Short Models for Unit Interval Graphs

Min Chih Lin [a,1],
Francisco J. Soulignac [a,2] and Jayme L. Szwarcfiter [b,3]

[a] *Departamento de Computación, FCEN, Universidad de Buenos Aires, Buenos Aires, Argentina.*
`{oscarlin,fsoulign}@dc.uba.ar`

[b] *Universidade Federal do Rio de Janeiro, Inst. de Matemática, NCE and COPPE, Caixa Postal 2324, 20001-970 Rio de Janeiro, RJ, Brasil.*
`jayme@nce.ufrj.br`

Abstract

We present one more proof of the fact that the class of proper interval graphs is precisely the class of unit interval graphs. The proof leads to a new and efficient $O(n)$ time and space algorithm that transforms a proper interval model of the graph into a unit model, where all the extremes are integers in the range 0 to $O(n^2)$, solving a problem posed by Gardi (Discrete Math., 307 (22), 2906–2908, 2007).

Keywords: algorithms and data structures, graph theory, efficient representation, proper interval graphs, unit interval graphs

[1] Partially supported by UBACyT Grants X456 and X143, and PICT ANPCyT Grant 1562.

[2] Partially supported by UBACyT Grant X456, PICT ANPCyT Grant 1562 and by a grant of the YPF Foundation.

[3] Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, FAPERJ, Brasil.

1 Introduction

For a graph, denote by n and m the number of vertices and edges, respectively. A graph is an *interval graph* if its vertices can be put in a one-to-one mapping with a set of intervals of the real line in such a way that two vertices of the graph are adjacent if and only if their corresponding intervals have nonempty intersection. The set of intervals is called an *interval model* or *interval representation* of the graph. If in some interval model of the graph no interval is properly contained in some other interval then the graph is a *proper interval graph*, while the model is a *proper interval model*. If also all the intervals are of the same size, the graph is a *unit interval graph* and the model is a *unit interval model*.

By definition, every unit interval model is a proper interval model, so every unit interval graph is also a proper interval graph. In 1969, Roberts [7] proved that the converse is also true; every proper interval graph admits a unit interval model. There are several $O(n + m)$ time algorithms that recognize if a given graph is a proper interval graph, which output a proper interval model in the affirmative case (e.g. [2]). So, the recognition problem for unit interval graphs is well solved. However, for the representation problem, which is to find a unit interval model of a proper interval graph, there are not so many linear time algorithms. There are two possible inputs for the representation problem, namely the graph represented by its adjacency lists or a proper interval model represented by its family of intervals. We now give a brief description of the known representation algorithms for unit interval graphs.

In 1998, Corneil et al. [3] developed an $O(n + m)$ time algorithm for the recognition and representation problems, when the input is the adjacency lists of the graph. Their algorithm is divided into two main phases. In the first phase, the algorithm produces an ordering of the vertices of the graph that reflects the order of the intervals in a proper interval model. In the second phase, the algorithm tranforms this ordering into a unit interval model of the graph, using a postorder traversal of a breath-first search tree. One of the main properties about the output model is that every extreme is represented by an integer in the range $[0, n^2]$, which is asymptotically the best possible.

In 1999, Bogart and West [1] gave a new simple and constructive proof of the "Proper = Unit" Theorem. This proof yields — as it is — an $O(n^2)$ operations algorithm to transform a proper interval model into a unit interval model. However, the extremes of the intervals could be of exponential value with respect to n. Thus, each operation costs $O(n)$ time and, moreover, the output model is not efficient, as it is of quadratic size with respect to the size of the input. In the book of Spinrad [8] there is a short description of the second phase of the algorithm by Corneil et al., but the input ordering is replaced with a proper interval model of the graph. As it was described by Spinrad, the algorithm runs in $O(n+m)$ time, so it is quadratic with respect to the size of the input model. More recently, Gardi [4] gave a new proof of the "Proper = Unit" Theorem, which yields an $O(n)$ operations algorithm, but once again the extremes may have exponential value. Thus, in the worst case it takes $O(n^2)$ time and space.

