Information Processing Letters 116 (2016) 391–395

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A linear time algorithm to compute a maximum weighted independent set on cocomparability graphs

Ekkehard Köhler [a], Lalla Mouatadid [b]

[a] Brandenburg University of Technology, 03044 Cottbus, Germany
[b] University of Toronto, Toronto, ON M5S 2J7, Canada

ARTICLE INFO

Article history:
Received 16 January 2015
Received in revised form 30 November 2015
Accepted 3 December 2015
Available online 11 December 2015
Communicated by Ł. Kowalik

Keywords:
Maximum weight independent set
Cocomparability graphs
Posets
Minimum weight vertex cover

ABSTRACT

The maximum weight independent set (WMIS) problem is a well-known NP-hard problem. It is a generalization of the maximum cardinality independent set problem where all the vertices have identical weights. There is an $\mathcal{O}(n^2)$ time algorithm to compute a WMIS for cocomparability graphs by computing a maximum weight clique on the corresponding complement of the graph [1]. We present the first $\mathcal{O}(m+n)$ time algorithm to compute a WMIS directly on the given cocomparability graph, where m and n are the number of edges and vertices of the graph respectively. As a corollary, we get the minimum weight vertex cover of a cocomparability graph in linear time as well.

1. Introduction

Given a graph $G(V, E)$, an *independent set* (also called *stable set*) $I \subseteq V$, is a subset of pairwise non-adjacent vertices. For $G(V, E, w)$ being a graph together with a weight function $w : V \to \mathbb{R}$, the weighted maximum independent set (WMIS) problem asks for an independent set $I \subseteq V$ such that $\sum_{v \in I} w(v)$ is maximum. This problem is a generalization of the maximum cardinality independent set problem where all vertices have equal weights. The WMIS problem has been widely studied as it naturally arises in different applications, such as scheduling [2], combinatorial auctions [3], molecular biology [4] to name a few. The problem is NP-hard for arbitrary graphs; we restrict ourselves to the class of cocomparability graphs and present a linear time algorithm for this case.

Let $G(V, E)$ be a graph where $n = |V|$ and $m = |E|$, and let $N(v)$ (resp. $N[v]$) denote the open (resp. closed) neighbourhood of vertex v; $N(v) = \{u \in V \mid uv \in E\}$ and $N[v] = N(v) \cup \{v\}$. A graph $G(V, E)$ is a *cocomparability* graph if its complement is a *comparability* graph. A graph $G(V, E)$ is a comparability graph if E admits an acyclic transitive orientation. That is, if $uv, vw \in E$, and they are oriented $u \to v$, and $v \to w$ then uw has to be contained in E and must be oriented $u \to w$. Cocomparability graphs are a subfamily of perfect graphs and have been well studied. Many problems on this graph class are solved by computing the complement of the given graph, and translating the problem into its complement problem on comparability graphs. This transformation necessitates $\Omega(n^2)$ computation, whereas for some problems direct solutions in $\mathcal{O}(n+m)$ are possible. Finding a WMIS in a cocomparability graph, for example, is equivalent to finding a maximum weighted clique in its complement. There exists a linear time dynamic programming algorithm to compute the maximum weight clique on a comparability graph, given a transitive orientation of the edges [1]. This implies an $\mathcal{O}(n^2)$ time algorithm to compute a WMIS on a cocomparability graph.

The idea to solve problems directly on cocomparability graphs instead of going over to the complement graph has been around for a while and a number of problems have

E-mail addresses: ekkehard.koehler@b-tu.de (E. Köhler), lalla@cs.toronto.edu (L. Mouatadid).

http://dx.doi.org/10.1016/j.ipl.2015.12.001
0020-0190/

been solved in this way, such as domination [5] and the minimum feedback vertex set problem [6]. Recently, there have been new approaches for solving problems directly on the given cocomparability graph. In [7] for instance, Mertzios and Corneil presented the first polynomial time algorithm to solve the longest path problem on cocomparability graphs, and in [8] Corneil et al. gave the first near linear time certifying algorithm to compute a minimum path cover, and thus a Hamilton path (if one exists), directly on cocomparability graphs. Motivated by this idea, we present the first linear time algorithm to compute a WMIS directly on a cocomparability graph. The unweighted case has been known to take $\mathcal{O}(m+n)$ time [9]. As a corollary to our result, we also get the minimum weight vertex cover of a cocomparability graph in linear time.

Cocomparability graphs have a vertex ordering characterization, known as a *cocomparability order* σ, or an *umbrella-free* order; more precisely, an ordering $\sigma = v_1 \prec_\sigma v_2 \prec_\sigma \cdots \prec_\sigma v_n$ is a cocomparability order iff for any triple $u \prec_\sigma v \prec_\sigma w$ with $uw \in E$, either $uv \in E$ or $vw \in E$ or both [5]. In other words, σ does not contain an *umbrella*, which is a triple of vertices $u \prec_\sigma v \prec_\sigma w$ with $uw \in E$ but $uv, vw \notin E$. In [10], McConnell and Spinrad presented an algorithm to compute such an ordering in $\mathcal{O}(m+n)$ time. We use their algorithm, denoted as $\sigma \leftarrow ccorder(G)$ to compute such an ordering.

