Computational Geometry 46 (2013) 774–804

Contents lists available at SciVerse ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

On upward point set embeddability ☆

Michael Kaufmann [a], Tamara Mchedlidze [b,*], Antonios Symvonis [c]

[a] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany*
[b] *Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Germany*
[c] *Dept. of Mathematics, National Technical University of Athens, Greece*

A R T I C L E I N F O

Article history:
Received 8 September 2011
Accepted 23 November 2012
Available online 27 November 2012
Communicated by D. Wagner

Keywords:
Point set embedding
Upward planar digraphs
Directed trees
Outerplanar digraphs

A B S T R A C T

We study the problem of upward point set embeddability, that is the problem to decide whether an n-vertex directed graph has an upward planar drawing when its vertices have to be placed on the points of a given point set of size n. We first present some positive and negative results concerning directed trees and convex point sets. Next, we prove that upward point set embeddability can be solved in polynomial time for the case of a directed tree and a convex point set. Further, we extend our approach to the class of outerplanar directed graphs. This implies that upward point set embeddability can be efficiently solved for the case of convex point sets. Finally, we show that the general problem of upward point set embeddability is $\mathcal{NP}$-complete even for 2-convex point sets.
© 2012 Elsevier B.V. All rights reserved.

1. Introduction

A *planar straight-line embedding* of a graph G into a point set S is a mapping of each vertex of G to a distinct point of S and of each edge of G to the straight-line segment between the corresponding end points so that no two edges cross each other. Planar straight-line embeddings for outerplanar graphs and trees were studied by Gritzmann et al. [1], Bose [2], and Bose et al. [3]. Cabello [4] proved that the problem to decide whether a given planar graph admits a planar straight-line embedding into a given point set is $\mathcal{NP}$-hard. Planar graph embeddings into point sets, where edges are allowed to bend, have also been studied (see, e.g., [5–9]).

An *upward planar directed graph* is a digraph that admits a planar drawing such that each edge is represented by a curve monotonically increasing in the y-direction. An *upward point set embedding* (*UPSE* for short) of an upward planar digraph G into a point set S is a mapping of each vertex of G to a distinct point of S and of each edge to the straight-line segment between its corresponding end points such that no two edges cross and for each edge (u,v) the condition $y(u) < y(v)$ holds, where by $y(u)$ we denote the y-coordinate of the point to which vertex u was mapped. *Upward point set embeddability* is the decision problem of whether a given digraph has an UPSE into a given point set.

Upward point set embeddability was first studied by Giordano et al. [10]. The authors studied the version of the problem where bends on edges are allowed and showed that every planar *st*-digraph admits an upward point set embedding with at most two bends per edge. Upward point set embeddability with a given mapping, i.e., where a correspondence between the nodes and the point set is part of the input, was studied in [11,12]. Recently, straight-line upward point set embeddings were studied in [13,14] and many interesting partial results were presented. More specifically, several families of directed

☆ This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.
* Corresponding author.
E-mail addresses: mk@informatik.uni-tuebingen.de (M. Kaufmann), mched@iti.uka.de (T. Mchedlidze), symvonis@math.ntua.gr (A. Symvonis).

0925-7721/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comgeo.2012.11.008

trees were presented, which admit an UPSE into every convex point set, i.e., directed caterpillars, directed hourglass trees. On the other hand, it was demonstrated in [14] that there exist directed trees which do not admit an UPSE into all convex point sets. An immediate question that arises from these facts is whether the existence of an UPSE of a directed tree into a convex point set can be efficiently tested. In this paper we answer this question in affirmative. Further, we extend this result to directed outerplanar graphs by showing that it can be tested in polynomial time whether a directed outerplanar graph admits an upward planar embedding into a given point set.

A question naturally raised by this result is how the complexity of the problem changes if we consider more complex point sets. Regarding this we prove that the upward point set embeddability problem is $\mathcal{NP}$-complete even when the point set consists of two nested convex point sets.

Summarizing, the contributions of this paper are as follows:

1. We extend the positive results given in [13,14] by showing that any directed switch tree admits an upward planar straight-line embedding into every point set in convex position.
2. We study directed k-switch trees, a generalization of directed switch trees (a directed 1-switch tree is exactly a directed switch tree). From the construction given in [14, Theorem 5], we know that for $k \geqslant 4$ not every directed k-switch tree admits an upward planar straight-line embedding into any convex point set. Then we fill the gap for 2 and 3-switch trees, by showing that, for any $k \geqslant 2$ there exists a class of directed k-switch trees $\mathcal{T}_n^k$, and a point set S in convex position, such that any $T \in \mathcal{T}_n^k$ does not admit an upward planar straight-line embedding into S.
3. We present a polynomial dynamic programming algorithm for testing whether a directed tree admits an UPSE into a given convex point set.
4. We extend our approach to directed outerplanar graphs and show that it can be tested in polynomial time whether a given directed outerplanar graph admits an UPSE into a given point set. Since any graph admitting a planar embedding into a convex point set is an outerplanar graph, our result implies that the upward point set embeddability can be efficiently solved for convex point sets and general directed graphs.
5. Finally, we prove that the general upward point set embeddability is $\mathcal{NP}$-complete and it remains so even when the point set consists of two nested convex point sets.

Preliminary versions of the research presented in this paper have appeared in [15,16].

The paper is structured as follows: In Section 2 we present the notation that is used throughout the paper. In Section 3 we present positive and negative results for directed trees. In Section 4 we show that the upward point set embeddability can be solved in polynomial time for directed trees and convex point sets. In Section 5 we extend this result to directed outerplanar graphs. The construction is along the same lines as for directed trees, but technically more involved. Therefore, we decided to keep the descriptions separated. Finally, in Section 6 we present the $\mathcal{NP}$-hardness result.

2. Notation – preliminaries

2.1. Point sets

Let S be a set of n points on the plane. We assume that the points of S are in general position, i.e., no three of them lie on the same line. Moreover, we also assume that no two points of S share the same y-coordinate; if they do, a slight rotation of the coordinate axes can ensure that all points have distinct y-coordinates. The *convex hull* $CH(S)$ of S is the point set that is obtained as a convex combination of the points of S. A point set such that no point is in the convex hull of the others is called a *point set in convex position*, or a *convex point set*. Given a point set S, by $t(S)$ (resp., $b(S)$) we denote the top (bottom) point of S i.e., the point with the largest (resp., smallest) y-coordinate.

Consider a point set S and its convex hull $CH(S)$. Let $L_1 = S \setminus CH(S), \dots, L_m = L_{m-1} \setminus CH(L_{m-1}), L_{m+1} = L_m \setminus CH(L_m)$. If m is the smallest integer such that $L_{m+1} = \emptyset$, we say that S is *m-convex*.

A *one-sided convex point set* S is a convex point set in which $b(S)$ and $t(S)$ are adjacent on the border of $CH(S)$. If $t(S)$ and $b(S)$ appear adjacent and in this order on the border of $CH(S)$ as we traverse it in the clockwise (resp., counterclockwise) direction, then the one-sided convex point set is called a *left-sided convex point set* (resp., *right-sided convex point set*). A point set consisting of at most two points is considered to be either a left-sided or a right-sided convex point set. A convex point set which is not one-sided, is called a *two-sided convex point set*.

