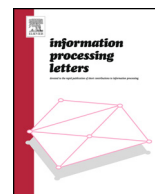




ELSEVIER

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl


A faster algorithm for the cluster editing problem on proper interval graphs


 Min Chih Lin^{a,*}, Francisco J. Soulignac^b, Jayme L. Szwarcfiter^{c,d}
^a CONICET and Departamento de Computación, Universidad de Buenos Aires, Argentina

^b CONICET and Departamento de Ciencia y Tecnología, Universidad Nacional de Quilmes, Argentina

^c Instituto de Matemática, NCE and COPPE, Universidade Federal do Rio de Janeiro, Brazil

^d Instituto Nacional de Metrologia, Qualidade e Tecnologia, Brazil

ARTICLE INFO

Article history:

Received 4 December 2014

Received in revised form 21 July 2015

Accepted 21 July 2015

Available online 26 July 2015

Communicated by B. Doerr

Keywords:

Graph algorithms

Cluster editing problem

Proper interval models

Linear space algorithm

ABSTRACT

We develop a linear-space $O(n + m)$ time algorithm to solve the cluster editing problem for proper interval models, where n and m are the number of vertices and edges of the represented graph.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Data clustering based on graphs is a relevant topic with applications in different areas, for instance, computational biology. In general terms, the problem consists of transforming a graph into a set of *clusters*, that is a subset of vertices similar in some way. The process of transformation consists of removing or adding edges to the original graph. In general, the term *edge edit* is employed to mean either an edge removal or addition.

The most common and extensively studied clustering problem is that of transforming a graph into a set of disjoint cliques, by edge edits. In this case, each cluster corresponds to a disjoint clique. A natural parameter to measure the effort of this transformation is that of counting the number of edits. The corresponding CLUSTER EDITING prob-

lem, whose goal is to determine whether an input graph G can be transformed into a set of disjoint cliques by at most k edge edits, was proven NP-hard by Křivánek and Morávek [1] (see also [2]).

In spite of its hardness, the problem was proved to be fixed-parameter tractable, a result that follows from [3]. It admits a kernel having at most $2k$ vertices, which can be constructed in $O(nm)$ time, employing an algorithm by Chen and Meng [4]. Alternatively, a kernel with $O(k^2)$ vertices can be obtained in $O(n + m)$ time [5]. Other possible parameterizations have been considered by Damaschke [6] and Komusiewicz and Uhlmann [7]. An exact algorithm of complexity $O^*(1.62^k)$ has been described by Böcker [8]. ILP formulations of the CLUSTER EDITING problem have been proposed by Böcker et al. [9], while heuristics have recently been described by Bastos et al. [10]. A generalization of the original cluster editing so as to allow overlaps among the clusters has been considered by Damaschke [11] and Fellows et al. [12]. There is also an extensive literature on the related clustering problem, where only edge removals are allowed (e.g. [7]). See [13] for a recent review, with several complexity results.

* Corresponding author.

E-mail addresses: oscarlin@dc.uba.ar (M.C. Lin), francisco.soulignac@unq.edu.ar (F.J. Soulignac), jayme@nce.ufrj.br (J.L. Szwarcfiter).

The extensive literature on FPT algorithms for solving the CLUSTER EDITING problem apparently does not extend in the same degree to polynomial time solvable cases. One of these polynomial time cases is that of proper interval graphs. Manna [14] described an algorithm of time and space complexity $O(n^2)$ for the latter class.

The reason why proper interval graphs are of interest for the CLUSTER EDITING problem is because each vertex of a proper interval graph can be associated to a point of the real line in such a way that two vertices are adjacent if and only if their corresponding points are at distance at most one (see [15]). So, similarity between vertices is expressed by the distance of their corresponding points.

In the present paper, we describe a variation of Manna’s dynamic programming algorithm, which reduces its complexity to $O(n + m)$ time and $O(n)$ space.

A cluster graph C is a graph whose components C_1, \dots, C_k are cliques. Every cluster graph C is uniquely represented by the family $\mathbb{C} = \{C_1, \dots, C_k\}$. A clustering of a graph G is simply a partition of its vertex set; each member of a clustering \mathbb{C} is a cluster of \mathbb{C} . The value of \mathbb{C} is the number $\text{val}(\mathbb{C})$ of edits required to transform G into the cluster graph represented by \mathbb{C} . The optimal value for G is $\text{opt}(G) = \min\{\text{val}(\mathbb{C}) \mid \mathbb{C} \text{ is a clustering of } G\}$. Those clusterings of G with value $\text{opt}(G)$ are referred to as optimal. With this terminology, CLUSTER EDITING is the problem of finding an optimal clustering of G .