This paper is organized as follows. In Section 2 we present an overview of the algorithm, followed by its formal description and in Section 3, we prove the correctness of the algorithm, present implementation details and the complexity analysis.

2. The algorithm

Let $G(V, E, w)$ be a weighted cocomparability graph and let $X \subseteq V$ be the subset of vertices with non-positive weight, i.e., $X = \{v : w(v) \leq 0\}$. Any vertex $v \in X$ that belongs to an independent set S will not increase the total weight of S. Therefore if $X \neq \emptyset$, we can restrict ourselves to $G[V \setminus X]$, which is also a cocomparability graph that can easily be computed in $\mathcal{O}(m+n)$ time.

Suppose $G(V, E, w)$ is a cocomparability graph with positive weight function $w : V \to \mathbb{R}_{>0}$. Using the algorithm in [10], we compute a cocomparability order σ of V in $\mathcal{O}(m+n)$ time where $\sigma = v_1 \prec_\sigma v_2 \prec_\sigma \cdots \prec_\sigma v_n$. We then construct a new permutation τ of the vertices as follows: we process one vertex at a time according to the order imposed by σ from left to right. To each v_i we associate an updated weight $\tilde{w}(v_i)$ and an [independent] set $\mathcal{S}_{v_i}$ (containing v_i) of total weight $\tilde{w}(v_i)$. The vertices from v_1 to v_i are then reordered such that the new ordering is non-decreasing with respect to their updated weights $\tilde{w}$; τ_i denotes the resulting permutation on the processed vertices $v_1, \ldots, v_i$. In other words, for vertices v_k, v_j $(1 \leq k, j \leq i, k \neq j)$,

$$\text{if } v_k \prec_{\tau_i} v_j \text{ then } \tilde{w}(v_k) \leq \tilde{w}(v_j). \tag{1}$$

Initially τ_1 is just $\{v_1\}$, $\tilde{w}(v_1) = w(v_1)$, and $\mathcal{S}_{v_1} = \{v_1\}$. For every vertex v_i $(i > 1)$, we scan through τ_{i-1} from right to left, looking for the rightmost non-neighbour of v_i.

Algorithm 1: CCWMIS

Input: $G = (V, E, w)$, $w : V \to \mathbb{R}_{>0}$
Output: A maximum weight independent set together with its weight

```
σ ← ccorder(G(V, E)) ;                          // σ = (v_1, v_2, ..., v_n)
for i ← 1 to n do
    w̃(v_i) ← w(v_i);
    S_{v_i} ← {v_i};
τ_1 ← (v_1);                                    // Constructing τ_i
for i ← 2 to n do
    Choose u to be rightmost non-neighbour of v_i with respect to τ_{i−1};
    if u exists then
        w̃(v_i) ← w(v_i) + w̃(u);
        S_{v_i} ← {v_i} ∪ S_u;
    τ_i ← insert(v_i, τ_{i−1});
        // Insert v_i into τ_{i−1} such that τ_i stays ordered with respect to w̃(·)
z ← the rightmost vertex in τ_n;
return S_z and w̃(z);
```

Let u denote such a vertex (if it exists); $\tilde{w}(v_i)$ and $\mathcal{S}_{v_i}$ are then set to

$$\tilde{w}(v_i) = w(v_i) + \tilde{w}(u)$$

$$\mathcal{S}_{v_i} = \{v_i\} \cup \mathcal{S}_u.$$

If no such vertex u exists, then

$$\tilde{w}(v_i) = w(v_i)$$

$$\mathcal{S}_{v_i} = \{v_i\}.$$

τ_i is the permutation of $\{v_1, \ldots, v_i\}$ created by inserting v_i into τ_{i-1} such that (1) holds and thus preserving the non-decreasing order of the updated weights. Since the weights are strictly positive, it is easy to see that $\tilde{w}(v_i) = w(v_i) + \tilde{w}(u)$ implies $\tilde{w}(v_i) > \tilde{w}(u)$ and thus also implies $u \prec_{\tau_i} v_i$.

Notice that if there exists a vertex x in τ_{i-1} such that $\tilde{w}(x) = \tilde{w}(v_i)$, then v_i is inserted to the right of vertex x in τ_{i-1}. We say that a vertex v_i has been *processed* as soon as it is inserted into τ_{i-1} and thus τ_i is created. When all vertices are processed, we have determined τ_n. We return $\mathcal{S}_z$ as a maximum weight independent set of G and $\tilde{w}(z)$ as its corresponding total weight, where z is the rightmost vertex in τ_n.

We now present the formal description of the algorithm; recall that $ccorder(G)$ is the procedure presented in [10] to compute a cocomparability order in $\mathcal{O}(m+n)$ time.

We illustrate the algorithm using a cocomparability graph and a corresponding cocomparability ordering given in Fig. 1. Table 1 shows how τ_i is created by the algorithm. Recall that the vertices are processed in σ's order and vertex v_i is inserted into τ_{i-1} according to its updated weight.

3. Correctness, complexity analysis, and robustness

Recall that $\mathcal{S}_{v_i}$ is the set associated with v_i recursively constructed by finding u, the rightmost non-neighbour of v_i in τ_{i-1}; in other words $\mathcal{S}_{v_i}$ denotes a set of vertices including v_i whose weights sum up to $\tilde{w}(v_i)$. Therefore