

Fully Dynamically Maintaining Minimal Integral Separator for Threshold and Difference Graphs ^{*}

Tiziana Calamoneri, Angelo Monti, and Rossella Petreschi

Computer Science Department,
“Sapienza” University of Rome, Italy
`calamo/monti/petreschi@di.uniroma1.it`

Abstract. This paper deals with the well known classes of threshold and difference graphs, both characterized by separators, i.e. node weight functions and thresholds. We show how to maintain minimum the value of the separator when the input (threshold or difference) graph is fully dynamic, i.e. edges/nodes are inserted/removed. Moreover, exploiting the data structure used for maintaining the minimality of the separator, we handle the operations of disjoint union and join of two threshold graphs.

Keywords: fully dynamic graphs, threshold graphs, difference graphs, chain graphs, threshold signed graphs, graph operations.

1 Introduction

In many applications of graph algorithms, *graphs* are *fully dynamic*, i.e. both edges and nodes may be inserted or eliminated.

Typically, one would like to answer to a precise query on the fully dynamic graph, so the goal is to update the data structure after dynamic changes, rather than having to recompute it from scratch each time.

Threshold graphs constitute a very important and well studied graph class, since they find applications in several fields, such as psychology, parallel processing, scheduling, and graph theory. For this reason, threshold graphs have been defined many times in the literature (see, e.g. [2, 5]), and have been widely studied. *Difference graphs* (also known as *chain graphs*)—that are strictly related to threshold graphs, though incomparable—had similar destiny and have been independently introduced (see, e.g. [3, 9]). For a comprehensive survey on threshold graphs, difference graphs and related topics, see [6].

Among the numerous equivalent definitions of threshold and difference graphs, many of them exploit a node weight function and a threshold. This pair is called a *separator* and of course it is not unique. It is of interest to determine a *minimum separator*, i.e. a separator with minimum value of the threshold. Orlin [7] presented an algorithm for minimizing the threshold w.r.t. one of these definitions in linear time in the number of nodes.

^{*} Partially supported by the Italian Ministry of Education and University, PRIN project “AMANDA: Algorithmics for MAssive and Networked DATA”

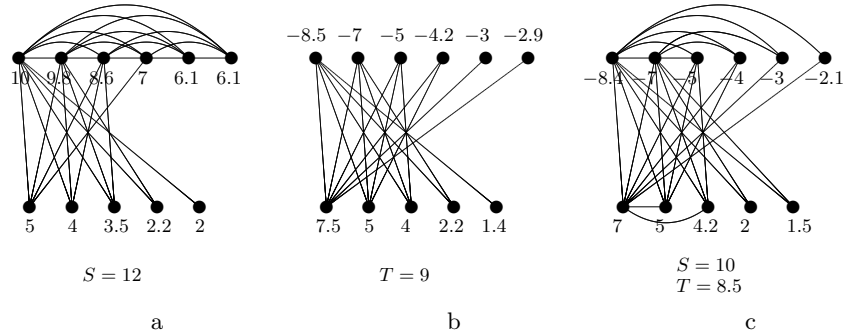


Fig. 1. a. A threshold graph; b. A difference graph; c. A threshold signed graph.

In this paper we consider a different (equivalent) definition, also based on a threshold and a node weight function. After a pre-computation, we have always available the minimum separator after fully dynamically changing the graph. Both the pre-computation and each linear time operation of addition/deletion of either an edge or a node with all its incident edges are performed in linear time w.r.t. the number of different degrees of the current graph.

So, this is a contribution to the problem of the dynamic maintenance of threshold and difference graphs. To the best of our knowledge, few works deal with this topic. Namely, in [8] the problem of dynamically recognizing some classes of graphs (and among them threshold graphs) is handled. In [4] the authors consider the problem of adding/deleting edges with the aim of transforming a given graph into a threshold graph with the minimum number of changes.

We conclude this paper with a section that, exploiting the data structure used for maintaining the minimality of the separator, handles the operations of disjoint union and join of two threshold graphs.

2 Preliminaries

In this section, we list some definitions and properties, all from [6]. For the sake of clarity, we reorganized them in order to optimize the presentation.

Definition 1. A graph $G = (V, E)$ is a threshold graph if there is a mapping $a : V \rightarrow \mathbb{R}^+$ and a positive real number S such that

$$a(v) < S \text{ for all } v \in V \quad (1)$$

$$\{v, w\} \in E \text{ if and only if } a(v) + a(w) \geq S \quad (2)$$

The pair (a, S) will be called separator for graph G .

In Figure 1.a a threshold graph with one of its separators is depicted.

