Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Discrete Mathematics 37 (2011) 219–224

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

Computing and Counting Longest Paths on Circular-Arc Graphs in Polynomial Time

George B. Mertzios [a,1] Ivona Bezáková [b,2]

[a] *Caesarea Rothschild Institute for Computer Science, University of Haifa, Israel*

[b] *Department of Computer Science, Rochester Institute of Technology, NY, USA*

Abstract

The longest path problem asks for a path with the largest number of vertices in a given graph. The first polynomial time algorithm (with running time $O(n^4)$) has been recently developed for interval graphs. Even though interval and circular-arc graphs look superficially similar, they differ substantially, as circular-arc graphs are not perfect. In this paper, we prove that for *every* path P of a circular-arc graph G, we can appropriately "cut" the circle, such that the obtained (not induced) interval subgraph G' of G admits a path P' on the same vertices as P. This non-trivial result is of independent interest, as it suggests a generic reduction of a number of path problems on circular-arc graphs to the case of interval graphs with a multiplicative linear time overhead of $O(n)$. As an application of this reduction, we present the first polynomial algorithm for the longest path problem on circular-arc graphs, which turns out to have the same running time $O(n^4)$ with the one on interval graphs, as we manage to get rid of the linear overhead of the reduction. This algorithm computes in the same time an n-approximation of the number of different vertex sets that provide a longest path; in the case where G is an interval graph, we compute the exact number. Moreover, our algorithm can be directly extended with the same running time to the case where every vertex has an arbitrary positive weight.

Keywords: Circular-arc graphs, interval graphs, longest path problem, counting, approximation algorithm, dynamic programming.

[1] Email: `mertzios@cs.technion.ac.il`

[2] Email: `ib@cs.rit.edu`

doi:10.1016/j.endm.2011.05.038

1 Introduction

The Hamiltonian path problem, i.e. the problem of deciding whether a given graph contains a simple path that visits all its vertices, is one of the most well known and well studied NP-complete problems [2], with numerous applications. The most natural optimization version of this problem is the longest path problem, which has been also extensively studied over the past several decades as it has a number of applications, e.g. in computational biology [3].

Very recently, prompted by an open problem statement in [10], a polynomial time algorithm has been developed for interval graphs [4] with running time $O(n^4)$. This algorithm has been followed by two independent algorithms for the longest path problem on the much greater class of *cocomparability* graphs (one with running time $O(n^4)$ [9] and one with running time $O(n^8)$ [5]).

Circular-arc graphs naturally extend interval graphs: interval graphs are the intersection graphs of intervals on the real line, while circular-arc graphs are intersection graphs of arcs on a circle. Such an intersection model is called an *interval* (resp. *circular-arc*) *representation*. Although circular-arc graphs look superficially similar to interval graphs, several combinatorial problems (e.g. minimum coloring) behave very differently on these classes of graphs. The main reason for that is that there are two ways to travel from one point to another on a circle, as opposed to just one on the real line.

In this article we present the first polynomial algorithm for the longest path problem on circular-arc graphs by showing that the problem reduces to the case of interval graphs. The significance of our reduction comes from the fact that a path in a circular-arc graph can have a spiral-like form and this makes it hard to "cut" the circle to create an interval graph that maintains the length of a longest path. Note here that also other problems on circular arc graphs have been reduced to the interval graph case. However, for problems that search for a set (e.g. an independent set) the reduction is fairly natural, since "cutting" the circle does not destroy the set. On the other hand, "cutting" a sequence (such as a path) breaks the sequence into many parts. In this article we overcome this issue by showing that for *every* path P of a circular-arc graph G, we can appropriately "cut" the circle, such that the obtained (not necessarily induced) interval subgraph admits a path P' on the same vertices as P.

This result suggests a generic reduction of a number of path problems (such as the Hamiltonian and the longest path problems) on circular-arc graphs to the corresponding problem on interval graphs with a multiplicative linear time overhead of $O(n)$. However, by exploiting deeper the structure of circular-arc graphs, we manage to get rid of this overhead for the longest path problem.

In particular, we introduce the crucial notion of *normal* paths in circular-arc graphs, which can be thought of as *"monotone representatives"* of all paths. Indeed, we prove that every path P of a circular-arc graph G can be restructured as a normal path on the same vertices. Our dynamic programming algorithm searches for a *longest normal* path in a circular-arc graph with n vertices and has the same running time $O(n^4)$ as the one on interval graphs.