We use a convenient representation of proper interval graphs (see Fig. 1). A rightmost function (of order n) is a nondecreasing function $r: \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that $r(i) \geq i$ for every $1 \leq i \leq n$; note that $r(n) = n$. The leftmost function associated to r is the nondecreasing function $\ell: \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that $\ell(i) = \min\{j \mid i \leq r(j)\}$ for every $1 \leq i \leq n$; note that $\ell(1) = 1$ and $\ell(i) \leq i$ for every $1 < i \leq n$. Each rightmost function r defines a graph $G(r)$ with vertex set $\{v_1, \dots, v_n\}$ where v_i is adjacent to $v_j \neq v_i$ if and only if $\ell(i) \leq j \leq r(i)$. A graph G is a proper interval (PIG) graph when it is isomorphic to $G(r)$ for some rightmost function r . In such a case, we say that G admits (ℓ, r) , whereas (ℓ, r) is a straight representation of G . For the sake of notation, we assume the vertices of a PIG graph are always v_1, \dots, v_n . It is not hard to see that a graph is PIG if and only if it is the intersection graph of a family of inclusion-free intervals on the real line (see Fig. 1 or [16]). In fact, the recognition algorithm by Deng et al. [16] outputs a straight representation for any input PIG graph.

Let G be a graph with a straight representation (ℓ, r) . We write $d(j) = j - \ell(j)$ for every $1 \leq j \leq n$, i.e., $d(j)$ is the number of neighbors of v_j in v_1, \dots, v_{j-1} . For $0 \leq i \leq j \leq n$, we denote by $G_{i,j}$ the subgraph of G induced by v_{i+1}, \dots, v_j , while $m_{i,j}$ is the number of edges of $G_{i,j}$. (We exclude v_i from $G_{i,j}$ to simplify index manipulation below.) For the special case $i = 0$, we write $G_j = G_{0,j}$ and $m_j = m_{0,j}$, so $m = m_n$. Note that G_0 is an empty graph that has no vertices and $m_0 = 0$ edges; we need G_0 in order to deal with the base case of the recurrence relation defining the dynamic programming algorithm.

2. The algorithm

In this section we improve Manna’s algorithm by reducing its time and space complexity. Both the algorithm by Manna and our implementation are based on the following theorem, which has been originally conjectured by Damaschke, see [14].

Theorem 1. (See [14].) Every PIG graph admits an optimal clustering in which:

$$\begin{aligned} & \text{each cluster consists of } v_{i+1}, \dots, v_j \\ & \text{for some } 0 \leq i < j \leq n. \end{aligned} \tag{cons}$$

Remark 1. A word of caution is required here. As defined in Section 1, the ordering v_1, \dots, v_n depends on which straight representation (ℓ, r) of G is taken. The article by Manna, however, is described in terms of some restricted PIG models. Every PIG model defines a so-called PIG ordering of the vertices of G . The original version of Theorem 1 [14] holds only for those PIG orders defined by unitary PIG models. However, Roberts [15] proved that every connected PIG graph admits at most two PIG orderings, one the reverse of the other, and it is a well known fact that v_1, \dots, v_n is a PIG ordering of G (e.g. [16]). Thus, Theorem 1 holds for v_1, \dots, v_n as well. \square

For $0 \leq i \leq j \leq n$, let:

- $\bar{m}_{i,j} = \binom{j-i}{2} - m_{i,j}$ be the number of non-edges of $G_{i,j}$, i.e., the number of edges that must be inserted to transform $G_{i,j}$ into a clique, and
- $\text{cut}_{i,j}$ be the number of edges of G joining a vertex in $\{v_1, \dots, v_i\}$ with a vertex in $\{v_{i+1}, \dots, v_j\}$, i.e., the number of edges that must be removed so as to disconnect $G_{i,j}$ from G_i .

We can compute $\text{opt}(G) = \text{opt}(G_n)$ with the recurrence relation of the next theorem.

Theorem 2. For every $0 \leq j \leq n$

$$\begin{aligned} \text{opt}(G_j) &= \begin{cases} 0 & \text{if } j \leq 1 \\ \min_{j-2d(j)-1 \leq i < j} \{\text{opt}(G_i) + \bar{m}_{i,j} + \text{cut}_{i,j}\} & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

Proof. The theorem is true for $j = 1$; suppose $j > 1$ and let:

- \mathbb{C}_j be an optimal clustering for G_j satisfying (cons), and
- $C = \{v_{i+1}, \dots, v_j\}$ be the cluster of \mathbb{C}_j that contains v_j .

Consider the clustering \mathbb{C} obtained from \mathbb{C}_j by splitting C into $C \setminus \{v_j\}$ and $\{v_j\}$. (If $j = i + 1$, then $\mathbb{C} = \mathbb{C}_j$.) By definition, v_j is adjacent to $\min\{d(j), j - i - 1\}$ vertices of

Download English Version:

<https://daneshyari.com/en/article/427229>

Download Persian Version:

<https://daneshyari.com/article/427229>

[Daneshyari.com](https://daneshyari.com)