithm for recognizing interval graphs. *SIAM Journal of Computing*, 18:68–81, 1989.

[MS94] R.M. McConnell and J.P. Spinrad. Linear-time modular decomposition and efficient transitive orientation of undirected graphs. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 536–545, 1994.

[MS99] R.M. McConnell and J.P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201:189–241, 1999.

[RS97] T. Raschle and K. Simon. On the p_4-components of graphs. Technical Report 261, Institute of Theoretical Computer Science, ETH Zürick, 1997.

[RTL76] D. J. Rose, R. E. Tarjan, and G. S. Leuker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal of Computing*, 5(2):266–283, 1976.