Each given convex point set S may be considered to be the union of two specified (at the time S is given) one-sided convex point sets, one left-sided which is denoted by $L(S)$ and is referred to as the *left side* of S, and one right-sided which is denoted by $R(S)$ and is referred to as the *right side* of S. When there is no confusion regarding the point set S we refer to, for simplicity, we use the terms L and R instead of $L(S)$ and $R(S)$, respectively. Each of the points $b(S)$ and $t(S)$ belongs to either $L(S)$ or $R(S)$ but not to both.

A subset of points of a convex point set S is called *consecutive* if its points appear consecutively as we traverse the convex hull of S in clockwise direction. Given that all points of S have distinct y-coordinates, we can refer to the first, the second, the third, etc., lowest point on the left (right) side of S. By p_i^L, $1 \leqslant i \leqslant |L(S)|$, we denote the i-th lowest point on the left side of S. Similarly, by p_i^R, $1 \leqslant i \leqslant |R(S)|$, we denote the i-th lowest point on the right side of S.

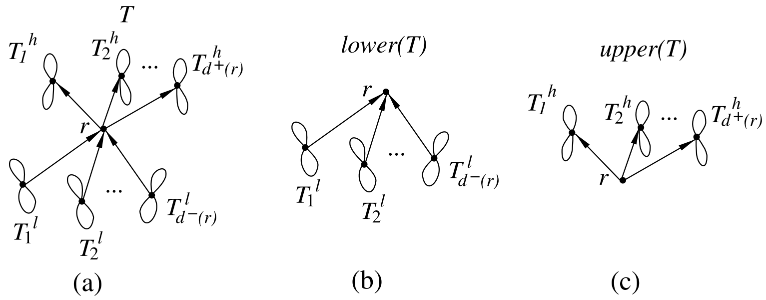

Fig. 1. (a) A rooted at vertex r tree T and its subtrees $T_1^l, \ldots, T_{d^-(r)}^l$, $T_1^h, \ldots, T_{d^+(r)}^h$. (b) The subtree *lower*(T) of T. (c) The subtree *upper*(T) of T.

Let $S_{a..b,c..d} = \{p_i^L \mid a \leqslant i \leqslant b\} \cup \{p_i^R \mid c \leqslant i \leqslant d\}$ denote the subset of S consisting of $b - a + 1$ consecutive points on the left side of S, starting from point p_a^L in the clockwise direction, and of $d - c + 1$ consecutive points on the right side, starting from point p_c^R in the counterclockwise direction. For simplicity, for a one-sided point set S we use the notation $S_{a..b}$.

In this paper, we assume that queries of the form *"Find the i-th point on the left/right side of the convex point set S"* can be answered in $O(1)$ time, e.g., the points on each side of S are stored in an array in ascending order of their y-coordinates.

Let l be a line on the plane, which is not parallel to the x-axis. We say that point p *lies to the right of* l (resp., *to the left of* l) if p lies on a semi-line that originates on l, is parallel with the x-axis, and is directed towards $+\infty$ (resp., $-\infty$). Similarly, if l is a line on the plane, which is not parallel to the y-axis, we say that point p *lies above* l (resp., *below* l) if p lies on a semi-line that originates on l, is parallel with the y-axis, and is directed towards $+\infty$ (resp., $-\infty$).

2.2. Graphs

A graph $G = (V, E)$ is *planar* if it has a drawing Γ without edge crossings. Drawing Γ splits the plane into connected regions called *faces*; the unbounded region is the *outer* face. The cyclic ordering of edges around each vertex of Γ together with a choice of the outer face is a *planar embedding* of G. A graph G is called *outerplanar* if it has a drawing such that all it vertices appear in the outer face. The corresponding embedding is called *outerplane*.

The graphs we study in this paper are directed, we call them *digraphs* for short. By (u, v) we denote an arc directed from u to v. We use the notation $\{u, v\}$ to denote arc (u, v) of digraph $G = (V, E)$ if $(u, v) \in E$ or arc (v, u) if $(v, u) \in E$. If vertex u of digraph G is mapped to point p and v of G is mapped to point q, which is located below p, then we say that $\{u, v\}$ is drawn *upward* if $(v, u) \in E$.

A vertex of a digraph with in-degree (resp. out-degree) equal to zero is called a *source* (resp. *sink*). A vertex of a digraph which is either a source or a sink is referred to as a *switch*.

A digraph D is called *path-DAG* (resp. *cycle-DAG*), if its underlying graph is a simple path (resp. cycle). A *monotone path* $(v_1, v_2, \ldots, v_k)$ is a path-DAG containing arcs (v_i, v_{i+1}), $1 \leqslant i \leqslant k - 1$. A *directed tree* T is a digraph whose underlying graph is a tree. An *outerplanar digraph* G is a digraph whose underlying graph is outerplanar. A *switch tree* is a directed tree T, such that, each vertex of T is a switch. Note that the longest monotone path of a switch tree has length one.[1] Based on the length of the longest monotone path, the class of switch trees can be generalized to that of k-switch trees. A *k-switch tree* is a directed tree, such that its longest monotone path has length k. According to this definition a switch tree is a 1-switch tree.

A directed tree T is a *rooted tree* if one of its vertices, denoted by $r(T)$, is designated as its *root*. We then say that T *is rooted at* vertex $r(T)$. By $d^-(v)$ (resp., $d^+(v)$) we denote the in-degree (resp., the out-degree) of vertex v of T. By $d(v)$ we denote the total degree of vertex v, i.e., $d(v) = d^-(v) + d^+(v)$.

Let T be a rooted tree and let $r = r(T)$ be its root. Let $T_1^l, \ldots, T_{d^-(r)}^l$, $T_1^h, \ldots, T_{d^+(r)}^h$ be the rooted subtrees of T obtained by removing from T its root r and r's incident arcs and having as their roots the vertices that are adjacent to r by either an incoming or an outgoing arc (see Fig. 1.a). Trees $T_1^l, \ldots, T_{d^-(r)}^l$, $T_1^h, \ldots, T_{d^+(r)}^h$ are called the *subtrees of* T. Note that the superscripts "l" and "h" indicate whether a particular subtree of T is connected to r by an arc incoming to r or by an arc outgoing from r, respectively. When r has only incoming or only outgoing edges we omit the superscripts "l" and "h".

The rooted subtree of T consisting of T's root, r, together with $T_1^l, \ldots, T_{d^-(r)}^l$ is called the *lower subtree of* T and is also rooted at r. The lower subtree of T is denoted by *lower*(T) (Fig. 1.b). Similarly, the rooted subtree of T consisting of T's root, r, together with $T_1^h, \ldots, T_{d^+(r)}^h$ is called the *upper subtree of* T and is also rooted at r. The upper subtree of T is denoted by *upper*(T) (Fig. 1.c).

Let T be a tree rooted at r and let v be a vertex of T different from r. The removal of the edge of T that is incident to v and is contained in the path connecting v to r produces two subtrees of T. The subtree which contains vertex v is denoted by $T(v)$.

[1] The *length* of a directed path is the number of arcs in the path.

2.3. Known and preliminary results on UPSE of rooted directed trees

We present some known and preliminary results on UPSE of rooted directed trees that will be utilized by our algorithms. Binucci et al. [14] proved the following lemma concerning the placement of the subtrees of T in an UPSE of T on a convex point set.

Lemma 1. *(See Binucci et al. [14].) Let T be an n-vertex directed tree rooted at r and let S be any convex point set of size n. Let $T_1, T_2, \dots, T_{d(r)}$ be the subtrees of T. Then, in any UPSE of T into S, the vertices of subtree T_i are mapped to a set of consecutive points of S, $1 \leqslant i \leqslant d(r)$.*

The following theorem by Binucci et al. [14] presents their negative result for trees and convex point sets:

Theorem 1. *(See Binucci et al. [14].) For every odd integer $n \geqslant 5$, there exists a $(3n+1)$-vertex directed tree T and a convex point set S of size $3n+1$ such that T does not admit an UPSE into S.*

The following lemma represents a tool for proving time complexity bounds in the following sections.