In [4], Gardi states that it would be interesting to find a **direct** linear time and space algorithm for the representation problem when a proper interval model is given as input. By direct, Gardi means without the use of breath-first search. This problem involves two interesting questions by their own. The first one is to find a linear time and space algorithm for the representation problem. The second one is to do it without traversing any kind of tree. Lin and Szwarcfiter gave an answer to the former question in [6], where they develop an $O(n)$ time and space algorithm to transform a proper circular-arc model into a unit circular-arc model, if possible. When the input of the algorithm is a proper interval model, the output of the algorithm is a unit interval model. However, this algorithm is too general for unit interval graphs, and it requires several traversals of the input model and of trees, and the use of network flow techniques.

In the first part of this work we show how to implement the algorithm by Corneil et al. in $O(n)$ time and space, when the input is a proper interval model. In the second part we present a new $O(n)$ time and space algorithm for the representation problem without traversing any kind of tree. The algorithm is simple, it just requires two traversals of the input model. In each traversal only additions of variables are made, all between numbers in the range 0 to $O(n^2)$. Also, the procedure gives a new proof of the "Proper = Unit" Theorem. The work is based in a new characterization of unit interval graphs given in [5].

Now we describe the notation employed. Let $\mathcal{I} = \{I_i\}_{1 \leq i \leq n}$ be an interval model, i.e., a family of open intervals over the real line. For $I_i \in \mathcal{I}$, write $I_i = (s_i, t_i)$ and call *extremes* to s_i and t_i, while s_i is the *beginning point* and t_i is the *ending point*. The *extremes (resp. beginning points and ending points)* of $\mathcal{I}$ are those of all the intervals $I_i \in \mathcal{I}$. We will always assume that $s_1 < s_2 < \ldots < s_n$. Observe that if $\mathcal{I}$ is proper then also $t_1 < t_2 < \ldots < t_n$. A *segment* (e_1, e_2) is an interval of the real line defined by two consecutive extremes e_1, e_2 of $\mathcal{I}$, which are called the *left* and *right* extremes of the segment, respectively. An *s-sequence* is a maximal sequence of consecutive beginning points containing no ending points. Similarly, a *t-sequence* is a maximal sequence of consecutive ending points containing no beggining points. The leftmost s-sequence and the rightmost t-sequence are the *outer* sequences while the other sequences are the *inner* sequences. A beginning (resp. an ending) point is *inner* if it belongs to an inner sequence and it is *separated* if the extreme that is immediately to its left (resp. right) is an ending (resp. a beginning) point.

2 Corneil et al. algorithm

Let $\mathcal{I} = \{I_i\}_{1 \leq i \leq n}$ be a proper interval model of a connected graph G. The algorithm by Corneil et al. uses a special breath-first search tree of G, where the root vertex corresponds to I_1. Such a tree can be constructed from $\mathcal{I}$ as follows. For $i = 2, \ldots, n$, let $PREV(I_i)$ be the interval whose ending point appears first from s_i. Let T be an ordered tree with vertices $v_1, \ldots, v_n$ such that

(i) v_i is the parent of v_j if and only if $I_i = PREV(I_j)$ and

(ii) if v_i and v_j are siblings then v_i appears before v_j if and only if $i < j$.

A unit interval model can be computed from T in a simple manner. For $1 \leq i \leq n$, define $a_i = nl_i + k_i$ and $b_i = a_i + n$, where l_i is the level of v_i in T and k_i is the postition of v_i is a postorder traversal of T. Then, the model $\mathcal{I}' = \{(a_i, b_i) \mid 1 \leq i \leq n\}$ is a UIG model of G [3].

It is not hard to compute all the values of $PREV$ in $O(n)$ time. Therefore, Corneil et al. algorithm runs in $O(n)$ time and space.

3 Another Proof of the "Proper = Unit" Theorem

Let $\mathcal{I} = \{I_i\}_{1 \leq i \leq n}$ be a proper interval model of a connected graph G. The algorithm below includes new intervals into $\mathcal{I}$, constructing an extended model $\mathcal{I}'$ containing $\mathcal{I}$ and such that all the inner extremes of $\mathcal{I}'$ are separated.

Separation algorithm.

