Theoretical Computer Science 531 (2014) 66–76

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A revisit of the scheme for computing treewidth and minimum fill-in

Masanobu Furuse, Koichi Yamazaki *,[1]

Gunma University, 1-5-1 Tenjin-cho, Kiryu, Gunma, 376-8515, Japan

ARTICLE INFO

Article history:
Received 17 July 2013
Received in revised form 7 November 2013
Accepted 4 March 2014
Communicated by P. Widmayer

Keywords:
Minimal separator
Potential maximal clique
Treewidth
Minimum fill-in

ABSTRACT

In this paper, we reformulate the scheme introduced by Bouchitté and Todinca in [1], which computes treewidth and minimum fill-in of a graph using a dynamic programming approach. We will call the scheme *BT scheme*. Although BT scheme was originally designed for computing treewidth and minimum fill-in, it can be used for computing other graph parameters defined in terms of minimal triangulation. In this paper, we reformulate BT scheme so that it works for computing other graph parameters defined in terms of minimal triangulation, and give examples of other graph parameters.

1. Introduction

In [1,2], Bouchitté and Todinca introduced a dynamic programming approach for computing treewidth and minimum fill-in of a graph, and they showed that, using the dynamic programming approach, treewidth and minimum fill-in can be computed in polynomial time in the number of minimal separators. We will call the dynamic programming approach the *BT scheme*. Although BT scheme was originally designed for computing treewidth and minimum fill-in, it can be used for computing other graph parameters defined in terms of minimal triangulation. (Note that computing treewidth and minimum fill-in both can be translated into problems on minimal triangulation.) Indeed, several variants of BT scheme have been developed to compute other graph parameters/problems parameter by parameter: *tree-length* [3] via *chordal sandwich problem*, *treecost* [4], and the perfect phylogeny problem [5]. To unify those variants, we reformulate BT scheme so that it works for computing other graph parameters defined in terms of minimal triangulation.

The importance of establishment of BT scheme is that it unifies the polynomial computability of treewidth and minimum fill-in for the several graph classes: circle graphs [6,7], circular-arc graphs [8,7], cographs [9], chordal bipartite graphs [10,11], weakly chordal graphs [12], and d-trapezoid graphs [13]. Those graph classes have a polynomial number of minimal separators. In fact, it was conjectured that treewidth and minimum fill-in are computable in polynomial time for the classes of graphs with a polynomial number of minimal separators [14,15], and Bouchitté and Todinca [1,2] proved that the conjecture holds.

BT scheme is based on two types of recursive formulas: one is on minimal separators in [16]:

* Corresponding author.
E-mail addresses: furuse@comp.cs.gunma-u.ac.jp (M. Furuse), koichi@cs.gunma-u.ac.jp (K. Yamazaki).

[1] An earlier version of this research was presented at the 7th Japan Conference on Computational Geometry and Graphs (JCCGG 2009) 2009.

http://dx.doi.org/10.1016/j.tcs.2014.03.013

$$tw(G) = \min_{S \in \Delta_G} \max_{C \in \mathcal{C}(S)} tw\big(R(S,C)\big),$$

$$mfi(G) = \min_{S \in \Delta_G} \Big(fill(S) + \sum mfi\big(R(S,C)\big)\Big),$$

and the other is on *potential maximal cliques* in [1]:

$$tw\big(R(S,C)\big) = \min_{S \subset \Omega \subseteq (S,C)} \max\big(|\Omega| - 1, tw\big(R(S_i, C_i)\big)\big),$$

$$mfi\big(R(S,C)\big) = \min_{S \subset \Omega \subseteq (S,C)} \Big(fill(\Omega) - fill(S) + \sum mfi\big(R(S_i, C_i)\big)\Big),$$

where S and Ω mean a minimal separator and a potential maximal clique, respectively. (See Section 3 for details.) To modify BT scheme so as to be able to compute not only treewidth and minimum fill-in but also other graph parameters defined in terms of minimal triangulation, we reformulate the recursive formulas in Section 3.