Definition 2. A graph $G = (V, E)$ is a difference graph if there is mapping $a : V \rightarrow \mathbb{R}$ and a positive real number T such that

$$|a(v)| < T \text{ for all } v \in V \quad (3)$$

$$\{v, w\} \in E \text{ if and only if } |a(v) - a(w)| \geq T \quad (4)$$

The pair (a, T) will be called separator for graph G .

The node set of a difference graph G can be partitioned as $V = U \cup W$, where $U = \{v \in V : a(v) \geq 0\}$ and $W = \{v \in V : a(v) < 0\}$; both U and W induce a stable set and hence G is bipartite with bipartition (U, W) . A difference graph with one of its separators is shown in Figure 1.b.

Although Definitions 1 and 2 are very similar, the two defined classes are incomparable. Nevertheless they are strictly related, as shown by the following theorem:

Theorem 1. *A bipartite graph $G = (U \cup W, E)$ is a difference graph if and only if adding to G all possible edges with both ends in the same side of the bipartition (either U or W) yields a threshold graph.*

Given a graph $G = (V, E)$, we denote by $\deg(v)$ the degree of node v .

Definition 3. *Let $G = (V, E)$ be a graph whose distinct node-degrees are $\delta_1 < \dots < \delta_m$, and let $\delta_0 = 0$ (even if no node of degree 0 exists). Let $D_i = \{v \in V \text{ s.t. } \deg(v) = \delta_i\}$ for $i = 0, \dots, m$; D_i is called i -th box; the sequence D_0, \dots, D_m is called the degree partition of G .*

Definition 4. *Let $G = (U \cup W, E)$ be a bipartite graph and let X be either U or W . The distinct node-degrees for the nodes in X are $\delta_1^X < \dots < \delta_{t_X}^X$ and let $\delta_0^X = 0$ (even if no node of degree 0 exists in the partition X). Let $D_i^X = \{v \in X \text{ s.t. } \deg(v) = \delta_i^X\}$ for $0 \leq i \leq t_X$; the sequence $D_0^U, \dots, D_{t_U}^U, D_0^W, \dots, D_{t_W}^W$ is called the bipartite degree partition of G .*

Lemma 1. *Let $G = (V, E)$ be a threshold graph with degree partition D_0, \dots, D_m and whose node set is partitioned into a clique K and a stable set I ; let $x \in D_i$ and $y \in D_j$ be two distinct nodes.*

1. $D_0 \cup \dots \cup D_{\lfloor m/2 \rfloor} = I$ and $D_{\lfloor m/2 \rfloor + 1} \cup \dots \cup D_m = K$;
2. If $e = \{x, y\} \notin E$, the graph $G' = (V, E \cup \{e\})$ is a threshold graph if and only if $i + j = m$;
3. If $e = \{x, y\} \in E$, the graph $G' = (V, E \setminus \{e\})$ is a threshold graph if and only if $i + j = m + 1$;
4. $e = \{x, y\} \in E$ if and only if $i + j \geq m + 1$.

Lemma 2. *Let $G = (U \cup W, E)$ be a difference graph with degree partition $D_0^U, \dots, D_{t_U}^U, D_0^W, \dots, D_{t_W}^W$, and let $x \in D_i^U$ and $y \in D_j^W$ be two distinct nodes.*

1. $t_U = t_W = t$;
2. If $e = \{x, y\} \notin E$, the graph $G' = (U \cup W, E \cup \{e\})$ is a difference graph if and only if $i + j = t$;
3. If $e = \{x, y\} \in E$, the graph $G' = (U \cup W, E \setminus \{e\})$ is a difference graph if and only if $i + j = t + 1$;
4. $e = \{x, y\} \in E$ if and only if $i + j \geq t + 1$.

Given a graph G , we recall that the *neighborhood* $N(v)$ of a node v is the set of all neighbors of v , and its *closed neighborhood* is $N[v] = N(v) \cup \{v\}$.

Lemma 3. *Given a graph $G = (V, E)$, the vicinal preorder is a binary relation on the nodes of V such that $u \succeq v \Leftrightarrow N[u] \supseteq N[v]$. The vicinal preorder is total on V if G is a threshold graph G and on U and V if G is a difference graph.*

We now introduce a superclass of both threshold and difference graphs.

Definition 5. [1] *A graph $G = (V, E)$ is a threshold signed graph if there is a mapping $a : V \rightarrow \mathbb{R}$ and two positive real numbers S and T such that*

$$|a(v)| < \min\{S, T\} \tag{5}$$

$$\{v, w\} \in E \text{ iff either } |a(v) + a(w)| \geq S \text{ or } |a(v) - a(w)| \geq T. \tag{6}$$

The triple (a, S, T) will be called *separator* for graph G .

