Computer Science Journal of Moldova, vol.**23**, no.1(67), 2015

Construction of a transitive orientation using B-stable subgraphs*

Nicolae Grigoriu

Abstract

A special method for construction of transitive orientations of the undirected graph $G = (X; U)$ is proposed. The method uses an iterative procedure for factorization of graph G. Factorization procedure consists in replacing of a B-stable subgraph with a vertex. Transitive orientations are obtained by a polynomial time algorithm which is presented in the paper.

Keywords: stable subgraph, B-stable subgraph, transitive orientation, graph factor

1 Introduction

Transitively orientable graphs offer solutions for some theoretical problems [3], [8] and have many practical applications [2], [5], [9], [10]. This class of graphs was studied by many mathematicians [4], [6], [8] et al. As a result, a number of theoretical results and algorithms [1] was obtained. Transitively orientable graphs can be characterised by some special structures, like stable subgraphs and non-triangulated chains. In the mathematical literature a transitively orientated graph $\overrightarrow{G} = (X; \overrightarrow{U})$ is the graph for which there is satisfied the transitive relation, $[x, y] \in \overrightarrow{U_G} \,\&\, [y, z] \in \overrightarrow{U_G} \Longrightarrow [x, z] \in \overrightarrow{U_G}$, where $x, y, z \in X_G$. A transitively orientable graph is an undirected graph for which such an orientation of edges can be assigned, that the resulted directed graph is transitive.

* This work was supported by 11.817.07.47A project Ref. Nr. 53/instU

2 B-stable subgraphs

Definition 1. *[11] Subgraph $G(A)$ is called stable subgraph of the graph G, if $\forall x \in X \setminus A$ and $\forall y \in A$ only one of the following relations holds:*

1. $[x, y] \in U_G$;
2. $[x, y] \notin U_G$.

Definition 2. *[7] Stable subgraph F of the undirected graph $G = (X; U)$ is called B-stable subgraph if 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. *[7] 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:*

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

We use the Depth-First-Search algorithm in order to find a B-stable subgraph in the graph G. For each level a potential subgraph is obtained as a candidate to be a B-stable subgraph. We use the recursive procedure $POTENTIAL(x, y)$. As well, for each processed vertex a special class $processed(x)$ with boolean values $TRUE$ or $FALSE$ is attached.

Algorithm 1.

Input: Vertex x that is adjacent to the potential B-stable subgraph and a vertex y that is included in the set of vertices of the potential B-stable subgraph.
Output: The set of vertices that defines a potential B-stable subgraph.

```
Procedure POTENTIAL(x, y)
    For each z ∉ Γ(x):
      If Γ(z) = Γ(x):
        E ⟵ z;
    If E ≠ ∅:
      E ⟵ y;
    For each z ∈ Γ(y):
      If processed(z) = FALSE & Γ(z)∪{z} ⊆ Γ(x)∪{x}:
        processed(z) ⟵ TRUE;
        P ⟵ z;
        POTENTIAL(x, z);
    If E = ∅:
      Return P;
Return E;
```

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

Proof. All cycles in the $POTENTIAL(x, y)$ procedure 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 choose the maximal degree in graph, then time needed for construction of the potential B-stable subgraph is $O(\Delta)$. □

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

Proof. The algorithm of processing potential B-stable subgraphs uses the recursive procedure $SBS(G)$. 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 $SBS(G)$ each vertex of the graph is explored. So, for each iteration the procedure $POTENTIAL(x, x)$ is called. As it is proved in the Theorem 2, the procedure $POTENTIAL(x, x)$ can be executed in $O(\Delta)$ time. As a result, we can obtain a B-stable subgraph in $O(n\Delta)$ time, if we use the $SBS(G)$ procedure, where n is the number of vertices of the graph G and Δ is the maximum degree of a vertex. □

This procedure is presented in the algorithm below.

Algorithm 2.

```
Input: The adjacency list of the graph G.
Output: B-stable F or the graph G if it doesn't contain
any stable subgraph.
Procedure SBS(G)
    If G is not a complete graph:
      S ⟵ G;
      SORT(S);
      For each x ∈ X_S:
        processed(x) ⟵ TRUE;
        P ⟵ ∅; E ⟵ ∅:
        G ⟵ POTENTIAL(x, x):
        If G ≠ S:
          SBS(G);
Return G;
```

3 Transitive orientations of the graph

Let F_0 be a stable subgraph of the graph $G = (X; U)$. We denote by G/F_0 the graph that can be obtained from G by the following rules:

1. B-stable 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 edge $[x_{F_0}, z]$.

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

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

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

with the following properties:

a) F_i is a B-stable subgraph in the graph G^i, where $0 \leq i \leq k-1$, (consider that $G^0 = G$);

b) the graph $G^k = G^{k-1}/F_{k-1}$ does not contain B-stable subgraphs.