Lemma 2. *Let T be an n-vertex directed tree rooted at vertex r. The values $|lower(T(v))|$ and $|upper(T(v))|$ for all $v \in T$ can be computed in total $O(n)$ time.*

Proof. Values $|lower(T(v))|$ and $|upper(T(v))|$ for all $v \in T$ can be computed by performing a post order traversal of tree T. Thus, assume that vertex v is incident to edges $(v_1^l, v), \dots, (v_{d^-(v)}^l, v)$, $(v, v_1^h), \dots, (v, v_{d^+(v)}^h)$. Note that one of the vertices $v_1^l, \dots, v_{d^-(v)}^l, v_1^h, \dots, v_{d^+(v)}^h$ is contained in the path connecting v with r. Without loss of generality assume that this vertex is v_1^h. Then the values $|lower(T(v))|$ and $|upper(T(v))|$ can be computed as follows

$$\bigl|lower\bigl(T(v)\bigr)\bigr| = \sum_{i=1}^{d^-(v)} \bigl(\bigl|lower\bigl(T\bigl(v_i^l\bigr)\bigr)\bigr| + \bigl|upper\bigl(T\bigl(v_i^l\bigr)\bigr)\bigr| - 1\bigr),$$

$$\bigl|upper\bigl(T(v)\bigr)\bigr| = \sum_{i=2}^{d^+(v)} \bigl(\bigl|lower\bigl(T\bigl(v_i^h\bigr)\bigr)\bigr| + \bigl|upper\bigl(T\bigl(v_i^h\bigr)\bigr)\bigr| - 1\bigr).$$

The computation of all values $|lower(T(v))|$ and $|upper(T(v))|$, for all $v \in T$ takes $O(n)$ time since we perform a single addition for each edge of T. □

The following lemma concerns the UPSE of a rooted tree into a *one-sided* convex point set. It can be considered to be a simple restatement of a result by Heath et al. [17, Theorem 2.1].

Lemma 3. *Let T be an n-vertex directed tree rooted at r and S be a one-sided convex point set of size n. Let $T_1, T_2, \dots, T_{d(r)}$ be the subtrees of T. Then, T admits an UPSE into S with the following properties*:

(i) *Each T_i, $1 \leqslant i \leqslant d(r)$, is drawn on consecutive points of S.*
(ii) *If the root r of T is mapped to point p_r then there is no arc connecting a point of S below p_r to a point of S above p_r.*

By utilizing Lemma 2 and Lemma 3, we prove the following

Lemma 4. *Let T be an n-vertex directed tree rooted at r and S be a one-sided convex point set of size n. Then, an UPSE of T into S, satisfying the properties of Lemma 3, can be obtained in $O(n)$ time.*

Proof. Let $k = |lower(T)|$ be the size of subtree $lower(T)$ (rooted at r). It immediately follows that in an UPSE of T into S satisfying the properties of Lemma 3 there are $k-1$ vertices of T (all belonging to $lower(T)$) that are placed below r. Thus, r is mapped to the k-th lowest point of S. This point, say p_r, can be computed in $O(1)$ time. Having decided where to place the root r, the UPSE of T can be completed in $O(n)$ time by recursively embedding the vertices of $lower(T)$ ($upper(T)$) to the points of S below (above) p_r. Note that by Lemma 2, the values $|lower(T(v))|$, $|upper(T(v))|$ for all $v \in T$ can be computed in total $O(n)$ time. □

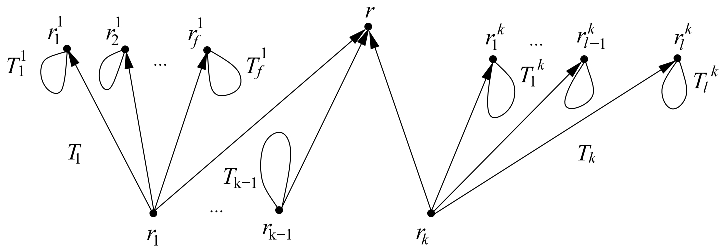

Fig. 2. The switch tree T as described in the proof of Theorem 2.

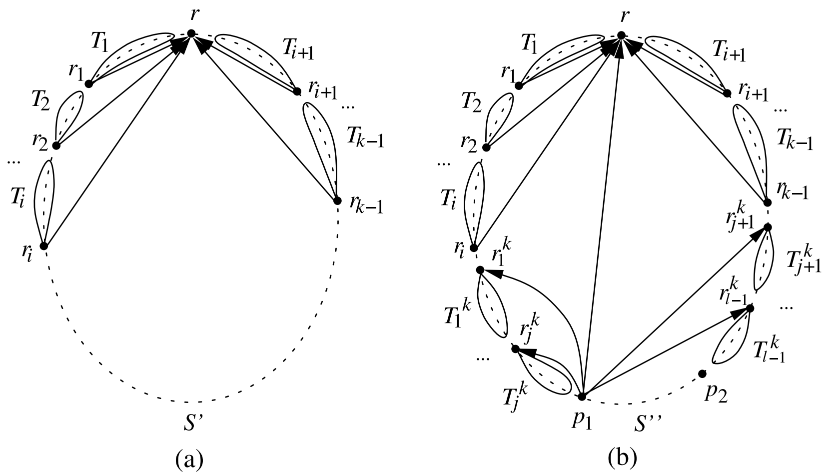

Fig. 3. The construction used in Theorem 2.

3. Positive and negative results for directed trees

3.1. Embedding a switch tree into a convex point set

In this section we enrich the positive results presented in [13,14] by proving that any switch tree has an UPSE into any point set in convex position. During the execution of the algorithms presented in the following lemmata, which embed a tree T into a point set S, a *free point* is a point of S to which no vertex of T has been mapped yet.

Theorem 2. *Let T be a switch tree and S be a convex point set such that $|S| = |T|$. Then, T admits an UPSE into S.*

Proof. Let r be a sink vertex of tree T. In the following we prove that tree T admits an UPSE into S so that vertex r is mapped to the highest point of S.

Let $T_1, \ldots, T_k$, where $k = d(r)$, be the subtrees of T (Fig. 2) and let $r_1, \ldots, r_k$ be the roots of $T_1, \ldots, T_k$, respectively. Observe that, since T is a switch tree and r is a sink, vertices $r_1, \ldots, r_k$ are sources.