Consider $X = \{x \in V \text{ s.t. } a(x) < 0\}$ and $Y = \{x \in V \text{ s.t. } a(x) \geq 0\}$.

As highlighted in Figure 1.c, we can see a threshold signed graph as constituted by two threshold graphs, G^- and G^+ respectively induced by X and Y , that are connected by a difference graph D . Notice that for X we consider the opposite of the a 's values.

3 A data structure for computing minimal integral separator for threshold or difference graphs

Although in Definition 1 a threshold graph $G = (V, E)$ is a graph having a separator with non-negative *real* values, it is common to equivalently require the separator to have non-negative *integral* values (i.e. an integral separator) [6]. We say that an integral separator (a, S) for G is *minimum* if for any other integral separator (a', S') for G we have $S \leq S'$. In the following theorem we show that the value of S of a minimum integral separator (a, S) of G is given by the cardinality of the degree partition of G plus 1.

Theorem 2. *Let $G = (V, E)$ be a threshold graph with degree partition D_0, \dots, D_m . The pair (a, S) , where $S = m + 1$ and for each node $v \in V$, $a(v) = i$ if $v \in D_i$, is a minimal integral separator of G .*

Proof. First of all, we prove that (a, S) is a separator, i.e. that it satisfies the two inequalities of Definition 1. Note that for each $v \in V$ it holds $0 \leq a(v) \leq m < S$ thus the pair (a, S) satisfies Inequality 1. Moreover, Inequality 2 follows from Item 4 in Lemma 1. Trivially, (a, S) is integral.

Let us now prove that (a, S) is minimal. By contradiction, let (a, S) be not minimal, and let (a', S') be an integral separator for G such that $S' < S$. Observe that only isolated nodes can have weight equal to zero (indeed, if $\{u, v\} \in E$ and $a'(u) = 0$ then $a'(u) + a'(v) = a'(v) < S'$ from Inequality 1, but this contradicts Inequality 2). Moreover, notice that two nodes u and v having the same weight

necessarily behave in the same way (i.e. for any other node $w \in V$, it holds that $\{u, w\} \in E$ if and only if $\{v, w\} \in E$), so nodes having different degree cannot have the same weight. All this implies that the function a' on the non isolated nodes assumes at least m different strictly positive weights. Thus Inequality 1 implies that $S' \geq m+1$. The chain $m+1 = S > S' \geq m+1$ proves the minimality of (a, S) . \square

From the same reasonings, we deduce the following theorem for a difference graph G .

Theorem 3. *Let $G = (U \cup W, E)$ be a difference graph with bipartite degree partition $D_1^U, \dots, D_t^U, D_1^W, \dots, D_t^W$. The pair (a, S) , where $S = 2t + 1$ and, for each node $v \in U$, $a(v) = -i$ if $v \in D_i^U$ and $a(v) = i$ if $v \in D_i^W$, is a minimal integral separator of G .*

Notice that Orlin [7] shows how to minimize the threshold for threshold graphs considering an equivalent definition requiring that the sum of the weights of the nodes of any independent set has to be smaller than the threshold. The value of this threshold is larger than the value of S computed in this paper.

Now we present two data structures for representing threshold and difference graphs allowing us to compute in a natural way minimal integral separators for these graph classes, according to Theorems 2 and 3. Since a threshold graph $G = (V, E)$ is univocally determined by its degree partition D_0, \dots, D_m , we may store G using two arrays $\delta[0..m]$ and $\mu[0..m]$, where $\delta[i]$ represents the degree δ_i of the nodes in the i -th box D_i , and $\mu[i]$ represents its cardinality, $|D_i|$. Similarly, let $G = (U \cup W, E)$ be a difference graph with bipartite degree partition $D_1^U, \dots, D_t^U, D_1^W, \dots, D_t^W$. This partition univocally determines G , and G may be represented using the arrays $\delta_X[0..t]$ and $\mu_X[0..t]$, where $\delta_X[i]$ represents the degree of the nodes in the i -th box D_i^X , and $\mu_X[i]$ represents its cardinality $|D_i^X|$, $X \in \{U, W\}$. If $G = (V, E)$ is a threshold graph, i is the weight associated to all the $\mu[i]$ nodes belonging to the box D_i of degree δ_i . Similarly, if $G = (U \cup W, E)$ is a difference graph, i is the weight associated to all the $\mu_X[i]$ nodes belonging to box D_i^X of degree $\delta_X[i]$, with $X \in \{U, W\}$.

Exploiting these two data structures, the following theorem holds:

Theorem 4. *Given a threshold (difference) graph G by means of its (bipartite) degree partition, its minimal integral separator can be found in time linear w.r.t. the number of different degrees in G .*

In Sections 4 and 5, each time we speak about a graph G (either threshold or difference), G is represented by means of the arrays δ and μ .

