STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LX**, Number 2, 2015

ALGORITHMIC APPROACH IN REORIENTATION OF COMPARABILITY GRAPHS

SERGIU CATARANCIUC AND NICOLAE GRIGORIU

Abstract. In this article we present methods and algorithms for arcs reorientation in a transitive orientation of a comparability graph. These methods are based on special classes of subgraphs called B-stable subgraphs. A stable subgraph F of the undirected graph $G = (X; U)$ is called B-stable if F has no common vertices with any other stable sugraph M of G or F is proper subgraph of M. Algorithms of the reorientation of arcs are based on the factorization procedure.

1. Introduction

Sorting still remains a very actual problem in combinatorics and computer science. Partially ordered sets sorting is a research field with many valorous results [1], [2]. It is well known that partially ordered set can be presented as a transitive oriented graph [8].

We use the reorientation of arcs in a transitive orientation of the graph related to the poset in order to get rearrangement of elements in a partially ordered set. Methods described in [3] offer solution based on the orientation of one arc in the graph. In this paper we try to describe a method for reorientation of a set of arcs.

This article is organised as follows. In Section 2 we describe stable subgraphs as basics for manipulation of the transitive orientation of the graph. In this section we provide an algorithm for the construction of a B-stable subgraph. In Section 3 we describe a method for reorientation of arcs in a graph such that the resulting orientation is also transitive. We present two approaches: minimal reorientation of arcs in graph and reorientation of a given set of arcs.

Received by the editors: July 28, 2015.

2010 *Mathematics Subject Classification.* 05C17, 05C20.

1998 *CR Categories and Descriptors.* G.2.2 [**Discrete Mathematics**]: Graph Theory – *Graph algorithms*.

Key words and phrases. graph factor, B-stable subgraph, transitively orientable graph.

2. Stable subgraphs

Transitively orientable graphs have been studied based on implication classes [3] and stable subgraphs [9]. In this paper we use the second approach.

Recall that a subgraph F with the set of vertices X_F is called stable subgraph of the graph $G = (X; U)$ [9], $X_F \subset X$ if $\forall x \in X \setminus X_F$ only one of the following relations holds:

(1) $[x, y] \in U_G$, $\forall y \in X_F$;
(2) $[x, y] \notin U_G$, $\forall y \in X_F$.

Definition 1. [4] *Graph $F = (X_F; U_F)$ is called B-stable subgraph of the undirected graph $G = (X; U)$ if F is stable subgraph of G and for every stable subgraph M of G one of the following conditions is satisfied:*

(1) $X_F \cap X_M = \emptyset$;
(2) $X_F \subseteq X_M$.

Theorem 1. [4] *If F is a B-stable subgraph of the graph $G = (X; U)$ and $x \in X_G \setminus X_F$ is a vertex adjacent to the set X_F, then for every transitive orientation $\overrightarrow{G}$ only one of the following relations holds:*

(1) $[x, y] \in \overrightarrow{U_G}, \forall y \in X_F$;
(2) $[y, x] \in \overrightarrow{U_G}, \forall y \in X_F$.

Remark 1. *If F is a stable subgraph of the undirected graph $G = (X; U)$, then for every vertex $x \in X_G \setminus X_F$ so that $[x, y] \in U_G$, where $y \in X_F$, the following relation holds:*

$$deg(x) \geq deg(y). \tag{1}$$

Remark 2. *If F is a B-stable subgraph, then:*

$$deg(x) > deg(y). \tag{2}$$

Let $G = (X; U)$ be a transitively orientable graph, and F a subgraph of it. Suppose that $\overrightarrow{G} = (X; \overrightarrow{U})$ is a transitive orientation of G. We will denote by $\overrightarrow{F} = (X_F; \overrightarrow{U_F})$ the directed subgraph of $\overrightarrow{G}$, defined by the subgraph F. The following relation holds, $\overrightarrow{U}_{\overrightarrow{G}} = \overrightarrow{U}_{\overrightarrow{G} \setminus \overrightarrow{F}} \cup \overrightarrow{U}_{\overrightarrow{F}}$, where $\overrightarrow{U}_{\overrightarrow{G} \setminus \overrightarrow{F}}$ is the set of all arcs of the graph $\overrightarrow{G}$ except for arcs from $\overrightarrow{F}$.