1. Let $\mathcal{I}' = \mathcal{I}$.
2. While there are non-separated inner ending points in $\mathcal{I}'$ do the following. Let t_i be the leftmost of such ending points and s_j be the beginning point immediately preceding t_i. Insert in $\mathcal{I}'$ a new interval (s, t), placing s immediately after t_i, and placing t immediately after t_j.
3. While there are non-separated inner beginning points in $\mathcal{I}'$ do the following. Let s_i be the rightmost of such beginning points and t_j be the ending

point immediately succeding s_i. Insert in $\mathcal{I}'$ a new interval (s, t), placing t immediately before s_{i-1}, and placing s immediately before s_j.

It can be proved that the above algorithm finishes and that $\mathcal{I}'$ is a proper interval model. The separated model can be used to obtain a unit interval model for graph G as follows from our new proof of Roberts' Theorem.

Theorem 3.1 ([7]) *Every proper interval graph admits a unit interval model.*

Proof. Let $\mathcal{I}$ be a proper interval model and apply the separation procedure to obtain a proper interval model $\mathcal{I}' \supseteq \mathcal{I}$ in where every inner extreme is separated. Let $\mathcal{I}^u = \{I_i^u\}_{1 \leq i \leq l}$ where $I_i^u = (2i - 1, 2(i + k))$ and k is number of intervals that intersect the leftmost interval of $\mathcal{I}'$. In [5] it is proved that $\mathcal{I}^u$ is a unit interval model with the same intersection graph as $\mathcal{I}'$. Then the submodel of $\mathcal{I}^u$ induced by the intervals corresponding to $\mathcal{I}$ is a unit interval model of the intersection graph of $\mathcal{I}$. □

4 Our Algorithm

Let $\mathcal{I} = \{I_i\}_{1 \leq i \leq n}$ be a proper interval model of a graph G and let $\mathcal{I}'$ be the output of the Separation Algorithm when applied to $\mathcal{I}$. It can be proved that there are at most $2n$ extremes of $\mathcal{I}' \setminus \mathcal{I}$ between consecutive extremes of $\mathcal{I}$. Therefore, any algorithm for constructing $\mathcal{I}'$ would require $O(n^2)$ steps, which precludes an $O(n)$ time algorithm — our initial goal. However, we do not have to construct $\mathcal{I}'$ explicitly. We need to find just the submodel of $\mathcal{I}'$ induced by the extremes of $\mathcal{I}$, which of course has $2n$ extremes.

Denote by $\ell(e)$ the length of the segment of $\mathcal{I}$, whose left extreme is $e \neq t_n$. Also, let $r(e)$ be number of extremes of $\mathcal{I}'$ that belong to the segment of $\mathcal{I}$, having e as its left extreme. We can determine precisely the unit model, by finding appropiate values for ℓ. With this purpose, we compute the $r(e)$ values for every extreme e, since we know that by setting $\ell(e) := r(e) + 1$ we obtain a unit interval model as in Theorem 3.1. The algorithm below computes the values of r and ℓ by simulating the Separation Procedure.

Proper to Unit Transformation.

Preprocesing. Define $\ell(e) := 1$ and $r(e) := 0$, for every extreme e of $\mathcal{I}$, except t_n. For an extreme e, denote by e^- and e^+ the extreme of $\mathcal{I}$ lying immediately at the left and right of e, respectively.

Stage 1. Traverse $\mathcal{I}$ from left to right, disregarding the beginning points. Let t_i be the current ending point, and s_j the closest beginning point at the left of t_i. Compute $\ell(t_i) := \ell(t_i) + 2r(t_i)$. If t_i is an inner extreme then also compute $r(t_j) := r(t_j) + r(t_i)$, and additionally if t_i^+ is an ending point then add one to both $\ell(t_i)$ and $r(t_j)$.

Stage 2. Traverse $\mathcal{I}$, as obtained from Stage 1, from right to left, disregarding the ending points. Let s_i be the current beginning point and t_j the closest ending point at the right of s_i. Compute $\ell(s_i^-) := \ell(s_i^-) + 2r(s_i^-)$. If s_i is an inner extreme then also compute $r(s_j^-) := r(s_j^-) + r(s_i^-)$, and additionally if s_i^- is a beginning point then add one to both $\ell(s_i^-)$ and $r(s_j^-)$.

At termination, the unit model is given by the lengths $\ell(e_k)$ of the segments. There is no difficulty to slightly modify the algorithm, so as to obtain the actual positions of all extremes, during Stage 2. The time and space required by the algorithm is clearly $O(n)$, while the length of the largest segment of the unit model is less than $2n$.