4 Adding/deleting an edge to threshold/difference graphs

In this section we study how to get a new graph, obtained by adding/deleting an edge from a graph that is either a threshold or a difference graph, and to

keep immediately available the knowledge of the minimum separator for the new graph. In order to make easier the exposition, preliminarily we consider two functions, operating on the data structures introduced in Section 3.

By **IncreaseDeg** (δ, μ, i, dim) we denote the operation of updating arrays $\delta[0..dim]$ and $\mu[0..dim]$ when the degree of a node in box D_i , $0 \leq i \leq dim$, is increased by one. **IncreaseDeg** can have as consequence the appearance of a new box (if $i = dim$ or if the degree of the nodes in D_{i+1} is different from the degree of nodes in D_i plus one). On the other hand, this increment can also have as consequence the disappearance of box D_i (if $i \neq 0$, $|D_i| = 1$ and the degree of the nodes in D_{i+1} is equal to the degree of the nodes in D_i plus one). Symmetrically, we may consider the operation **DecreaseDeg** (δ, μ, i, dim) of updating arrays $\delta[0..dim]$ and $\mu[0..dim]$ when the degree of a node in box D_i , $0 < i \leq dim$, is decreased by one.

Let us now consider a threshold graph G . Let $\{x, y\}$, $x \in D_i$ and $y \in D_j$, the edge to add/delete to/from G . Items 2 and 3 of Lemma 1 give a characterization of the indices i and j to ensure that the modified graph is still a threshold graph; namely, $i + j = m$ in case of insertion, and $i + j = m + 1$ in case of deletion.

We present two operations, **InsEdge** (δ, μ, i, m) and **DelEdge** (δ, μ, i, m) , that update the data structure when an edge is added between a node in box D_i and a node in box D_{m-i} and when an edge is deleted between a node in box D_i and a node in box D_{m+1-i} , respectively. Observe that with the insertion of an edge, the degrees of its endpoints are increased by one. Thus we can call twice subroutine **IncreaseDeg**, once on a node in box D_i and once on a node in box D_{m-i} . We have just to take into account that the increment of the degree of node in box D_i can change the index of the box of the other endpoint. Analogous considerations hold for the deletion of an edge. These observations give rise to the following simple algorithms:

InsEdge (δ, μ, i, m) $j \leftarrow m - i;$ $a \leftarrow m;$ IncreaseDeg $(\delta, \mu, i, m);$ CASE($m - a$) -1 : IncreaseDeg $(\delta, \mu, j - 1, m);$ 0 : IncreaseDeg $(\delta, \mu, j, m);$ +1 : IncreaseDeg $(\delta, \mu, j + 1, m);$	DelEdge (δ, μ, i, m) $j \leftarrow m + 1 - i;$ $a \leftarrow m;$ DecreaseDeg $(\delta, \mu, i, m);$ CASE($m - a$) -1 : DecreaseDeg $(\delta, \mu, j - 1, m);$ 0 : DecreaseDeg $(\delta, \mu, j, m);$ +1 : DecreaseDeg $(\delta, \mu, j + 1, m);$
---	---

Since after the execution of **IncreaseDeg** (**DecreaseDeg**), m may potentially vary from m to $m \pm 1$, with the execution of **InsEdge** (**DelEdge**) the number of boxes can potentially vary from m to $m \pm 2$. In Figure 2 we show that all the five possibilities may actually occur.

Assume now that G is a difference graph. The algorithms for adding/ eliminating an edge in G are based on the same idea presented for the algorithms on threshold graphs, but they are even simpler because the data structure used for representing these graphs keeps separated the bipartition (and so, adding a new box after the first call of **IncreaseDeg** does not affect the index of the other

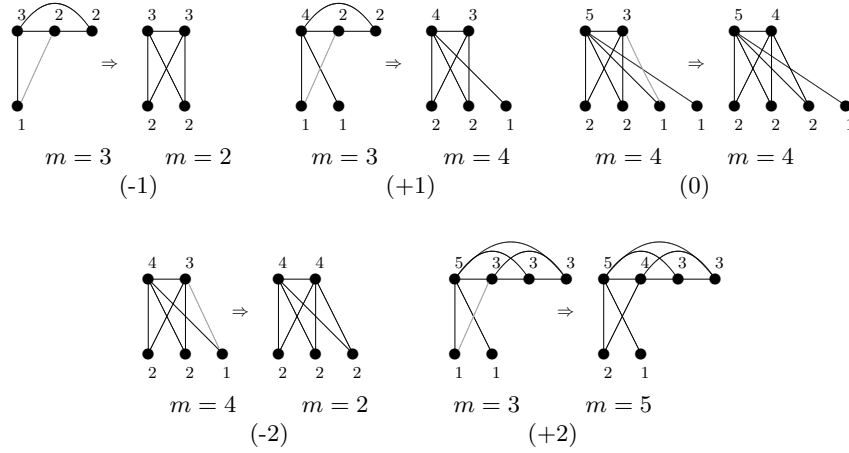


Fig. 2. Examples proving that all 5 cases in algorithm **InsEdge** are possible. Grey edges represent the edges that are going to be added. (In order to consider **DelEdge**, figures must be read from right to left.)