We draw T on S as follows. We start by placing the trees $T_1, T_2, \ldots$ on the left side of the point set S as long as they fit, using the highest free points first. This can be done in an upward planar fashion by Lemma 3 (Fig. 3.a). Assume that T_i is the last placed subtree. Then, we continue placing the trees $T_{i+1}, \ldots, T_{k-1}$ on the right side of the point set S. This can be done due to Lemma 3. Note that the remaining free points are consecutive on S, denote these points by S'. To complete the embedding we draw T_k on S'. Let $T_1^k, \ldots, T_l^k$, where $l = d(r_k)$, be the subtrees of T_k and let $r_1^k, \ldots, r_l^k$ be their roots, respectively (Fig. 2). Note that $r_1^k, \ldots, r_l^k$ are all sinks. We start by drawing $T_1^k, T_2^k, \ldots$ as long as they fit on the left side of point set S', using the highest free points first. This can be done in an upward planar fashion by Lemma 3. Assume that T_j^k is the last placed subtree (Fig. 3.b). Then, we continue on the right side of the point set S' with the trees $T_{j+1}^k, \ldots, T_{l-1}^k$. This can be done again by Lemma 3. Note that there are exactly $|T_l^k| + 1$ remaining free points since we have not yet drawn T_l^k and vertex r_k of T_k. Denote by S'' the remaining free points and note that S'' consists of consecutive points of S. If S'' is a one-sided point set then we can proceed by using Lemma 3 again and the result follows trivially. Assume now that S'' is a two-sided convex point set and let p_1 and p_2 be the highest points of S'' on the left and on the right, respectively. W.l.o.g., let $y(p_1) < y(p_2)$. Then, we map r_k to p_1. By using the lemma recursively, we can draw T_l^k on $S'' \setminus \{p_1\}$ so that

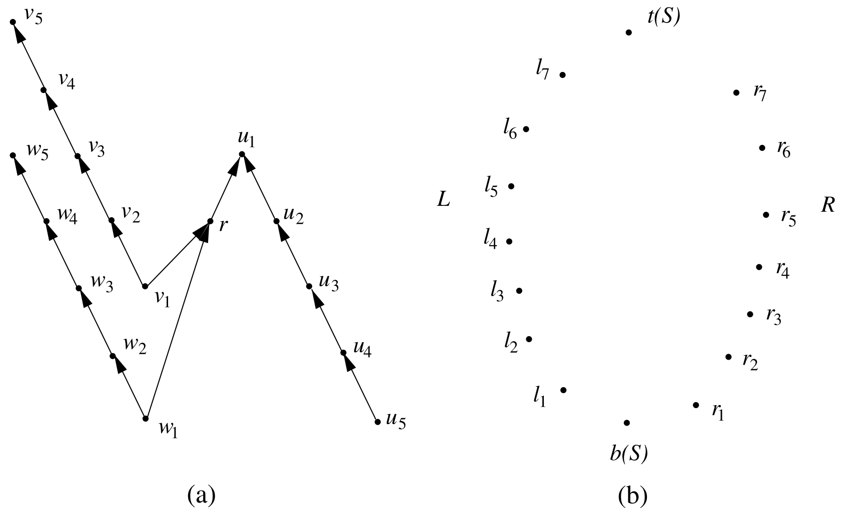

Fig. 4. (a)–(b) A 4-switch tree T and a point set S, such that T does not admit an UPSE into S.

r_l^k is mapped to p_2. The proof is completed by observing that all edges connecting r_k to $r_1^k, \ldots, r_l^k$ and $r_1, \ldots, r_k$ to r are upward and do not cross each other. □

3.2. k-Switch trees which do not have an UPSE into some convex point sets

Binucci et al. [14] (see also Theorem 1) presented a class of trees and corresponding convex point sets, such that any tree of this class does not admit an UPSE into its corresponding point set.

The $(3n+1)$-size tree T constructed in the proof of Theorem 1 [14] has the following structure (see Fig. 4.a for the case $n = 5$). It consists of: (i) one vertex r of degree three, (ii) three monotone paths of n vertices: $P_u = (u_n, u_{n-1}, \ldots, u_1)$, $P_v = (v_1, v_2, \ldots, v_n)$, $P_w = (w_1, w_2, \ldots, w_n)$, (iii) arcs (r, u_1), (v_1, r) and (w_1, r).

The $(3n+1)$-convex point set S, used in the proof of Theorem 1 [14], consists of two extremal points on the y-direction, $b(S)$ and $t(S)$, the set L of $(3n-1)/2$ points $l_1, l_2, \ldots, l_{(3n-1)/2}$, comprising the left side of S and the set R of $(3n-1)/2$ points $r_1, r_2, \ldots, r_{(3n-1)/2}$, comprising the right side of S. The points of L and R are located so that $y(b(S)) < y(r_1) < y(l_1) < y(r_2) < y(l_2) < \cdots < y(r_{(3n-1)/2}) < y(l_{(3n-1)/2}) < y(t(S))$. See Fig. 4.b for $n = 5$.

Note that the $(3n+1)$-node tree T described above is an $(n-1)$-switch tree. Hence a straightforward corollary of Theorem 1 [14] is the following statement.

Corollary 1. *For any $k \geqslant 4$, there exists a k-switch tree T and a convex point set S of the same size, such that T does not admit an UPSE into S.*

From Section 3.1, we know that any switch tree T, i.e., a 1-switch tree, admits an UPSE into any convex point set. The natural question raised from this result and Corollary 1 is whether an arbitrary 2-switch or 3-switch tree has an UPSE into any convex point set. This question is resolved by the following theorem.

Theorem 3. *For any $n \geqslant 5$ and for any $k \geqslant 2$, there exists a class $\mathcal{T}_n^k$ of $(3n+1)$-vertex k-switch trees and a convex point set S, consisting of $3n+1$ points, such that any $T \in \mathcal{T}_n^k$ does not admit an UPSE into S.*

Proof. For any $n \geqslant 5$ we construct the following class of trees (see Fig. 5.a). Let P_u be an n-vertex path-DAG on the vertex set $\{u_1, u_2, \ldots, u_n\}$, enumerated in the order they are presented in the underlying undirected path of P_u, and such that arcs (u_3, u_2), (u_2, u_1) are present in P_u. Let also P_v and P_w be two n-vertex path-DAGs on the vertex sets $\{v_1, v_2, \ldots, v_n\}$ and $\{w_1, w_2, \ldots, w_n\}$ respectively, enumerated in the order they are presented in the underlying undirected path of P_v and P_w, and such that arcs (v_1, v_2), (v_2, v_3) and (w_1, w_2), (w_2, w_3) are present in P_v and P_w, respectively. Let $T(P_u, P_v, P_w)$ be a tree consisting of P_u, P_v, P_w, vertex r and arcs (r, u_1), (v_1, r), (w_1, r).

Let $\mathcal{T}_n^k = \{T(P_u, P_v, P_w) \mid \text{the longest directed path in } P_u,\ P_v \text{ and } P_w \text{ has length } k\}$, $k \geqslant 2$. So, $\mathcal{T}_n^k$ is a class of $(3n+1)$-vertex k-switch trees. Let S be a convex point set as described in the beginning of the section. Next we show that any $T \in \mathcal{T}_n^k$ does not admit an UPSE into point set S.