The sequence (3) constructed by the rules mentioned above is called **complete sequence of graph factors** of the graph G. Mention that this sequence contains $k+1$ undirected graphs.

Let A and B be two B-stable subgraphs of the undirected graph $G = (X; U)$.

Lemma 1. *If*

$$G, G^1 = G/A, G^2 = G^1/A_1, \ldots, G^k = G^{k-1}/A_{k-1} \tag{4}$$

$$G, G^1 = G/B, G^2 = G^1/B_1, \ldots, G^t = G^{t-1}/B_{t-1} \tag{5}$$

are two complete sequences of graph factors, then $k = t$.

Proof. In other words, we should prove that the number of steps needed to get the graph, on which the factorization operation cannot be applied for each subgraph, is the same.

From Lemma 1 [7] it results that B-stable subgraphs A and B are independent. So, if we apply the factorization procedure in the graph G on the subgraph A, then the resulting graph factor $G^1 = G/A$

contains the B-stable subgraph B. The same logic can be applied on the factorization of the subgraph B. So, it doesn't matter how we choose the B-stable subgraph on which the factorization operation is applied. Certainly, other B-stable subgraphs are in the resulting graph factors.

Next, we need to show that a bijective application is between the graphs G/A and G/B. In other words, we should prove that the factorization of the subgraph A does not generate another B-stable subgraph that cannot be obtained by factorization of the subgraph B.

Let x_A be a vertex obtained in the factorization operation applied on the B-stable subgraph A. Suppose that in the resulting graph $G^1 = G/A$ there is another B-stable subgraph A_1 so that $x_A \in X_{A_1}$. Based on Lemma 1 from [7], the intersection of the B-stable subgraphs is an empty set. It means that in the sequence $G, G^1 = G/A, G^2 = G^1/A_1, \ldots, G^k = G^{k-1}/A_{k-1}$ there can not be another B-stable subgraph that contains vertex x_A. We obtain a new vertex that can be part of another B-stable subgraph as the result of the factorization operation. This graph also can be used in the factorization of the resulting graph factor. This fact indicates that depending on the vertex which was factorized first, every resulting B-stable subgraph from the factorization operation can be obtained from different steps of this operation. The order of the graphs can be changed in the construction of the sequence of the graph factors, but not the number of them. This lemma is proved based on these results. □

Remark 3. *For every sequence of the graph factors of the graph G the number of transitive orientations of the G is the same.*

Let $G^0 = G, G^1 = G^0/F_0, G^2 = G^1/F_1, \ldots, G^{p+1} = G^p/F_p$ be a complete sequence of graph factors, where G^{p+1} does not contain any B-stable subgraphs, F_i is a B-stable subgraph of the graph G^i, $0 \leq i \leq p$.

Theorem 4. *Number of the transitive orientations of the graph G can* *be calculated by the following formula:*

$$\tau(G) = \tau(G^{p+1}) \prod_{i=1}^{p} \tau(F_i). \quad (6)$$

The number of transitive orientations of the resulting graph factor is multiplied by the number of transitive orientations of the B-stable subgraph on which the factorization was applied, because the transitive orientation of the B-stable subgraphs is obtained independently from the rest of the graph.

Algorithm 3. *The algorithm for calculation of number of transitive orientations in a graph*

Input: undirected graph G.
Output: Number of transitive orientations $\tau(G)$.
Initialisation: $\tau(G) \longleftarrow 1$; $i \longleftarrow 1$, $G/F_0 \longleftarrow G$.
Step 1. Determination of B-stable subgraph F_i;
Step 2. Calculation of number $\tau(F_i)$;
Step 3. $\tau(G) \longleftarrow \tau(G) \cdot \tau(F_i)$;
*Step 4. If $\tau(G) = 0$ **STOP**: G is not transitively orientable;*
*Step 5. If $F_i = G/F_i$ **STOP**: Return $\tau(G)$;*
Step 6. Generate graph factor G/F_i;
Step 7. $i \longleftarrow i + 1$; Go to Step 1.