References

[1] K. P. Bogart and D. B. West. A short proof that "proper = unit". *Discrete Math.*, **201** (1-3), 21–23, (1999).

[2] D. G. Corneil. A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs. *Discrete Appl. Math.*, **138** (3), 371–379, (2004).

[3] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu and A. P. Sprague. Simple linear time recognition of unit interval graphs. *Inform. Process. Lett.*, **55** (2), 99–104, (1995).

[4] F. Gardi. The Roberts characterization of proper and unit interval graphs. *Discrete Math.*, **307** (22), 2906–2908, (2007).

[5] M. C. Lin, D. Rautenbach, F. J. Soulignac and J. L. Szwarcfiter. Powers of Cycles, Powers of Paths and Distance Graphs. Submitted 2008.

[6] M. C. Lin and J. L. Szwarcfiter. Unit circular-arc graph representations and feasible circulations. *SIAM J. Discrete Math.*, **22**, 409–423, (2008).

[7] F. S. Roberts, Indifference graphs. In: *Proof Techniques in Graph Theory (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor, Mich., 1968)*, 139–146, (1969)

[8] J. P. Spinrad, *Efficient graph representations*, American Mathematical Society (2003).

A Appendix

In this appendix we prove that the Separation Algorithm finishes after inserting at most $O(n^2)$ arcs, and that it outputs a separated proper interval model. We assume from now on that all graphs are connected.

Correctness of the Separation Algorithm

Theorem A.1 *Let G be a proper interval graph. Then G is an induced subgraph of a graph H, admitting a proper interval model in which all inner extremes are separated.*

Proof. Let $\mathcal{I}$ be a proper interval model of G. Perform the Separation Algorithm and let $\mathcal{I}' \supseteq \mathcal{I}$ be the obtained model. By induction, we show that all inner extremes of $\mathcal{I}'$ are separated. We start by discussing the effects of Step 2. We prove first that this step in fact terminates. Let t_i be be the leftmost non-separated inner ending point, (s, t) be the newly inserted interval, and s_j be the beginning point preceding t_i. Examine t_j and suppose first that t_j is an outer ending point. Then t is also an outer ending point and Step 2 will not separate t_j nor t. Then after the examination of t_i, the number of ending points still to be examined decreased by one. Next, discuss the alternative where t_j is an inner ending point. Then t is also so. Consequently, the number of ending points which we need still to examine remained unchanged. However, the new ending point t is closer to the leftmost outer ending point of $\mathcal{I}'$, in the sense that there are less t-sequences between the one that contains t and the outer t-sequence. Therefore Step 2 terminates.

Next, examine the actual effect of inserting (s, t). Clearly, t_i becomes separated, because s immediately follows it. Consequently, s is also separated. Furthermore. because the inclusion of t does not affect separated beginning points, we conclude that no newly non-separated beginning points have been created for separating t_i. Consequently, at the end of Step 2, all the ending points are separated, all the beginning points of the intervals inserted during Step 2 are also separated, while the separated beginning points of $\mathcal{I}$ have been preserved. Moreover, (s, t) can not contain nor be contained in any other interval (s_k, t_k) of $\mathcal{I}'$, otherwise (s_j, t_j) would also contain or be contained in (s_k, t_k), contradicting the model to be proper before the inclusion of (s, t).

Step 3 takes as input the model generated by Step 2 and transforms it into the final model $\mathcal{I}'$. The proof of its correctness is similar. We can conclude that after termination of Step 3, $\mathcal{I}$ is included in a proper interval model $\mathcal{I}'$, which has all its inner extremes separated. Then G is an induced subgraph of the intersection graph H of $\mathcal{I}'$, proving the theorem. □

The size of the separated model

Classify the segments (e_1, e_2) of a model into four types, according to the nature of e_1 and e_2. If e_1, e_2 are both beginning points (resp. ending points) then the segment is of type s-s (resp. t-t), while if e_1 is a beginning (resp. ending) point and e_2 is an ending (resp. beginning) point then the segment is of type s-t (t-s).