Let $T \in \mathcal{T}_n^k$. For the sake of contradiction, we assume that there exists an UPSE of T into S. By Lemma 1, each of the paths P_u, P_v and P_w of T is drawn on consecutive points of S. Denote by S_u, S_v and S_w the subsets of point set S, in which P_u, P_v and P_w are mapped to, respectively. Hence $|S_u| = |S_v| = |S_w| = n$. By construction of S, the largest subset

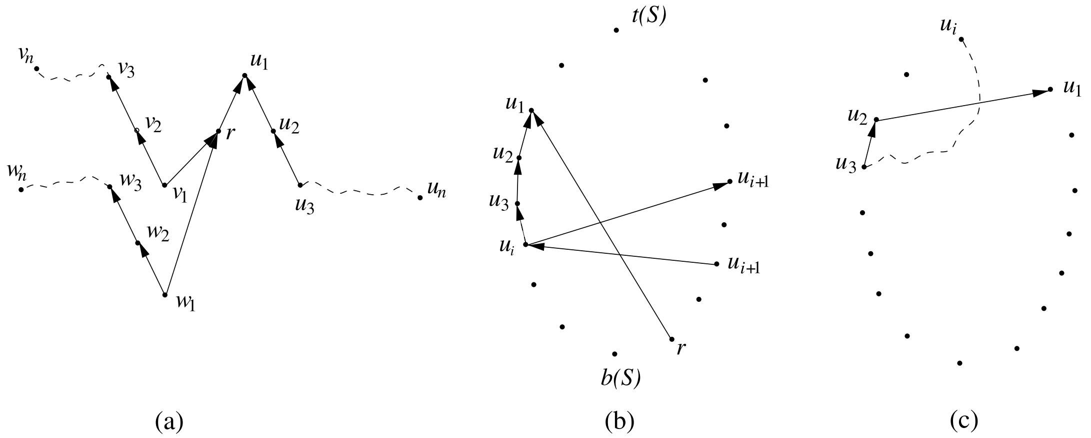

Fig. 5. (a) k-Switch tree, $k \geqslant 2$. (b)–(c) The construction of the proof of Statement 1, Cases 1 to 2.

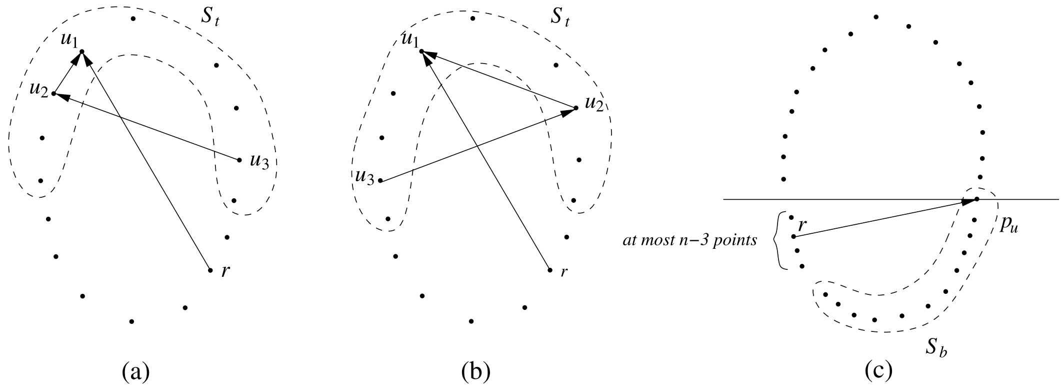

Fig. 6. (a)–(b) The construction of the proof of Statement 1, Cases 3 and 4. (c) The construction used in Statement 3.

of S which is a one-sided convex point set, contains two extremal points of S and has size $\lceil \frac{3n-1}{2} \rceil + 2 < 2n$, when $n \geqslant 5$. Thus, at least one of S_u, S_v and S_w is a two-sided convex point set. We denote by S_b and S_t any two-sided point sets, which consist of consecutive points of S, so that $|S_b| = |S_t| = n$, and $b(S) \in S_b$, $t(S) \in S_t$ respectively. Next, we show that in any UPSE of T on S, P_u cannot be drawn on S_b, while P_v and P_w cannot be drawn on S_t.

Statement 1. *For any UPSE of P_u into S_t there is a crossing created by the arcs of T.*

Proof. Recall that $S_t \subset S$ is a two-sided convex point set, so that $t(S) \in S_t$. Next we consider four cases based on whether vertices u_1, u_2, u_3 are drawn on the same side of S. The cases are exclusive since at least two of the vertices u_1, u_2, u_3 are drawn on the same side of S.

Case 1. Vertices u_1, u_2, u_3 are mapped to the same side of S, possibly including $t(S)$, say w.l.o.g. to the left side of S, see Fig. 5.b. Let u_{i+1} be the first vertex of P_u that is mapped to the right side of S. Notice that vertex u_i has to be below u_1. This is because vertex u_2 has to be below u_1, and if u_i would be above u_1, the path from u_2 to u_i would cross the edge (u_1, r). Then, since r is mapped to a point of $S \setminus S_t$, arc (r, u_1) crosses arc (u_i, u_{i+1}) (or arc (u_{i+1}, u_i)).

Case 2. Out of vertices u_1, u_2, u_3 only u_2, u_3 are mapped to the same side of S, possibly including $t(S)$, say w.l.o.g. to the left side of S, see Fig. 5.c. Then, u_1 is mapped to the right side of S. Note that u_2 cannot be mapped to $t(S)$, because then there is no point for u_1 to be mapped to, so that the drawing is upward. Hence, there is at least one point p higher than the end points of arc (u_2, u_1), that has to be visited by path P_u. Thus, path P_u crosses arc (u_2, u_1).

Case 3. Out of vertices u_1, u_2, u_3 only u_1, u_2 are mapped to the same side of S, possibly including $t(S)$, say w.l.o.g. to the left side of S. Then, u_3 is mapped to the right side of S (Fig. 6.a) and, as a consequence, arcs (r, u_1) and (u_3, u_2) cross.

Case 4. Out of vertices u_1, u_2, u_3 only u_1, u_3 are mapped to the same side of S, possibly including $t(S)$, say w.l.o.g. to the left side of S. Then, u_2 is mapped to the right side of S (Fig. 6.b) and, as a consequence, arcs (r, u_1) and (u_3, u_2) cross. □

The proof of the following statement is symmetrical to the proof of Statement 1.

Statement 2. *For any UPSE of P_u or P_w into S_b there is a crossing created by the arcs of T.*

So, we have proved that there is no upward planar embedding of T into S such that P_u is mapped to a set S_t, or such that P_v or P_w is mapped to a set S_b. Next, we prove that there is also no upward planar embedding of T into S such that P_u is mapped to S_b, and such that P_v or P_w is mapped to S_t.

Statement 3. *There is no UPSE of T into point set S, such that P_u is mapped to the points of S_b.*

Proof. Denote by p_u the point of S_b with the largest y-coordinate, see Fig. 6.c. By the construction of S and since S_b is a two-sided point set which contains n points, we infer that $S \setminus S_b$ contains at most $n-3$ points lower than p_u. Moreover, all of these points are on the side opposite to p_u. We observe the following: (i) r has to be placed lower than p_u, and hence r is placed on the opposite side of that of p_u, (ii) v_1 has to be placed lower than r, and since, by Lemma 1, P_v has to be mapped to consecutive points of S, the whole P_v is mapped to the points on the same side of r and lower than r. However, there are at most $n-4$ free points on the same side of r and lower than r, which is a contradiction since $|P_v| = n$. □

The following statement is symmetrical to Statement 3.

Statement 4. *There is no UPSE of T into point set S, such that P_v or P_w is mapped to the points of S_t.*

As we observed in the beginning of the proof of the theorem, at least one of P_u, P_v, P_w is mapped to a two-sided point set containing either $b(S)$ or $t(S)$. However, as it is proved in Statements 1 to 4 this is impossible. So, the theorem follows. □

4. Decision algorithm for UPSE of trees

In this section, we present a polynomial dynamic programming algorithm which tests whether an n-vertex directed tree T admits an UPSE into a convex point set S of size n. We first study a restricted UPSE problem that will be used later on by our main algorithm.

4.1. A restricted UPSE problem for rooted directed trees

Definition 1. In a *restricted UPSE* problem for trees we are given a directed tree T rooted at r, a convex point set S, and a point $p_r \in S$. We are asked to decide whether there exists an UPSE of T into S such that (i) the root r of T is mapped to point p_r and, (ii) each subtree of T is mapped to consecutive points on the same side (either L or R) of S.