We will say that F is independent transitively orientable subgraph of G if for every transitive orientation $\overrightarrow{F}^*$ of F, the set of arcs $\overrightarrow{U}_{\overrightarrow{G} \setminus \overrightarrow{F}} \cup \overrightarrow{U}_{\overrightarrow{F}^*}$ defines a transitive orientation $\overrightarrow{G}^*$ of the graph G. The independent transitively orientable subgraph F will be called ITO-subgraph.

From the facts mentioned above it means that in any transitive orientation $\overrightarrow{G}$ of the undirected graph $G = (X; U)$ if we change arcs of the subgraph

$\overrightarrow{F}$, which in G is ITO-subgraph, then the resulting orientation will be also transitive.

Lemma 1. *The subgraph F of the transitively orientable graph $G = (X; U)$ is B-stable, if and only if F is ITO-subgraph.*

Proof. ($\Longrightarrow$) Suppose that F is a B-stable subgraph. Based on Theorem 1, for all edges in U_F we have only one of the following relations:

(1) $[x, y] \in \overrightarrow{U_G}, \forall y \in X_F$;
(2) $[y, x] \in \overrightarrow{U_G}, \forall y \in X_F$.

where $x \in X_G \setminus X_F$. So, the orientation of the edges in U_F is not dependent on the orientation of edges of $U_G \setminus U_F$ in a transitive oriented graph $\overrightarrow{G}$. So, F is ITO-subgraph.

($\Longleftarrow$) Suppose that F is ITO-subgraph. We need to prove that F is a B-stable subgraph.

The fact that F is ITO-subgraph implies that for every transitie orientation of F only one of the following relation holds:

(1) $[x, y] \in \overrightarrow{U_G}, \forall y \in X_F$;
(2) $[y, x] \in \overrightarrow{U_G}, \forall y \in X_F$.

where $x \in X_G \setminus X_F$. This property implies that $[x, y] \in U_G$ where $x \in X_F$ and $y \in X_G \setminus X_F$. So, the subgraph F is stable. Let M be a stable subgraph in G, and $X_F \cap X_M \neq \emptyset$. Because F is ITO-subgraph the transitive orientation of the graph A does not have any influence on the orientation of F. This is possible only if F is subgraph of M. So, the graph F is B-stable. □

We present a solution for finding of B-stable subgraph. The task of construction of B-stable subgraph is split in two recursive algorithms. In the first algorithm we find a stable subgraph. In the second algorithm we check if the given subgraph is B-stable. The first found B-stable subgraph is returned. The input graph is presented as an adjacent list. Construction of a stable subgraph is based on the Depth-First-Search algorithm. For each processed vertex a special class *processed*(x) with Boolean values $TRUE$ or $FALSE$ is attached. The sets E and P that are presented in the *StableSubgraph* procedure which is defined in the second algorithm.

In the *StableSubgraph* function every vertex of the graph G is processed, and the smallest set of vertexes that satisfy the stable sugraph definition is returned.

We have a vertex x that is adjacent to the stable subgraph and a vertex y that is included in the set of vertices of the stable subgraph as input values. We get a set of vertices that defines a stable subgraph in the output of the algorithm.

Algorithm 1 Stable subgraph

```
function StableSubgraph(x, y)
    for all z ∉ Γ(x) do
        if Γ(z) = Γ(x) then
            E ← z
        end if
    end for
    if E ≠ ∅ then
        E ← y
    end if
    for all z ∈ Γ(y) do
        if processed(z) = FLASE & Γ(z) ∪ {z} ⊆ Γ(x) ∪ {x} then
            processed(z) ← TRUE
            P ← z
            StableSubgraph(x, z)
        end if
    end for
    if E = ∅ then
        return P
    end if
    return E
end function
```

Theorem 2. *Construction of the stable subgraph can be done in* $O(\Delta)$ *time, where* Δ *is the maximum degree of a vertex.*

Proof. All cycles in the $StableSubgraph(x, y)$ function have $\Gamma(x)$ items. Instructions in these cycles run in constant time. It means that for each cycle we have $O(\Gamma(x))$ time. If we chose the maximal degree in graph then time needed for construction of the potential B-stable subgraph is $O(\Delta)$. □

Further, we present an algorithm for construction of a B-stable subgraph based on the $StableSubgraph(x, y)$ function, described above. This algorithm is presented by a recursive function $BSS(G)$. As the previous algorithm this function is based on the Depth-First-Search algorithm. We find a stable subgraph in each level of the graph exploration and check if this subgraph is B-stable. The first found B-stable subgraph is returned. The adjacency list of the graph G is considered as input value of the $BSS(G)$ function. Output of the algorithm is a set of vertices that define a B-stable sugraph. Based on remark 1 and 2, main idea of the algorithm is to subtract a vertex from the graph G, and verify if the resulting subgraph is B-stable.