endpoint). Notice that Item 1 of Lemma 2 ensures that the number of boxes in the two classes is the same t . So, for difference graphs, t can either remain unaltered or to change to $t \pm 1$ and all the three possibilities may actually occur. The two algorithms for inserting and deleting an edge in a difference graph follow:

<p>D – InsEdge($\delta_U, \mu_U, \delta_W, \mu_W, i, t$) $j \leftarrow t - i$; IncreaseDeg(δ_U, μ_U, i, t); IncreaseDeg(δ_W, μ_W, j, t);</p>	<p>D – DelEdge($\delta_U, \mu_U, \delta_W, \mu_W, i, t$) $j \leftarrow t + 1 - i$; DecreaseDeg(δ_U, μ_U, i, t); DecreaseDeg(δ_W, μ_W, j, t);</p>
--	--

Note. All the algorithms described in this section are correct and maintain the minimality of the integral separators.

5 Adding/deleting a node to threshold/difference graphs

In this section we will work with nodes in an analogous way as we did in Section 4 with edges. Also in this case, we keep immediately available the knowledge of the minimum separator for the new graph. We start defining four functions, operating on the data structures introduced in Section 3.

By **+Node**(δ, μ, d, dim) we denote the operation of giving space to a new node of degree d either in a threshold graph or in a partition of a difference graph, without caring about the update of its neighbors (that will be done with another subroutine). This subroutine looks for the box where the new node must be inserted: if there exists a box D_i with degree d , μ_i is simply increased by one; otherwise a new box for the new node is created.

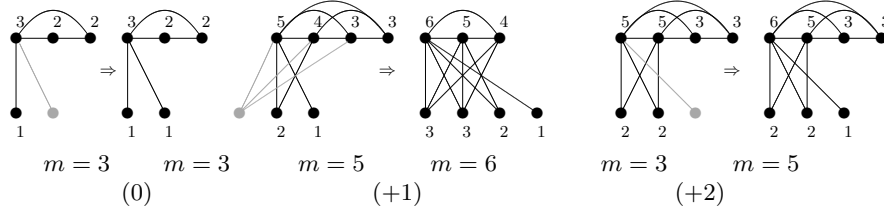


Fig. 3. Examples proving that all 3 cases in algorithm **InsNode** are possible. Grey nodes and edges represent the objects that are going to be added (in order to consider **DelNode** figure must be read from right to left).

By **IncreaseDegOfSetNode** (δ, μ, d, dim) we denote the operation of augmenting by one the degree of the d nodes of highest degree either in a threshold graph or in a partition of a difference graph. This subroutine increases by one the degree of all the boxes D_i s.t. $d - \sum_{s=j+1}^m |D_s| \geq 0$, while nodes of boxes D_1, \dots, D_{j-1} remain unchanged. For what concerns D_j , it is in general split into two boxes (precisely $d - \sum_{s=j+1}^m |D_s|$ nodes leave D_j to form a new box with degree augmented by one). We can define even the symmetric functions: by **-Node** (δ, μ, i, dim) we denote the operation eliminating from the data structure storing either a threshold or a difference graph a node in box D_i , $0 \leq i \leq dim$, regardless of its neighbors (whose degree will be updated with another subroutine). By **DecreaseDegOfSetNode** (δ, μ, d, dim) we denote the operation of decreasing by one the degree of the d nodes of highest degree either in a threshold graph or in a partition of a difference graph.

Let now $G = (V, E)$ be a threshold graph. Adding a new node of degree d to G yields a threshold graph if and only if the d neighbors of the new node are the d nodes with highest degrees (this can be easily deduced from Item 4 of Lemma 1). So, we can call **IncreaseDegOfSetNode** and observe that, after its execution, m could be increased by one. Then, we have to update the data structure by inserting the new node by means of **+Node**, and even in this case m could be increased by one. So, the number of the different degrees can potentially vary from m to $m + 2$. Figure 3 shows that all three possibilities can occur.