The next observation follows directly from the definition of a restricted UPSE.

Observation 1. In a restricted UPSE of a directed tree T rooted at r into a convex point set S, where the root r of T is mapped to point $p_r \in S$, no edge enters the triangles $\triangle(t(L), t(R), p_r)$ and $\triangle(b(L), b(R), p_r)$.

Fig. 7.a shows a tree T rooted at vertex r, a convex point set S consisting of a left-sided convex point set L and a right-sided convex point set R. Tree T has a restricted UPSE only if its root r is mapped to point $p_r \in L$ (Fig. 7.b). Mapping r to any other point $p \in S$ makes it impossible to map each subtree of T to consecutive points on the same side of S.

Without loss of generality, consider the case when vertex r is mapped to some specified point of S on its right side (see Fig. 8). In the restricted UPSE problem we consider in this section, each of the subtrees of T is mapped to one of the sides of S, thus some subtrees of T are mapped to the left side of S. The ordering in which the subtrees appear on the right side is not important. Thus, we just have to make sure that those of the subtrees $T^l_1, \ldots, T^l_{d^-(r)}$ which are mapped to the right side of S, lie below r, and those of the subtrees $T^h_1, \ldots, T^h_{d^+(r)}$ which are mapped to the right side, lie above r. For the subtrees of T, which are mapped to the left side, the order in which they appear may be essential for the upwardness of the embedding. Thus, among all $T^l_1, \ldots, T^l_{d^-(r)}$ (resp. $T^h_1, \ldots, T^h_{d^+(r)}$) which are mapped to the left side of S, let T^l_i (resp. T^h_j) be the one with the maximum number of nodes in *upper*(T^l_i) (resp. *lower*(T^h_j)). Intuitively, if the edge connecting T^l_i (resp. T^h_j) to r is not upward when T^l_i appear as the highest (resp. lowest) among all the subtrees $T^l_1, \ldots, T^l_{d^-(r)}$ (resp. $T^h_1, \ldots, T^h_{d^+(r)}$) mapped to the left side, then this edge will not be upward if any other of subtrees $T^l_1, \ldots, T^l_{d^-(r)}$ (resp. $T^h_1, \ldots, T^h_{d^+(r)}$), mapped to the left side, appears at place of T^l_i (resp. T^h_j). In the following paragraphs and Lemma 5 we formalize this idea.

Let T be a directed tree rooted at vertex r and let $\lambda = (T_1, \ldots, T_{d(r)})$ be an ordering of the subtrees of T. Let S be a convex point set and let Γ be an UPSE of T into S. We say that UPSE Γ *respects ordering* λ if for any two subtrees T_i

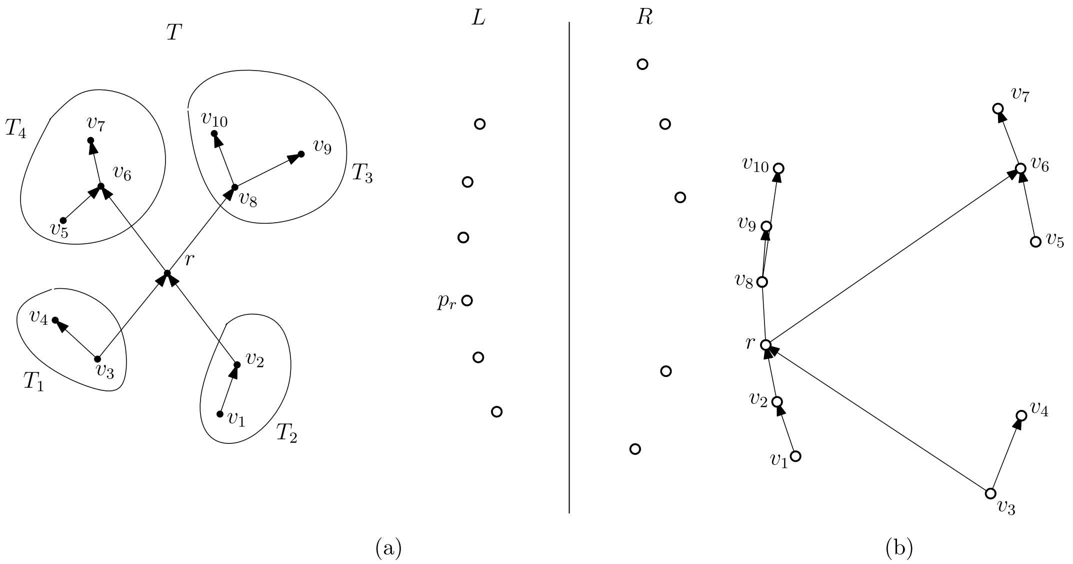

Fig. 7. (a) A tree T rooted at vertex r and a convex point set $S = L \cup R$. (b) A restricted UPSE of T into S so that r is mapped to point p_r. No restricted UPSE of T exists when r is mapped to any point other than p_r.

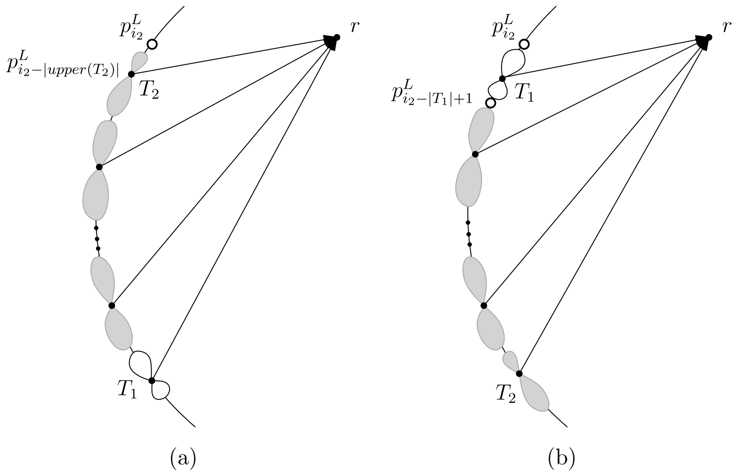

Fig. 8. The configuration of subtrees used in the proof of Lemma 5. (a) The drawing of subtrees T_1 and T_2 does not respect proper ordering λ. (b) Subtrees T_1 and T_2, as well as the subtrees placed between them, have been redrawn so that proper ordering λ is respected (as far as T_1 and T_2 are concerned).

and T_j, $1 \leqslant i \leqslant j \leqslant d(r)$, that are both mapped on the same side of S, T_i is mapped to a point set that is entirely below the point set T_j is mapped to.

Consider a tree T rooted at vertex r and let $\lambda = (T^l_1, \ldots, T^l_{d^-(r)}, T^h_1, \ldots, T^h_{d^+(r)})$ be an ordering of the subtrees of T. Ordering λ is called a *proper ordering* of the subtrees of T if it satisfies the following properties:

(i) $|upper(T^l_i)| \leqslant |upper(T^l_j)|$, $1 \leqslant i \leqslant j \leqslant d^-(r)$, and
(ii) $|lower(T^h_i)| \geqslant |lower(T^h_j)|$, $1 \leqslant i \leqslant j \leqslant d^+(r)$.

For example, ordering $\lambda_1 = (T_2, T_1, T_4, T_3)$ is a proper ordering of the subtrees of T in Fig. 7.a since $|upper(T_2)| < |upper(T_1)|$ and $|lower(T_4)| > |lower(T_3)|$ while ordering $\lambda_2 = (T_1, T_2, T_3, T_4)$ is not. Observe that in a proper ordering λ of T, the subtrees in the lower subtree of T appear before the subtrees in the upper subtree of T.