Theorem 5. *Calculation of the transitive orientation of the graph using B-stable subgraphs can be done in $O(kn\Delta)$ time, where k is the number of B-stable subgraphs in the sequence of the graph factors, n is the number of vertices in G and Δ is the maximum degree of a vertex.*

Proof. Consider that k is the length of the complete sequence of graph factors of the graph G. This sequence is generated during the graph factorization. The SBS procedure is called for every graph factor. This procedure uses $O(n\Delta)$ time. Calculation of the number $\tau(F_i)$ has no influence on the run time of the algorithm because it is constant. The 6th step of the algorithm runs in $O(n)$ time. Based on the mentioned facts, we have that the execution time of the algorithm is $O(kn\Delta)$. □

Compared to the algorithm developed by M.C.Golumbic in [6], which runs in $O(m\Delta)$ time, where m is the number of the edges of the graph and Δ is the maximum degree of a vertex, it looks like this algorithm is not so efficient. But many examples have proved that this algorithm is as good as the one presented in [6].

Next, we analyse the algorithm for construction of a transitive orientation based on the sequence of the graph factors.

Algorithm 4. *Algorithm for construction of a transitive orientation*

Input: Complete series of graph factors.
Output: A transitive orientation $\overrightarrow{G}$ of the graph G.
Step 1. Construction of transitive orientation of the graph factor G^i/F_i;
Step 2. If $i = 0$ Return: Orientation $\overrightarrow{G^0/F_0} = \overrightarrow{G}$;
Step 3. $i \longleftarrow i - 1$;
Step 4. Go to Step 1.

Theorem 6. *Construction of the transitive orientation of the graph can be made in $O(k\Delta)$ time, where Δ is the maximal degree of a vertex and k is the number of B-stable subgraphs in the sequence of the graph factors in G.*

Proof. The previous algorithm offers in each iteration a transitive orientation of the B-stable subgraphs. A transitive orientation is generated during the exploration of the sequence of the B-stable subgraphs. So, construction of the transitive orientation can be achieved in $O(\Delta)$ time. Exploration of the sequence of B-stable subgraphs can be executed in $O(k)$ time. In this case, total time necessary for a single transitive orientation is $O(k\Delta)$. □

4 Construction of a transitive orientation with a given set of arcs

The set of arcs that induces a transitive orientation is denoted as $\overrightarrow{A_G}$. The set $\overrightarrow{A_G}$ formally can be presented as follows: $\overrightarrow{A_G} = \{[x, y] \in \overrightarrow{U_G} | \overrightarrow{G} = (X_G; \overrightarrow{U_G})\}$, where $\overrightarrow{G}$ is a transitive orientation.

It is clear that not every set of arcs $\overrightarrow{A_G}$ is a subset of the $\overrightarrow{U_G}$. Next, we present conditions over set $\overrightarrow{A_G}$ to obtain $\overrightarrow{A_G} \subseteq \overrightarrow{U_G}$.

Remark 4. *If the arcs $[x, y]$, $[y, z]$, $[z, x] \in \overrightarrow{A_G}$, then the set $\overrightarrow{A_G}$ does not define a transitive orientation.*

Definition 3. *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 subgraph, 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 4. *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 the transitively orientable graph G, and E_F is an external factor defined by F, then for each $x \in X_{E_F}$ 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.

We can formalize the algorithm for the construction of the transitive orientation defined by the set of arcs $\overrightarrow{A_G}$ from the results mentioned above. This algorithm is split into two stages. We obtain a graph on which there cannot be applied the factorization operation as well as the set $\overrightarrow{A_{G/F_k}}$ in the first stage. The transitive orientation defined by the set $\overrightarrow{A_G}$ is obtained in the second stage of the algorithm.

Theorem 7. *Construction of the transitive orientation with a given set of arcs can be done in $O(pk\Delta)$ time, where p is the number of arcs in $\overrightarrow{A_G}$, k is the number of graph factors in the sequence G, and Δ is the maximal degree of the G.*

Proof. Algorithm of construction of a transitive orientation forced by a given set of arcs is a modification of the algorithm of construction of an arbitrary transitive orientation. Time needed for the construction of the transitive orientation with a given set of arcs is dependent only on the set $\overrightarrow{A_G}$ and number of B-stable subgraphs. So, it is necessary to explore all graph factors in order to get a transitive orientation for the graph. In every graph factor we define a transitive orientation for the arcs in the set $\overrightarrow{A_G}$. Suppose that k is the number of graph factors in sequence and p is the number of arcs in $\overrightarrow{A_G}$. As a result, the running time of the algorithm is $O(pk\Delta)$. □

If the set $\overrightarrow{A_G}$ has only one arc, then the time needed for construction of a transitive orientation is $O(k\Delta)$. Number of graph factors cannot be greater than number of edges in graph. The time needed for construction of a transitive orientation forced by an arc using operation of the graph factorization based on B-stable subgraphs is less than the time needed for construction of a transitive orientation using implication classes. The algorithm for construction of a transitive orientation forced by a set of arcs is presented below. This algorithm explores the sequence of graph factors. In the first stage we get the sequence of graph factors and the modified set of arcs $\overrightarrow{A^i_G}$ for each graph. In the second stage the complete sequence of graph factors is parsed backwards. We obtain a transitive orientation for each graph factor until the last graph is the needed one.