Our algorithm significantly simplifies the approach of [4], by eliminating the "dummy vertices" that were essential in [4] [3]. This simplification allows us to compute in the same time bound also the total number of longest normal paths in the given circular-arc representation, which constitutes an n-approximation of the (exponentially large in worst case) number of vertex sets that provide a longest path of G. In the case where G is an interval graph, this computation is exact. The motivation for studying counting problems is that they are often related to their sampling counterpart [6]. In particular, sampling of paths is mainly used in a variety of heuristics, e.g. in motion planning.

Finally, in contrast to [4], all the above results can be directly extended (with the same running time as well) to the case where every vertex is assigned a positive weight. However, for simplicity of the presentation, we present here only the unweighted case. For a full version of this article, see [8].

2 Reduction to the case of interval graphs

Let $G = (V, E)$ be a circular-arc graph. We denote the arc I_v of a vertex $v \in V$ with endpoints ℓ_v and r_v by $I_v = [\ell_v, r_v]$. We always consider the arcs in the *counter-clockwise* direction, i.e. ℓ_v (resp. r_v) is the first (resp. last) point of $[\ell_v, r_v]$ (also referred to as the *left* and *right* endpoint of $[\ell_v, r_v]$, respectively) when traveled in the counter-clockwise direction. The intuition for the terminology comes from imagining standing on the arc and facing the center of the circle; then ℓ_v is on the left and r_v on the right endpoint of $[\ell_v, r_v]$, respectively. Observe here that, if $uv \in E$ for two vertices $u, v \in V$, then $r_u \in I_v$ or $r_v \in I_u$ (or both).

Given an interval graph $G = (V, E)$ along with an interval representation R, we can define an ordering of V by sorting the intervals in R according to their right endpoints. In such a *right-end ordering* π of an interval graph G, we can define a total order $<_\pi$ in V such that $u <_\pi v$ for two vertices $u, v \in V$ if u appears to the left of v in π. Similarly we define the *right-end circular ordering* $\pi = (u_0, u_1, \ldots, u_{n-1})$ of the set V of vertices of a circular-arc graph $G = (V, E)$, which results after sorting the arcs of a circular-arc representation R according to their right endpoints. Note that, in contrast

[3] The algorithm of [4] has three phases, during which it adds these dummy vertices to construct a second auxiliary graph.

to interval graphs, we can not define for circular-arc graphs a total order $<_\pi$ of V, since there are two ways to travel from one point to another on a circle. For any $i \in \mathbb{Z}$, we may refer in the following to the vertex $u_{(i \bmod n)}$ (resp. to the points $\ell_{u_{(i \bmod n)}}$ and $r_{u_{(i \bmod n)}}$ of the circle) as u_i (resp. as ℓ_{u_i} and r_{u_i}).

Furthermore, for every vertex u_i of a circular-arc graph G, we define a (not necessarily induced) interval subgraph G_{u_i} obtained from G by *"cutting"* the circle and the arcs of a circular-arc representation R of G appropriately, as follows. Let u_i be a vertex of G and $I_{u_i} = [l_{u_i}, r_{u_i}]$ be the corresponding arc in R. Consider an arbitrary point x_i of the circle between $r_{u_{i-1}}$ and r_{u_i} in the *counter-clockwise* direction. Then, replace in R every arc $I_{u_j} = [l_{u_j}, r_{u_j}]$ such that $x_i \in I_{u_j}$ by the arc $[x_i, r_{u_j}]$; denote by G_{u_i} the resulting interval graph. Intuitively, G_{u_i} is the interval graph obtained by "cutting" R immediately to the left of r_{u_i}. We are now ready to state the main theorem of this section.

Theorem 2.1 *Let $G = (V, E)$ be a circular-arc graph, R be a circular-arc representation of G, and P be any path of G. Then there exists a vertex $v \in V$ and a path P' with $V(P') = V(P)$, such that P' is also a path of the interval graph G_v.*

This structural theorem suggests a generic reduction of a number of path problems on circular-arc graphs to the corresponding problem on interval graphs. For instance, in order to solve the Hamiltonian or the longest path problem on a circular-arc graph $G = (V, E)$ with n vertices, it suffices to execute n times *any* known algorithm on interval graphs, once for each interval graph G_v, where $v \in V$. This implies an immediate reduction to the Hamiltonian and the longest path problems on interval graphs with a multiplicative linear time overhead of $O(n)$.

3 Computation and counting of longest paths in circular-arc graphs

In this section we exploit deeper the structure of circular-arc graphs, in order to get rid of this linear time overhead for the longest path problem. In particular, we present the first polynomial algorithm (with running time $O(n^4)$) that computes a longest path of a circular-arc graph $G = (V, E)$ with n vertices. In order to present our algorithm, we introduce the crucial notion of a *normal path* of a circular-arc graph (cf. Definition 3.3). Using our structural Theorem 2.1, we are able to prove the basic property that for every path P of a circular-arc graph G, there exists another path P' on the same vertices, which is normal in G, cf. Theorem 3.4. Therefore, normal paths can be thought as "representatives" of several non-normal paths. Recall first the notion of a normal path in an interval graph G, given a right-end ordering π of G [4]. A similar notion of a normal path in interval graphs has appeared in [7] (referred to as a *straight* path), as well as in [1].