Algorithm 2 B-stable subgraph

```
function BSS(G)
    if G is not complete graph then
        S ← G
        SORT(S)
        for all x ∈ X_S do
            processed(x) ← TRUE
            P ← ∅ & G ← ∅
            G ← StableSubgraph(x, x)
            if G ≠ S then
                BSS(G)
            end if
        end for
    end if
    return G
end function
```

Theorem 3. *Construction of the B-stable subgraph can be done in $O(n\Delta)$ time, where n and Δ are respectively the number and the maximum degree of vertices of the graph G.*

Proof. The algorithm of the processing of the potential B-stable subgraphs uses the recursive procedure $BSS(G)$. This procedure also explores the graph G using the Depth-First-Search technique.

Graph G is sorted in a descending order based on the degree of the vertices. In the procedure $BSS(G)$ each vertex of the graph is explored. So, for each iteration the $StableSubgraph(x, x)$ function is called. As it is proved in the Theorem 2 the $StableSubraph(x, x)$ procedure can be executed in $O(\Delta)$ time. As a result, we obtain a B-stable subgraph in $O(n\Delta)$ time, if we use the $BSS(G)$ procedure, where n is the number of vertices of the graph G and Δ is the maximum degree of a vertex. □

Remark 3. *Because the StableSubgraph algorithm reduces graph dimension on each iteration, and BSS algorithm stops when graph does not support any changes the BSS function terminates in a finite number of iterations.*

Remark 4. *Based on remark 2 and lemma 1 we can observe that algorithm 2, returns a B-stable subgraph or the whole graph.*

As it was mentioned in the lemma 1 a transitively orientable graph can have more than one orientation. In the next section there will be described a method for reorientation of arcs in a given transitive orientation.

3. Reorientation of a comparability graph

Let F_0 be a B-stable subgraph of the G. We denote by G/F_0 the graph obtained from the graph G by the following rules:

(1) the subgraph F_0 is replaced with the vertex x_{F_0};
(2) all edges $[x, z]$, $\forall x \in X_{F_0}$, $z \in X_G \setminus X_{F_0}$, are replaced with the $[x_{F_0}, z]$.

The graph G/F_0 is called the **graph factor** that corresponds to the B-stable subgraph F_0. The operation of obtaining the graph factor G/F_0 is called **factorization** [5].

If the graph $G^1 = G/F_0$ also contains a B-stable subgraph F_1, then we can get a new graph factor G^1/F_1 from G^1 using the factorization procedure. If this graph also contains a B-stable subgraph then we could repeat the same procedure until we get a graph factor that does not contain any B-stable subgraphs. We can obtain a sequence of undirected graphs:

$$G, G^1 = G/F_0, G^2 = G^1/F_1, \ldots, G^k = G^{k-1}/F_{k-1} \tag{3}$$

with the properties:

(1) F_i is a B-stable subgraph in the graph G^i, where $0 \leq i \leq k-1$, (consider that $G^0 = G$);
(2) the graph $G^k = G^{k-1}/F_{k-1}$ does not contain B-stable subgraphs.

The sequence (3) is called **complete sequence of graph factors** of the graph G.

Reconstruction of the transitive orientation gives the solution for many theoretical and practical problems (see [2], [7]). It is very important to set some specific conditions in order to get the new transitive orientation based on the existing one. We can reorient one arc to achieve a new transitive orientation by using implication classes [6]. In many cases it is necessary to consider reorientation of a set of arcs. Further we will use B-stable subgraphs and factorization procedure in order to accomplish this task.

Definition 2. *If $F = (X_F; U_F)$ is a B-stable subgraph of the transitively orientable graph G, then the set of edges U_F is called the internal factor defined by the subgraph F.*

Let F be a B-stable subgraph of the graph G. Internal factor defined by F is denoted as I_F.

Remark 5. *If transitively orientable graph $G = (X; U)$ does not contain any B-stable subgraphs then the set of edges U_G defines the internal factor of the graph G.*

Let F be a B-stable subgraph of the transitively orientable graph G and I_F is an internal factor defined by the subgraph F, then the next remark holds.