Lemma 5. *Let T be an n-vertex directed tree rooted at vertex r, λ be a proper ordering of the subtrees of T, and S be a convex point set of size n. Then, if there exists a restricted UPSE of T into S, there also exists a restricted UPSE of T into S that respects λ.*

Proof. Consider a restricted UPSE Γ of T into S and assume that it does not respect ordering λ. Consider any two subtrees T_1 and T_2 of T that are mapped on the same side of S, say both are drawn on the left side of S and T_1 is drawn below T_2, and assume that they appear in reverse order in λ.

First observe that both T_1 and T_2 belong either to the lower or to the upper subtree of T. If they do not, and since they do not respect ordering λ, T_1 belongs to the upper subtree of T and T_2 in the lower subtree of T. Then, one of the edges $(r, r(T_1))$, $(r(T_2), r)$ is not upward in Γ, which is a contradiction since we assumed that Γ is an UPSE of T into S.

Without loss of generality assume that both T_1 and T_2 belong to the lower subtree of T (the proof where they both belong to the upper subtree of T is symmetric). Let the highest point of T_2 be mapped to the i_2-th lowest point on the left side of S, i.e., point $p^L_{i_2}$ (see Fig. 8.a). Consider the drawing Γ' obtained from Γ by shifting downwards by $|T_1|$ points the drawing of subtree T_2 and of all the subtrees drawn between T_1 and T_2 in Γ, and by drawing T_1 (as it was drawn in Γ) at the $|T_1|$ points $\{p^L_{i_2} \dots p^L_{i_2-|T_1|+1}\}$ (see Fig. 8.b). The resulting drawing Γ' is obviously planar. In order to prove that Γ' is a restricted UPSE it is sufficient to prove that both edges $(r(T_1), r)$ and $(r(T_2), r)$ remain upward. Edge $(r(T_2), r)$ obviously remains upward since vertex $r(T_2)$ is mapped to a point in Γ' lower than the point it was mapped in Γ. The root $r(T_2)$ of subtree T_2 was mapped to point $p^L_{i_2-|upper(T_2)|}$ in Γ. Since Γ does not respect the proper ordering λ, it holds that T_2 appears before T_1 in λ and, thus, $|upper(T_2)| \leqslant |upper(T_1)|$. So, in Γ' vertex $r(T_1)$ is mapped to a point that is at or below the one vertex $r(T_2)$ was mapped in Γ. We conclude that edge $(r(T_1), r)$ is upward in Γ' and, thus, Γ' is a restricted UPSE.

By repeatedly identifying pairs of subtrees that cause a restricted UPSE drawing not to respect λ and by transforming the drawing as described above, we can obtain a restricted UPSE drawing for tree T on S that respects the proper ordering λ of the subtrees of T. □

Theorem 4. *Let T be an n-vertex directed tree rooted at vertex r, L and R be left-sided and right-sided convex point sets, resp., such that $S = L \cup R$ is a convex point set of size n, and p_r a point of S. The restricted UPSE problem with input T, S and p_r can be decided in $O(d(r)n)$ time. Moreover, if a restricted UPSE for T, S and p_r exists, it can also be constructed in $O(d(r)n)$ time.*

Proof. Let $v_1, \dots, v_{d(r)}$ be the vertices adjacent to r and let $T_1, \dots, T_{d(r)}$ be the subtrees of T, rooted at $v_1, \dots, v_{d(r)}$, respectively. Recall that by definition of subtree T_i, it is true that $T_i = T(v_i)$, $i = 1, \dots, d(r)$. Thus, by Lemma 2, the values $|lower(T_i)|$ and $|upper(T_i)|$, $i = 1, \dots, d(r)$, can be computed in total $O(n)$ time. By applying a bucket sort, we can obtain a proper ordering $\lambda = (T_1, T_2, \dots, T_{d(r)})$ in $O(n)$ time.

By Lemma 5, it is enough to test whether there exists a restricted UPSE that respects λ. Thus, we will describe a dynamic programming algorithm that tests whether there exists a restricted UPSE on input T, L, R and p_r. We assume that point p_r belongs to the left side of S. The case where p_r belongs to the right side of S is symmetrical.

Our dynamic programming algorithm uses a two-dimensional $d(r) \times |L|$ matrix M. Value $M[i, j]$ is *TRUE* if and only if there exists a restricted UPSE of the subtree of T induced by r and $T_1, \dots, T_i$ that uses all the j lowest points of the left-sided point set L and as many consecutive points as required in the lowest part of the right-sided convex point set R. Recall that $\{u, v\}$ denotes arc (u, v) if $(u, v) \in T$ and it denotes arc (v, u) if $(v, u) \in T$, otherwise it is undefined.

For the boundary conditions of our dynamic programming we have that

$$M[0,0] = \mathit{TRUE},$$

$$M[1,j] = \begin{cases} \mathit{TRUE}, & \text{if } j = 0 \text{ and } \{r(T_1), p_r\} \text{ is upward}, \\ \mathit{TRUE}, & \text{if } j = |T_1| \text{ and } p_r \notin L_{1..|T_1|} \text{ and } \{r(T_1), p_r\} \text{ is upward}, \\ \mathit{FALSE}, & \text{otherwise}. \end{cases}$$

Let $\sigma = |T_1| + \dots + |T_i|$. $M[i, j]$, $1 < i \leqslant d(r)$ and $0 \leqslant j \leqslant |L|$, is set to *TRUE* if any of the following conditions is true; otherwise it is set to *FALSE*.

c-1: $M[i, j-1] = \mathit{TRUE}$ ***and*** $p_r = L_{j..j}$.
This is the case where point p_r happens to be the j-th point of L. There is no need to test for upwardness of $\{r(T_i), p_r\}$ since it has been already tested when entry $M[i, j-1]$ was filled in.

c-2: $M[i-1, j-|T_i|] = \mathit{TRUE}$ ***and*** $p_r \notin L_{j-|T_i|+1..j}$ ***and*** $\{r(T_i), p_r\}$ ***is upward***.
In this case, T_i is placed on L. We know that T_i fits on L since $j < |L|$, however, we must make sure that it also holds that p_r is not one of the $|T_i|$ topmost points of $L_{1..j}$.

c-3: $M[i-1, j] = \mathit{TRUE}$ ***and*** $\sigma - j \leqslant |R|$ ***and*** $\{r(T_i), p_r\}$ ***is upward***.
In this case, T_i is placed to R. Thus, we need to make sure that at least $\sigma - j$ points exist on $|R|$.

Note that since trees $T_1, \dots, T_{d(r)}$ are considered by the described algorithm in this specific order, we get that the resulting UPSE, if exists, respects the ordering λ.