By **InsNode** (δ, μ, d, m) we denote the operation of updating the data structure storing threshold graph G when a node of degree d is added to the graph. The previous reasonings can be repeated when G is a difference graph (assuming, w.l.o.g., that the new node is inserted in partition U), so giving rise to **D-InsNode** $(\delta_U, \mu_U, \delta_W, \mu_W, i, t)$, that is the operation of updating the data structure when a node of degree d is added to the difference graph.

InsNode (δ, μ, d, m) +Node (δ, μ, d, m) IncreaseDegOfSetNode (δ, μ, d, m)	D-InsNode $(\delta_U, \mu_U, \delta_W, \mu_W, d, t)$ +Node (δ_U, μ_U, d, t) IncreaseDegOfSetNode (δ_W, μ_W, d, t)
---	--

Now we consider the problem of deleting nodes to a threshold graph. Any node-induced subgraph G' of G is a threshold graph (indeed for the graph G' use the mapping a restricted to the nodes of G' and the same value S). Thus the class of threshold graphs is closed under the deletion of an arbitrary node. Given a threshold graph, by **DelNode** (δ, μ, i, m) we denote the operation of updating the data structure when a node is deleted from box D_i , $0 \leq i \leq m$. This deletion is performed by **-Node** that can have as consequence the disappearance of box D_i (if $i \neq 0$ and $|D_i| = 1$). Thus m can decrease by one. Moreover, the $\delta[i]$ nodes with highest degree must have their degrees decreased by one. These nodes belong to boxes D_m, \dots, D_{m+1-i} . It can occur that the degree of nodes in box $m+1-i$ becomes equal to the degree of the nodes in box $m-i$ and, in this case, the two boxes merge and the number of boxes further decrease by one. Hence after deleting a node, the number of boxes in the degree partition can potentially vary from m to $m-2$ and all cases can occur, as shown in Figure 3.

Analogous reasonings can be done when G is a difference graph, and define **D-DelNode** $(\delta_U, \mu_U, \delta_W, \mu_W, i, t)$ as the operation of updating the data structure when a node is deleted (assuming w.l.o.g. that the new node is deleted from partition U).

DelNode (δ, μ, i, m)	D - DelNode $(\delta_U, \mu_U, \delta_W, \mu_W, i, t)$
$d \leftarrow \delta[i]$	$d \leftarrow \delta_U[i]$
-Node (δ, μ, i, m)	-Node (δ_U, μ_U, i, t)
DecreaseDegOfSetNode (δ, μ, d, m)	DecreaseDegOfSetNode (δ_W, μ_W, d, t)

Note. *All the algorithms described in this section are correct and maintain the minimality of the integral separators.*

6 Disjoint union and join of two threshold graphs

Given two graphs with disjoint node sets $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their *disjoint union* is the graph $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$; their *join* is the graph $G_1 + G_2$ obtained adding to their disjoint union all the edges that connect the nodes of the first graph with the nodes of the second graph. We observe that, if G_1 and G_2 are both threshold graphs, $G_1 \cup G_2$ and $G_1 + G_2$ are threshold signed graphs, where the difference graph connecting the two threshold graphs is either the null graph (in $G_1 \cup G_2$) or the complete bipartite graph (in $G_1 + G_2$).

In this section, exploiting the considerations done for the data structures presented in Section 3 for threshold and difference graphs, we introduce a data structure for representing a threshold signed graph. Thanks to this data structure, we handle the dynamic operations of disjoint union and join of two threshold graphs.

Given a threshold signed graph $G = (X \cup Y, E)$, two nodes u and w are *false twins* if they have the same neighborhood, i.e. $N(u) = N(w)$; they are *true twins* if they have the same closed neighborhood, i.e. $N[u] = N[w]$. We say that u and w are simply *twins* if they are either true or false twins and they belong to the same set, X or Y . Let us consider the partition of the node set into equivalence

classes, B_1, \dots, B_Δ , induced by the relation of being twins. Even though there is not a tie between the degree partition of a threshold signed graph and its structure, as in the case of threshold and difference graphs, it is possible to extend the reasonings done in the proof of Theorem 2 to this class of graphs. Indeed, if v is an isolated node it is not restrictive to assume $a(v) = 0$ and, obviously, if $a(v) = 0$ then v is an isolated node. Moreover, it is easy to see that two nodes having the same value of a are necessarily twins. From the other hand, if there are two twins u and w having $a(u) \neq a(w)$ (w.l.o.g. let $a(u) < a(w)$), we can easily modify function a in order to assign them the same value (that is $a(w)$ if u and w are connected and $a(u)$ otherwise). So, from now on, we consider only node weight functions assigning value 0 to each isolated node and the same value to each set of twins.