Remark 6. *If* $[x, y]$ *and* $[s, t]$ *are two arcs that are contained in the internal factor* I_F *defined by the B-stable subgraph* F*, then the transitive orientation defined by the arc* $[x, y]$ *is the same as the transitive orientation defined by the arc* $[s, t]$*.*

Definition 3. *Let* $x \in X_G \setminus X_F$*, where* F *is a B-stable subgraph of the transitively orientable graph* G*, is a vertex adjacent to the set* X_F*. Then, the set of edges* $[x, y]$*,* $\forall y \in X_F$ *is called the external factor defined by the subgraph* F*.*

Let F be a B-stable subgraph of the graph G. External factor defined by F is denoted as E_F.

Remark 7. *If* F *is a B-stable subgraph of graph* G*, and* E_F *is an external factor defined by* F*, then for every transitive orientation* $\overrightarrow{G}$*, where* $x \in X_{E_F}$ *and* $y \in X_F$ *only one of the following relations is satisfied:*

(1) $[x, y] \in E_F$;
(2) $[y, x] \in E_F$.

It means that all arcs in an external factor have the same direction.

3.1. **Minimal reorientation.**

Lemma 2. *If* E_{F_i} *is an external factor defined by the B-stable subgraph* F_i *then there is an internal factor* I_{F_j} *defined by a B-stable subgraph* F_j *so that* $E_{F_i} \subseteq I_{F_j}$*,* $1 \leq i \leq k-1$*,* $i+1 \leq j \leq k$*.*

Proof. Let x_{F_i} be a vertex obtained in the factorization operation of the graph G/F_i, and $[x_{F_i}, x_s]$ is an edge of the external factor E_{F_i}. If the vertex x_{F_i} is not contained in another B-stable subgraph, then it can be part of the last graph factor in the complete sequence $G, G^1 = G/F_0, G^2 = G^1/F_1, \ldots, G^k = G^{k-1}/F_{k-1}$. By the Remark 7 the edge $[x_{F_i}, x_s]$ is part of the set U_{G/F_k}. □

Lemma 3. *If* F_i *is a B-stable subgraph of* G *and* $E_{F_i} \subset I_{F_i}$ *then* E_{F_i} *forces the transitive orientation of the internal factor* I_{F_i}*.*

Next, we present an algorithm for the reorientation of minimal amount of arcs in a given transitive orientation. We use the $MinimalReorientation$ function. We use the complete series of graph factors $G^1, G^2, \ldots, G^k$ and the current transitive orientation $\overrightarrow{G}$ of the graph G as the input values for this function. New transitive orientation $\overrightarrow{G'}$ of the graph G is the output of the $MinimalReorientation$ function.

Theorem 4. *A new transitive orientation of the graph* G *using the function* $MinimalReorientation$ *can be done in* $O(k\Delta)$ *time, where* k *is the length of the complete sequence of the graph factors and* Δ *is the maximal degree of a vertex in the graph.*

Algorithm 3 Minimal Reorientation of arcs in a comparability graph

function MINIMALREORIENTATION($G^1, G^2, \ldots, G^k, \overrightarrow{G}$)
  $i \leftarrow 1$
  repeat
    $F_i \leftarrow BSS(G_i)$
    $i \leftarrow i + 1$
  until $U_{F_i} <> \emptyset$
  $\overrightarrow{G'} \longleftarrow \overrightarrow{G^i/F_i}$
  Reorientation of the arcs in the $\overrightarrow{F_i}$
  while $i > 0$ **do**
    $i \leftarrow i - 1$
    $\overrightarrow{G'} \longleftarrow \overrightarrow{G^i/F_i}$
  end while
  return $\overrightarrow{G'}$
end function

Next, we present a method for reorientation of given arcs in a transitive orientation of the graph G.

3.2. **Reorientation forced by a given set of arcs.** Let $G = (X; U)$ be a transitively orientable graph, and $\overrightarrow{G} = (X_G; \overrightarrow{U_G})$ is a transitive orientation of it. The direction of arcs in the subset $\overrightarrow{E_G} \subset \overrightarrow{U_G}$, $2 < |\overrightarrow{E_G}| < |\overrightarrow{U_G}|$, needs to be reversed. A new transitive orientation $\overrightarrow{G'} = (X_G; \overrightarrow{U'_G})$ so that $\overleftarrow{E_G} \subset \overrightarrow{U'_G}$, where $\overleftarrow{E_G} = \{[x, y] | [y, x] \in \overrightarrow{E_G}\}$, should be defined.

We can obtain the set $\overleftarrow{E_G}$ by the reversing of the arcs in $\overrightarrow{E_G}$. The resulting orientation is also transitive. Next, we present the necessary steps for the reconstruction of the orientation that contains all arcs from $\overleftarrow{E_G}$.