Theorem A.2 *Let $\mathcal{I}$ be a proper interval model and $\mathcal{I}'$ the model obtained from $\mathcal{I}$ after applying the Separation Algorithm. Then in $\mathcal{I}'$, there are less than $2n$ extremes of $\mathcal{I}' \setminus \mathcal{I}$ between any two consecutive extremes of $\mathcal{I}$.*

Proof. Consider first the effects of Step 2 in the Separation Algorithm. Let $\mathcal{I}_t$ be the set of intervals included in $\mathcal{I}$ by Step 2. Recall that this step iteratively considers non-singleton t-sequences T in the model and separates these ending points by including $|T| - 1$ new intervals. We attach a label to each ending point of $\mathcal{I} \cup \mathcal{I}_t$, as follows. The ending points $t_i \in \mathcal{I}$ are assigned the label i. Further, following the progress of the Step 2, whenever a new interval (s, t) is included for the purpose of separating an ending point having label i, from its predecessor in the model, the new ending point t also gets the label i. Clearly, there can be several ending points in the model sharing the same label. However, in any t-sequence all the ending points have distinct labels. Consequently, the size of any t-sequence at any stage of the algorithm is less than n. On the other hand, every non-singleton t-sequence T is separated by including one new beginning point of $\mathcal{I}_t$ between every consecutive pair of ending points of T. Consequently, T is transformed into a sequence of size $2|T| - 1$, in which the ending points and beginning points alternate. All but one of these extremes may perhaps belong to $\mathcal{I}_t$, but they are preceded and followed by a beginning point of $\mathcal{I}$. Consequently, in $\mathcal{I} \cup \mathcal{I}_t$ there are less than $2n$ extremes of $\mathcal{I}_t$ between any two extremes of $\mathcal{I}$. More precisely, less than i ending points of $\mathcal{I}_t$ have been included in the segment of $\mathcal{I}$ whose left extreme is t_i. Furthermore, all extremes of $\mathcal{I}_t$ have been included in segments of $\mathcal{I}$ of types t-t or t-s.

Next, we apply Step 3 to the model $\mathcal{I} \cup \mathcal{I}_t$. Denote by $\mathcal{I}_s$ the set of intervals introduced in the model by Step 3. That is, the final model is $\mathcal{I}' = \mathcal{I} \cup \mathcal{I}_t \cup \mathcal{I}_s$. We know that Step 2 separates all ending points, while preserving the already separated beginning points, and neither increasing the size nor creating new non-trivial s-sequences. Consequently, Steps 2 and 3 are independent. Furthermore, in $\mathcal{I}'$ there can be no three consecutive extremes, such that the first and the third belong to $\mathcal{I}_t$ and the second belongs to $\mathcal{I}_s$, or vice versa. So, similarly as above, we can conclude that in $\mathcal{I}'$ there are less than $2n$ extremes of $\mathcal{I}_s$ between any two consecutive extremes of $\mathcal{I}$. More precisely, less than $n - i$ beginning points of $\mathcal{I}_s$ have been introduced in the segment of $\mathcal{I}$ whose right extreme is s_i. Furthermore, all extremes of $\mathcal{I}_s$ have been included in segments of $\mathcal{I}$ of types s-s or t-s.

Finally, we examine the total number of extremes of $\mathcal{I}_t \cup \mathcal{I}_s$ which have been introduced in a segment z of $\mathcal{I}$, according to the type of z. If z is an s-s type segment, less than $2n$ extremes of $\mathcal{I}_s$ and zero extremes of $\mathcal{I}_t$ have been introduced. Similarly, if z is of type t-t then zero extremes of $\mathcal{I}_s$ and less than $2n$ extremes of $\mathcal{I}_t$ have been introduced. In an s-t type segment of $\mathcal{I}$ no extremes of $\mathcal{I}_t \cup \mathcal{I}_s$ can be included at all. It remains to examine the case when z is a t-s segment. Let t_i and s_j be the left and right extremes of z, respectively. We know that less than i ending points of $\mathcal{I}_t$ and less than $n - j$ beginning points of $\mathcal{I}_s$ have been included in z. That is, less than $2(i + n - j)$ extremes of $\mathcal{I}_s \cup \mathcal{I}_t$. However, we also know that $i < j$, because s_i must precede t_i. Consequently, there are less than $2n$ extremes of $\mathcal{I}' \setminus \mathcal{I} = \mathcal{I}_t \cup \mathcal{I}_s$ between any two consecutive extremes of $\mathcal{I}$. $\square$