As consequence of all these reasonings, we may store a threshold signed graphs by means of two arrays $\alpha[0..\Delta]$ and $\mu[0..\Delta]$: in $\alpha[i]$ there is the value of the weight assigned to the $\mu[i]$ nodes of B_i , $0 \leq i \leq \Delta$; if there are no isolated nodes $\alpha[0]$ and $\mu[0]$ are set to 0. Variables S and T store the two thresholds.

Let us now go back to consider the operations of disjoint union and join of two threshold graphs G_1 and G_2 stored as $\delta_1[0..m_1]$, $\mu_1[0..m_1]$ and $\delta_2[0..m_2]$, $\mu_2[0..m_2]$, respectively. Assume first that G_1 and G_2 have the same threshold S (i.e. $m_1 = m_2 = m$). Informally, the array $\alpha[1..\Delta]$ of both $G_1 \cup G_2$ and $G_1 + G_2$ is obtained by opportunely transcribing the values of the node weight function of the single threshold graphs (deduced through Theorem 2), the array $\mu[1..\Delta]$ is obtained by copying the values of μ_1 and μ_2 , while threshold S is kept unaltered. For what concerns threshold T , in the case of $G_1 \cup G_2$ it is set to a sufficiently large value in order to guarantee that no edges are in the difference subgraph, in the case of $G_1 + G_2$ it is set to a sufficiently small value in order to guarantee that the difference subgraph is a complete bipartite graph. In this latter case, T assumes a too small value, contradicting Property 5 of Definition 5, so we need to modify the values of the node weight function and of the two thresholds in order to restore the property.

The following lemmas formalize the operations of disjoint union and join of two threshold graphs:

Lemma 4. *Let be given two threshold graphs G_1 and G_2 by means of $\delta_1[0..m]$, $\mu_1[0..m]$ and $\delta_2[0..m]$, $\mu_2[0..m]$ and let $S = m + 1$ be their common threshold. The following function determines the threshold signed graph $G_1 \cup G_2$:*

```

DisjointUnion( $\delta_1, \mu_1, \delta_2, \mu_2, m$ )
 $\mu[0] \leftarrow \mu_1[0] + \mu_2[0]$ ;  $\alpha[0] \leftarrow 0$ ;
FOR  $i = 1$  TO  $m$  DO
   $\alpha[i] \leftarrow -i$ ;  $\mu[i] \leftarrow \mu_1[i]$ ;
FOR  $i = 1$  TO  $m$  DO
   $\alpha[m + i] \leftarrow i$ ;  $\mu[m + i] \leftarrow \mu_2[i]$ ;
 $\Delta \leftarrow 2m$ ;
 $S \leftarrow m + 1$ ;
 $T \leftarrow 2m + 1$ ;
RETURN ( $\alpha, \mu, S, T, \Delta$ ).

```

Proof. Both S and T determined by function **DisjointUnion** are greater than the modulo of each $\alpha[i]$, $i = 0, \dots, \Delta$ as far as they are defined.

Moreover, the two threshold subgraphs G^- and G^+ of $G_1 \cup G_2$ are exactly the same as G_1 and G_2 , respectively. Finally, no edge can satisfy the condition $\alpha[u] + \alpha[v] \geq T$ in view of the definition of T , so the difference subgraph D is empty. It follows that α , μ , S and T correctly define $G_1 \cup G_2$. \square

Lemma 5. *Let be given two threshold graphs G_1 and G_2 by means of $\delta_1[0..m]$, $\mu_1[0..m]$ and $\delta_2[0..m]$, $\mu_2[0..m]$ and let $S = m + 1$ be their common threshold. The following function determines the threshold signed graph $G_1 + G_2$:*

```

Join( $\delta_1, \mu_1, \delta_2, \mu_2, m$ )
 $\alpha[0] \leftarrow 0$ ;  $\mu[0] \leftarrow 0$ ;
IF  $\mu_1[0] \neq 0$ 
  THEN  $flag1 \leftarrow 0$ 
  ELSE  $flag1 \leftarrow 1$ 
FOR  $i = flag1$  TO  $m$  DO
   $\alpha[i + 1 - flag1] \leftarrow -i$ ;  $\mu[i + 1 - flag1] \leftarrow \mu_1[i]$ ;
IF  $\mu_2[0] \neq 0$ 
  THEN  $flag2 \leftarrow 0$ 
  ELSE  $flag2 \leftarrow 1$ 
FOR  $i = flag2$  TO  $m$  DO
   $\alpha[m + i + 2 - flag1 - flag2] \leftarrow i$ ;  $\mu[m + i + 2 - flag1 - flag2] \leftarrow \mu_2[i]$ ;
 $\Delta \leftarrow 2m + 2 - flag1 - flag2$ ;
 $S \leftarrow m + 1$ ;
 $T \leftarrow \min_{1 \leq i \leq \Delta} \{|\alpha[i]|\}$ ;
 $k \leftarrow m - T + 1$ ;
FOR  $i = 1$  TO  $m + 1 - flag1$  DO
   $\alpha[i] \leftarrow \alpha[i] - k$ ;
FOR  $i = m + 2 - flag1$  TO  $2m + 2 - flag1 - flag2$  DO
   $\alpha[i] \leftarrow \alpha[i] + k$ ;
 $S \leftarrow S + 2k$ ;
 $T \leftarrow T + 2k$ ;
RETURN ( $\alpha, \mu, S, T, \Delta$ ).