Definition 3.1 ([4]) Let $G = (V, E)$ be an interval graph and π be a right-end ordering of G. A path $P = (v_1, v_2, \ldots, v_k)$ of G is *normal* if v_1 is the leftmost vertex of $V(P)$ in π and v_i is the leftmost vertex of $N(v_{i-1}) \cap \{v_i, v_{i+1}, \ldots, v_k\}$ in π, for every $i = 2, \ldots, k$.

Lemma 3.2 (see [4, 7]) *Let $G = (V, E)$ be an interval graph, π be a right-end ordering of G, and P be a path of G. Then, there exists a normal path P' of G such that $V(P') = V(P)$.*

In the next definition we extend the notion of normal paths to the case of circular-arc graphs. For more details, we refer to [8].

Definition 3.3 Let $G = (V, E)$ be a circular-arc graph and π be a circular right-end ordering of G. A path P of G is *normal* if P is a normal path in the interval graph G_u for some vertex $u \in V$ (with respect to the right-end ordering σ of G_u induced by π).

Normal paths in circular-arc graphs behave similarly to normal paths in interval graphs. Indeed, the next theorem follows by the structural Theorem 2.1 and by the results of [4]. For more details see [8].

Theorem 3.4 *Let $G = (V, E)$ be a circular-arc graph, π be a circular right-end ordering of G, and P be a path of G. Then there exists a normal path P' of G with $V(P') = V(P)$.*

We are now ready to state the two main theorems of this section. The proofs of these theorems rely on the construction of certain subgraphs of G (two subgraphs for every ordered pair of vertices), which we use by dynamic programming to recursively construct longest normal paths.

Theorem 3.5 *Let $G = (V, E)$ be a circular-arc graph with n vertices and π be a right-end circular ordering of G. Then, a longest path P of G and the number N of all longest normal paths of G can be computed in $O(n^4)$ time.*

Theorem 3.6 *Let $G = (V, E)$ be a circular-arc graph with n vertices and π be a right-end circular ordering of G. Then, the number N computed in Theorem 3.5 is an n-*approximation *of the number of sets $S \subseteq V$, such that $V(P) = S$ for a longest path P of G. Furthermore, if G is an interval graph, then the* exact *number of such sets $S \subseteq V$ can be computed in $O(n^4)$ time.*

The bound of n for the approximation ratio of Theorem 3.6 is tight. For instance, let the circular-arc graph $G = (V, E)$ be an induced circle with n vertices. The algorithm of Theorem 3.5 will return $N = n$, one for each Hamiltonian path of G, while $S = V$ is the only set of vertices that provides a longest path of G. Moreover, as the following lemma states, there can be exponentially many such different sets $S \subseteq V$ in the worst case.

Lemma 3.7 *There exists a circular-arc graph $G = (V, E)$ with n vertices, such that there exist $2^{O(n)}$ sets $S \subseteq V$, such that $V(P) = S$ for a longest path P of G.*

References

[1] P. Damaschke. Paths in interval graphs and circular arc graphs. *Discrete Mathematics*, 112(1-3):49–64, 1993.

[2] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[3] I. S. Hermann Cuntz, Alexander Borst. Optimization principles of dendritic structure. *Theoretical Biology and Medical Modelling*, 4(08):21, 2007.

[4] K. Ioannidou, G. B. Mertzios, and S. D. Nikolopoulos. The longest path problem has a polynomial solution on interval graphs. *Algorithmica.* To appear.

[5] K. Ioannidou and S. D. Nikolopoulos. The longest path problem is polynomial on cocomparability graphs. In *Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, 2010. To appear.

[6] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43(2-3):169–188, 1986.

[7] J. M. Keil. Finding Hamiltonian circuits in interval graphs. *Information Processing Letters*, 20:201–206, 1985.

[8] G. B. Mertzios and I. Bezáková. Computing and counting the longest paths on circular-arc graphs in polynomial time. Technical report. Available at www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2010/CS/CS-2010-18.

[9] G. B. Mertzios and D. G. Corneil. A simple polynomial algorithm for the longest path problem on cocomparability graphs. Technical report available at http://arxiv.org/abs/1004.4560.

[10] R. Uehara and Y. Uno. Efficient algorithms for the longest path problem. In *15th Annual Int. Symp. on Alg. and Comp. (ISAAC)*, pages 871–883, 2004.