Algorithm 5. *Algorithm for construction of a transitive orientation based on the set* $\overrightarrow{A_G}$

Input: Transitively orientable graph G*, the set of arcs* $\overleftarrow{A_G}$.
Output: Transitive orientation $\overrightarrow{G}$ *of the graph* G.
STAGE I
Step 1. If $\overrightarrow{A_G}$ *is not transitively orientable:* ***STOP*** *- The set* $\overleftarrow{A_G}$ *does not define a transitive orientation;*
Step 2. $i \longleftarrow 1$, $\overrightarrow{A_{G^i/F_i}} \longleftarrow \overrightarrow{A_G}$;
Step 3. Defines B-stable subgraph F_i;
Step 4. Constructs graph factor G^i/F_i;
Step 5. Constructs set $\overrightarrow{A_{G^i/F_i}}$;
Step 6. If $G^i/F_i = G^{i-1}/F_{i-1}$: *Go to the 2nd stage;*
Step 7. $i \longleftarrow i+1$;
Step 8. Go to Step 2;
STAGE II
Initialisation: $i \longleftarrow k$;
Step 1. $p_i \longleftarrow |A_{G^i/F_i}|$, $j \longleftarrow 1$;
Step 2. If arc $u \in \overrightarrow{A_{G^i/F_i}}$ *another transitive orientation* ***STOP*** *- There cannot be constructed a transitive orientation defined by the set* $\overrightarrow{A_G}$;
Step 3. Construction of transitive orientation $\overrightarrow{G^i/F_i}$ *defined by the arc* u_{i_j};
Step 4. If $j \geq p_i$: *Go to Step 6;*
Step 5. $j \longleftarrow j+1$, *Go to Step 2;*
Step 6. Orientation of the edges that are not dependent on the set $\overrightarrow{A_{G^i/F_i}}$;
Step 7. If $i = 0$: *Return* $\overrightarrow{G^0/F_0}$;
Step 8. $i \longleftarrow i-1$;
Step 9. Go to Step 1;

5 Conclusions

In this article we have described algorithms for construction of an arbitrary transitive orientation and construction of a transitive orientation forced by the set of arcs.

Methods presented in [6] offer solution for construction of a transitive orientation based on the direction of only one arc. Algorithms described in this paper can be applied for transitive orientation of the graph forced by a set of arcs. This generalization has applications in many theoretical and practical problems.

References

[1] J. Bang-Jensen, G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, 776 p. Berlin 2007.

[2] S. C. Basak, G. Restrepo, J. L. Villaveces. *Advances in Mathematical Chemistry and Applications.* vol. 1. eISBN: 978-1-60805-928-7. 337 p. 2014

[3] F. Bonomoa, S. Mattiab, G. Orioloc. *Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem.* Theoretical Computer Science, vol. 412, nr. 45, pp. 6261 – 6268. 2011.

[4] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, K. Vušković. *Recognizing Berge graphs.* Combinatorica 25(2), pp. 143–186, 2005.

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

[6] M.C. Golumbic. *Graph Theory and Perfect Graphs* (Annals of Discrete Mathematics) 57. North-Holland Publishing Co., ISBN: 978-0-444-51530-8, Amsterdam, 340 p. 2004.

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

[8] S. Penrice. *On-line algorithms for ordered sets and comparability graphs.* Discrete Applied Mathematic 60, pp. 319–329, 1995.

[9] Y. Xuejun, W. Li, X. Jingling, D. Yu, Z. Ying. *Comparability Graph Coloring for Optimizing Utilization of Stream Register Files in Stream Processors.* Proceedings of the 14th ACM SIGPLAN symposium on Principles and practice of parallel programming 44(4), pp. 111–120, 2009.

[10] R. Wisniewski, A. Karatkevich, M. Adamski, D. Kur. *Application of comparability graphs in decomposition of Petri nets.* Human System Interactions (HSI), 2014 7th International Conference on. pp. 216 – 220, 2014.

[11] A.A. Zykov. *Fundamentals of graph theory.* Moskow, ISBN: 5-9502-0057-8, 664 p. 2004. (in Russian).

Nicolae Grigoriu Received February 8, 2015

Nicolae Grigoriu
State University of Moldova
60 A. Mateevici, MD-2009, Chişnău, Republic of Moldova
E–mail: *grigoriunicolae@gmail.com*