We need to apply the factorization procedure on the set $\overleftarrow{E_G}$. Let $[x, y]$ be an arc from $\overleftarrow{E_G}$. If $x \in X_F$, where F is a B-stable subgraph, then we replace the arc $[x, y]$ with the resulting arc $[x', y]$ from the factorization procedure of the subgraph F. If the whole arc is part of the factorized subgraph, then this arc is replaced with the new vertex x'.

We describe an algorithm for the reconstruction of the transitive orientation forced by the given set of arcs $\overleftarrow{E_G}$ based on the idea mentioned above.

This algorithm explores the sequence of the complete graph factors. The exploration of the sequence is done in both directions. In the forward exploration a new set $\overleftarrow{E}_{G^i/F_i}$ is attached for each graph factor. When the sequence of the graph factors is explored backwards, then the new transitive orientation is created based on the set of arcs that forces the orientation. We use the function *ForcedReorientation*. The complete sequence of the graph factors, set $\overleftarrow{E_G}$ and orientation $\overrightarrow{G}$ are considered as the input values for the function. New orientation of the graph is the output of the *ForcedReorientation* procedure.

Algorithm 4 Reorientation of arcs forced by a set of arcs

function FORCEDREORIENTATION($G^1, G^2, \ldots, G^k$, $\overleftarrow{E_G}$, $\overrightarrow{G}$)
    $i \leftarrow 1$
    while $G^i <> G^{i-1}$ **do**
        $\overrightarrow{E_{G^i/F_i}} \leftarrow$ Factorization of $\overrightarrow{E_{G^{i-1}/F_{i-1}}}$
        $i \leftarrow i + 1$
    end while
    while $i > 0$ **do**
        Construction of the $\overleftarrow{E_{G^i/F_i}}$
        $\overrightarrow{G'} \leftarrow \overrightarrow{G^i/F_i}'$
        $i \leftarrow i - 1$
    end while
    return $\overrightarrow{G'}$
end function

Theorem 5. *The run time of the algorithm used for the reorientation of the transitive orientation forced by a given set of arcs is $O(k\Delta)$, where k is the number of graph factors in the complete sequence and Δ is the maximal degree of a vertex in the graph.*

4. Conclusions

In this paper we presented algorithms for reorientation of arcs in comparability graphs. These algorithms are based on B-stable subgraphs and the factorization procedure. We can describe comparability graphs by using the factorization procedure of the B-stable subgraphs. Also, B-stable subgraphs can be used in calculation of number of the transitive orientations in a comparability graph. We consider two cases for the accomplishment of the comparability graph reorientation: minimal amount of arcs reorientation and reorientation of a specified set of arcs in a given transitive orientation. We proved that both algorithms run in polynomial time.

References

[1] C. Daskalakis, R. M. Karp, E. Mossel, S. Riesenfeld, E. Verbin. *Sorting and Selection in Posets.* CoRR, abs/0707.1532, (2007)

[2] V.A. Evstigneev. *Applications of the graph theory in programming.* Nauka, Moskow, 352 p. 1985. (in Russian).

[3] M. C. Golumbic. *Graph Theory and Perfect Graphs* (Annals of Discrete Mathematics) 57. North-Holland Publishing Co., ISBN: 978-0-444-51530-8, Amsterdam, (2004).

[4] N. Grigoriu. *B-stable subgraphs in undirected graphs.* The third conference of Mathematical Society of the Republic of Moldova, Chişinău, pp. 354–357, (2014).

[5] N.Grigoriu. *Minimal stable subgraphs in undirected graphs.* International conference, Mathematics & Information Technologies: Research and Education(MITRE-2013), Chişinău, pp. 48–49, (2013).

[6] P. Klavk, J. Kratochvl, T. Krawczyk, B. Walczak. *Extending Partial Representations of Function Graphs and Permutation Graphs.* Algorithms ESA 2012. 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings. pp. 671 – 682. ISBN: 978-3-642-33090-2

[7] B. Sands. *Unsolved problems.* Order 1, p. 311 – 313, (1985).

[8] B. Schroeder. *Ordered Sets: An Introduction.* Birkhuser, ISBN-13: 978-0817641283, (2003)

[9] A. A. Zykov. *Fundamentals of graph theory.* Moskow, ISBN: 5-9502-0057-8, (2004), (in Russian).

Faculty of Mathematics and Computer Science, State University of Moldova, Chisinau

E-mail address: {s.cataranciuc|grigoriunicolae}@gmail.com