```

Proof. Preliminarily, observe that $G_1 + G_2$ cannot have isolated nodes, so we set $\mu[0]$ to 0; moreover, if either G_1 or G_2 contain isolated nodes, a box needs to be added: we do this exploiting the two boolean variables $flag1$ and $flag2$.

So, even in this case, the two threshold graphs G^- and G^+ of $G_1 + G_2$ are exactly the same as G_1 and G_2 , respectively. T is set to the modulo of the smallest node weight in order to guarantee that the difference graph D is a complete bipartite graph. In this way, T results in a value that contradicts Property 5 of Definition 5. By incrementing the modulo of each $\alpha[i]$ of an opportune value k and S and T by $2k$, we are able to restore the inequality. \square

It remains to handle the case in which G_1 and G_2 have different thresholds. In this case, we propose to the functions described in the proofs of Lemmas 4 and 5 a preprocessing phase that equalize their thresholds as detailed in the following lemma, where with the notation $a' = xa + y$ (where x and y are integer values

and a is a node weight function) we compactly mean that, for each node v , $a'(v) = xa(v) + y$. We want to underline that now on we represent a threshold graph in terms of its separator, instead of in terms of our data structure, because the description of the equalization appears more comprehensive.

Lemma 6. *Let be given two thresholds graphs G_1 and G_2 and let (a_1, S_1) and (a_2, S_2) be their integral separators with $S_1 < S_2$. Then $(a'_1, S'_1) = (2a_1 + S_2 - S_1, 2S_2)$ is an integral separator for G_1 and $(a'_1, a'_2) = (2a_2, 2S_2)$ is an integral separator for G_2 .*

Proof. Let $\{v, w\}$ be an edge in G_1 , i.e. $a_1(v) + a_1(w) \geq S_1$; then $a'_1(v) + a'_1(w) = 2a_1(v) + S_2 - S_1 + 2a_1(w) + S_2 - S_1 \geq 2S_2 = S'_1$. In the same way, let v and w be not connected in G_1 , i.e. $a_1(v) + a_1(w) < S_1$; then $a'_1(v) + a'_1(w) = 2a_1 + S_2 - S_1 < 2S_2 = S'_1$. Finally, the pair (a'_1, S'_1) is a feasible integral separator since, for any node v , $a_1(v) < S_1$ implies $a'_1(v) < S'_1$. Analogous reasonings lead to prove that (a'_2, S'_2) is an integral separator for G_2 . \square

Notice that the values of the node weight function and of the two thresholds S and T of the resulting threshold signed graphs will be integral but not necessarily minimal.

References

1. C. Benzaken, P. L. Hammer and D. de Werra, Threshold characterization of graphs with Dilworth number two. *J. Graph Theory* 9, 245–267, 1985.
2. V. Chvátal and P. L. Hammer. Aggregation of inequalities in integer programming, *Annals of Discrete Mathematics* 1, 145–162, 1977.
3. O. C. Ferrers digraphs and threshold graphs. *Discrete Mathematics* 38, 33–46, 1982.
4. P. Heggernes and C. Papadopoulos. Single-edge monitor sequences of graphs and linear-time algorithms for minimal completions and deletions. *Theoretical Computer Science* 410, 1–15, 2009.
5. P. B. Henderson and Y. Zalcstein. A graph-theoretic characterization of the PV_{chunk} class of synchronizing primitives. *SIAM Journal of Computing* 6, 88–108, 1977.
6. N. V. R. Mahadev and U. N. Peled. Threshold Graphs and Related Topics, *Annals of Discrete Mathematics* 56, North-Holland, Amsterdam, 1995.
7. J. Orlin. The minimal integral separator of a threshold graph. *Annals of Discrete Mathematics* 1, 415–419, 1977.
8. R. Shamir and R. Sharan. A fully dynamic algorithm for modular decomposition and recognition of cographs. *Discrete applied Mathematics* 136, 329–340, 2004.
9. M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on algebraic and Discrete Methods* 3, 351–358, 1982.