It is known that treewidth (tw), minimum fill-in (mfi), and chordal sandwich problem between G_1 and G_2 ($csp(G_1, G_2)$) can be expressed as follows (see Section 2 for the notation):

- $tw(G) = \min_{H \in MT(G)} \max_{M \in MC(H)} |M| - 1$,
- $mfi(G) = \min_{H \in MT(G)} \sum_{e \in FE_G(H)} 1$,
- $csp(G_1, G_2) = \min_{H \in MT(G_1)} \sum_{e \in FE_G(H)} g(e)$, where $g(e) = \begin{cases} 0 & \text{if } e \in E(G_2) \\ 1 & \text{otherwise.} \end{cases}$

As we will show in Section 6, tree-length (tl) can be represented as

- $tl(G) = \min_{H \in MT(G)} \max_{M \in MC(H)} dist_G(M)$.

To unify those expressions, we consider two types of graph parameters, one is *clique type*: graph parameters expressed as $\min_{H \in MT(G)} \max_{M \in MC(H)} f(M)$, and the other is *fill-in type*: graph parameters expressed as $\min_{H \in MT(G)} \sum_{e \in FE_G(H)} f(e)$. The former corresponds to treewidth and the latter to minimum fill-in. Then, we show that BT scheme works for the graph parameters of both clique and fill-in types.

2. Definitions and fundamental results

Let G be a graph and U be a subset of $V(G)$.

notation For a vertex v in G, $N(v)$ denotes the neighbor set of v, and $N(U)$ denotes the set $\bigcup_{u \in U} N(u) - U$. $G[U]$ denotes the subgraph of G induced by U. We denote by $\mathcal{C}_G(U)$ the set of connected components of $G[V \setminus U]$, and by G_U the graph obtained from G by completing U, i.e., by adding an edge between every pair of non-adjacent vertices of U. For convenience, for a connected component $C \in \mathcal{C}_G(U)$, we often make no distinction between the component C and its vertex set $V(C)$, so C be used in the sense of $V(C)$. We will drop the subscript G when it is clear from the context. For example, we will write simply $\mathcal{C}(U)$ instead of $\mathcal{C}_G(U)$. $MC(G)$ denotes the set of maximal cliques of G. For $x, y \in V(G)$, $dist_G(x, y)$ denotes the distance between u and v in G. We denote by $fill_G(U)$ the number of non-edges of U in G.

component A component $C \in \mathcal{C}_G(U)$ is a *full component associated with* U if for each vertex $u \in U$ there is a vertex in $v \in C$ such that $\{u, v\} \in E(G)$.

separator ([2]) A subset $S \subseteq V(G)$ is an *a*, *b-separator* of G for two non-adjacent vertices $a, b \in V(G)$ if the removal of S from G separates a and b in different connected components. An a, b-separator S is *minimal* if no proper subset of S separates a and b. S is a *minimal separator* of G if there are two vertices a and b for which S is a minimal a, b-separator. We denote by Δ_G the set of all minimal separators of G.

triangulation ([2]) A graph is chordal if every cycle of length at least four has a *chord* (i.e. an edge joining two vertices that are not adjacent in the cycle). A *triangulation* of $G = (V, E)$ is a chordal graph $H = (V, E \cup F)$ such that $E \cap F = \emptyset$, and F is called the *fill-in edges* of H. We denote F by $FE_G(H)$. H is a *minimal* triangulation of G if no proper subgraph of H is a triangulation of G. $MT(G)$ denotes the set of minimal triangulations of G. It is known that $\Delta_H \subseteq \Delta_G$ (see e.g. Theorem 2.9 in [1]).

potential maximal clique ([2]) A vertex set Ω of G is called a *potential maximal clique* if there is a minimal triangulation H of G such that Ω is a maximal clique of H. We denote by Π_G the set of all potential maximal cliques of G. For convenience, we stretch $MT(\cdot)$ slightly as follows: for a potential maximal clique Ω in G, $MT(G, \Omega)$ denotes the set $\{H \mid H \in MT(G) \text{ and } \Omega \in MC(H)\}$.

block Let S be a minimal separator of G. For $C \in \mathcal{C}(S)$, we say that $(S, C) = S \cup C$ is a *block associated with* S (or simply *block* of S). A block (S, C) is a *full* if C is a full component associated with S. The graph $R(S, C)$ obtained from