Discrete Applied Mathematics 164 (2014) 383–399

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Computing and counting longest paths on circular-arc graphs in polynomial time

George B. Mertzios [a,*], Ivona Bezáková [b]

[a] School of Engineering and Computing Sciences, Durham University, Durham, United Kingdom
[b] Department of Computer Science, Rochester Institute of Technology, Rochester, NY, USA

ARTICLE INFO

Article history:
Received 28 September 2011
Received in revised form 8 June 2012
Accepted 21 August 2012
Available online 13 September 2012

Keywords:
Circular-arc graphs
Interval graphs
Longest path problem
Counting
Approximation algorithm
Dynamic programming

ABSTRACT

The longest path problem asks for a path with the largest number of vertices in a given graph. In contrast to the Hamiltonian path problem, until recently polynomial algorithms for the longest path problem were known only for small graph classes, such as trees. Recently, a polynomial algorithm for this problem on interval graphs has been presented in Ioannidou et al. (2011) [19] with running time $O(n^4)$ on a graph with n vertices, thus answering the open question posed in Uehara and Uno (2004) [32]. Even though interval and circular-arc graphs look superficially similar, they differ substantially, as circular-arc graphs are not perfect; for instance, several problems – e.g. coloring – are NP-hard on circular-arc graphs, although they can be efficiently solved on interval graphs. 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. In addition, by exploiting deeper the structure of circular-arc graphs, we manage to get rid of the linear time overhead of the reduction, and thus this algorithm turns out to have the same running time $O(n^4)$ as the one on interval graphs. Our algorithm, which significantly simplifies the approach of Ioannidou et al. (2011) [19], computes in the same time an n-approximation of the (exponentially large in worst case) 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, in contrast to Ioannidou et al. (2011) [19], this algorithm can be directly extended with the same running time to the case where every vertex has an arbitrary positive weight.

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 [13], with numerous applications. The most natural optimization version of this problem is the longest path problem, where the task is to find a path with the largest number of vertices. This problem has been also extensively studied over the past several decades and it plays an important role in a number of applications, for instance in computational biology [29,7].

* Corresponding author. Tel.: +44 1913342429; fax: +44 1913341701.
E-mail addresses: george.mertzios@gmail.com, george.mertzios@durham.ac.uk (G.B. Mertzios), ib@cs.rit.edu (I. Bezáková).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.08.024

In addition to both problems being NP-hard for general graphs, several prohibitive inapproximability results for the longest path problem appeared in [22]. In particular, for any $\varepsilon \in (0, 1)$, it is NP-hard to compute a path of length $n - n^{\varepsilon}$ in a graph with n vertices, even if it is known that the graph admits a Hamiltonian path. Moreover, there is no polynomial time constant-factor approximation algorithm for the longest path problem unless $P = \text{NP}$ [22]. To the best of our knowledge, the best known approximation algorithms achieve approximation ratio $O(n(\log\log n/\log n)^2)$ for general graphs [3]. Furthermore, the Hamiltonian path (and thus also the longest path) problem is NP-hard on many restricted classes of graphs, namely split graphs, chordal bipartite graphs, split strongly chordal graphs, directed path graphs, circle graphs, planar graphs, and grid graphs; see e.g. [25] for a list of related works.

On the positive side, polynomial time algorithms have been developed for the Hamiltonian path (and the related Hamiltonian cycle) problem on several graph classes, notably proper interval graphs [2], interval graphs [1,23], circular-arc graphs [8,30,18], and cocomparability graphs [9,10]. Another natural generalization of the Hamiltonian path problem, namely the minimum path cover problem, has also received considerable attention in the literature. The objective is to find the smallest number of vertex disjoint simple paths that cover all the vertices; polynomial time algorithms were given for this problem on interval graphs [1,6], cocomparability graphs [9], and lately also on circular-arc graphs [17,18].

In contrast to the Hamiltonian path problem, until recently only a few polynomial algorithms were known for the longest path problem, and these were restricted to trees [5], weighted trees and block graphs [32], bipartite permutation graphs [33], and ptolemaic graphs [31]. Very recently, prompted by an open problem statement in [32], a polynomial time algorithm has been developed for interval graphs with running time $O(n^4)$ on a graph with n vertices [19]. This algorithm has been followed by two independent polynomial algorithms for the longest path problem on the much greater class of *cocomparability* graphs (one with running time $O(n^4)$ [26] and one with running time $O(n^8)$ [20]).

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. That is, a graph G is interval (resp. circular-arc) if its vertices can be put in a one-to-one correspondence with a family of intervals (resp. arcs) on the real line (resp. on the circle), such that two vertices are adjacent in G if and only if their corresponding intervals (resp. arcs) intersect. Such an intersection model with intervals (resp. arcs) of an interval (resp. circular-arc) graph G is also called an *interval* (resp. *circular-arc*) *representation* of G. Several NP-complete problems have been studied on these graph classes, for example, a maximum independent set and a maximum clique be found in polynomial time [16], while, for example, the achromatic number problem is NP-complete for both classes of graphs [4].

Although circular-arc graphs look superficially similar to interval graphs, several combinatorial problems behave very differently on these classes of graphs. For example, the coloring problem is NP-complete for circular-arc graphs [14] while it can be solved greedily in linear time on interval graphs.

All optimization problems have a corresponding counting version. For the case of Hamiltonian paths, the counting version asks for the overall number of all Hamiltonian paths in a given graph. Counting problems are related to sampling [21], where, for example, for the case of Hamiltonian paths, the task is to sample one of the Hamiltonian paths uniformly at random. Sampling plays an important role in machine learning and other applied areas, and the problems of counting and sampling of paths in graphs have been studied extensively; see e.g. [24]. The problem of counting all Hamiltonian paths is #*P*-complete for general graphs [11], while approximation algorithms were given for several special classes of graphs, including dense graphs for which there exists a fully polynomial randomized approximation scheme (FPRAS) [11], and nearly regular graphs [12]. An importance sampling based framework, combined with cross and minimum entropy methods, achieved fast empirical results, closely approximating the optimum [28]. The problem of counting and sampling paths in graphs, especially the scenario of self-avoiding walks in lattice graphs, has been researched since the 1960s; see e.g. [24]. Most of the algorithms are heuristic without proofs of correctness; notable exceptions include [27].

Our contribution. 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 and it runs in time $O(n^4)$ on a graph with n vertices. Our algorithm significantly simplifies the approach of [19] that shows polynomial time solvability of this problem on interval graphs. This simplification consists of the elimination of the introduced "dummy vertices" that