When determining the value of an entry $M[i, j]$ we need to decide whether the arc $\{r(T_i), p_r\}$ is upward. In order to do that, we need to know the point to which $r(T_i)$ is mapped. Recall that T_i is mapped to $|T_i|$ consecutive points forming a one-sided convex point set. By Lemma 3, $r(T_i)$ has to be mapped to the $|lower(T_i)|$-th of these points. Also, by Lemma 2,

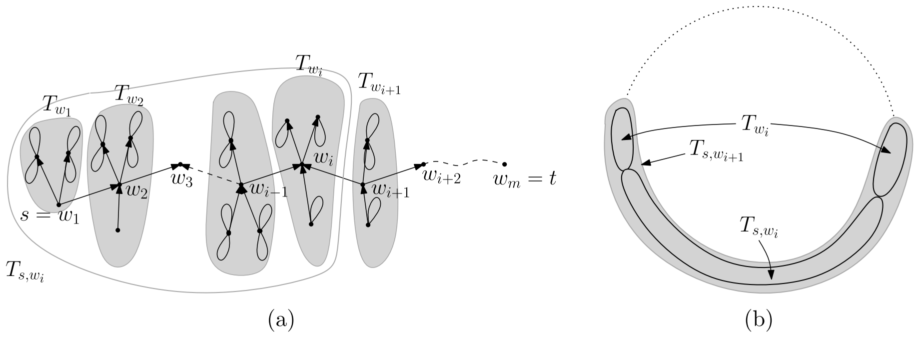

Fig. 9. (a) The decomposition of tree T based on a path between a source s and a sink t of T. (b) The structure of an UPSE of the tree T into point set S.

the values $|lower(T_i)|$, $1 \leqslant i \leqslant d(r)$ can be computed in total $O(n)$ time. The y-coordinate of the $|lower(T_i)|$-th point can be found in $O(1)$ time.

From Lemma 5, it follows that entry $M[d(r), |L|] = TRUE$ if and only if there exists a restricted UPSE of T into $L \cup R$ such that $r(T)$ is mapped to p_r.

Each entry of matrix M can be filled in $O(1)$ time. Thus, all entries of matrix M are filled in $O(n + d(r)|L|)$ time. In the event that a restricted UPSE of T into $L \cup R$ exists, we can construct it by storing in each entry $M[i, j]$ the side ("L" or "R") in which T_i was placed. This information, together with the fact that the restricted UPSE respects ordering λ is sufficient to construct the embedding. By Lemma 4, the construction of an UPSE of subtrees $T_1, \ldots, T_{d(r)}$ into corresponding one-sided point sets takes total $O(n)$ time. Thus the whole construction can be accomplished in $O(nd(r))$ time. □

We denote by $\mathcal{L}(T, L, R)$ the set of points $p \in L \cup R$ such that there exists a restricted UPSE of T into point set $L \cup R$ where the root of tree T is mapped to point p. The next theorem follows easily from Theorem 4 by testing each point of $L \cup R$ as a candidate host for $r(T)$.

Theorem 5. *Let T be an n-vertex directed tree rooted at vertex r and L and R be left-sided and right-sided convex point sets, resp., such that $S = L \cup R$ is a convex point set of size n. Then, the set $\mathcal{L}(T, L, R)$ can be computed in $O(d(r)n^2)$ time.*

Note. In this paper we only consider embeddings of n-vertex trees into point sets of size n. Thus, by definition $\mathcal{L}(T, L, R)$ is empty when $|T| \neq |L| + |R|$.

4.2. The testing algorithm

Let T be a directed tree and let S be a convex point set. In any UPSE of T into S, a source s and a sink t of T will be mapped to points $b(S)$ and $t(S)$, respectively. In this section, we present a dynamic programming algorithm that, given an n-vertex directed tree T, a source s and a sink t of T, and a convex point set S of size n, decides in polynomial time whether T has an UPSE into S so that s and t are mapped to $b(S)$ and $t(S)$, respectively. The application of this algorithm to all $\langle source, sink \rangle$ pairs of T, yields a polynomial time algorithm for deciding whether T has an UPSE into S.

Let s and t be a source and a sink vertex of T, respectively. Denote by $P_{s,t} = \{s = w_1, w_2, \ldots, w_m = t\}$ the (undirected) path connecting s and t in T, see Fig. 9.a. By T_{s,w_i}, $1 \leqslant i < m$, we denote the subtree of T that contains source s and is formed by the removal of edge $\{w_i, w_{i+1}\}$. By definition, $T_{s,w_m} = T$. Let $T_{w_i} = T_{s,w_i} \setminus T_{s,w_{i-1}}$, $1 < i \leqslant m$. By definition, $T_{w_1} = T_{s,w_1}$. By Lemma 1, we know that T_{s,w_i} is drawn on consecutive points of S, call this point set S_i (see also Fig. 9.b). Since by assumption s is mapped to $b(S)$, we infer that $b(S) \in S_i$. Similarly, in any UPSE of T into S, tree $T_{s,w_{i+1}}$ is also drawn on consecutive points of S that contain $b(S)$, call this point set S_{i+1}. Hence, tree $T_{w_{i+1}}$ is drawn on a set $S_{w_{i+1}} = S_{i+1} \setminus S_i$, that is, a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side.

Our dynamic programming algorithm maintains a list of points $\mathcal{P}(a, b, k)$, $0 \leqslant a \leqslant |L|$, $0 \leqslant b \leqslant |R|$, $1 \leqslant k \leqslant m$, such that

$$p \in \mathcal{P}(a, b, k) \iff \begin{cases} T_{s,w_k} \text{ has an UPSE into point set } S_{1..a,1..b} \text{ with} \\ \text{vertex } w_k \text{ mapped to point } p. \end{cases}$$

For the boundary conditions of our dynamic programming we have that

$$\mathcal{P}(a, b, 1) = \mathcal{L}(T_{w_1}, L_{1..a}, R_{1..b}) \cap \{b(S)\} \quad \text{where } a + b = |T_{w_1}|.$$

Since $w_1 = s$, and since by assumption vertex s is mapped to point $b(S)$, we insist that $\mathcal{P}(a, b, 1)$ is either $\{b(s)\}$ or $\emptyset$.

Assume that $T_{s,w_{i-1}}$ has an UPSE into some consecutive points of S, say $S_{1..a-a_1,1..b-b_1}$, with vertices s and w_{i-1} mapped to points $b(S)$ and q, respectively. Assume also that T_{w_i} has a restricted UPSE into $L_{a-a_1+1..a} \cup R_{b-b_1+1..b}$ with vertex w_i

Algorithm 1: TREE-UPSE(T, S, s, t)

input : A directed tree T, a point set S, a source s and a sink t of T. Path $(s = w_1, \ldots, w_m = t)$ is used to progressively build tree T from subtrees T_{w_i}, $1 \leqslant i \leqslant m$.
output : "YES" if T has an UPSE into S with s mapped to $b(S)$ and t mapped to $t(S)$, "NO" otherwise.

For $a = 0 \ldots |L|$
  For $b = 0 \ldots |R|$
    $\mathcal{P}(a, b, 1) = \mathcal{L}(T_{w_1}, L_{1..a}, R_{1..b}) \cap \{b(S)\}$
    For $k = 2 \ldots m$ //Consider tree T_{w_k}
      $\mathcal{P}(a, b, k) = \emptyset$
      For $i = 0 \ldots |T_{w_k}|$ //We consider the case where i vertices of T_{w_k} are placed to the left side of S
        if $(a - i \geqslant 0)$ **and** $(b - (|T_{w_k}| - i) \geqslant 0)$
          Let $\mathcal{L} = \mathcal{L}(T_{w_k}, L_{a-i+1..a}, R_{b-(|T_{w_k}|-i)+1..b})$
          //We consider all possible placements of w_{k-1}
          For each q in $\mathcal{P}(a - i, b - (|T_{w_k}| - i), k - 1)$
            //We consider all the possible placements of w_k
            For each p in $\mathcal{L}$
              if $(\{w_{i-1}, w_i\}$ drawn on line-segment (q, p) is upward)
                then add p to $\mathcal{P}(a, b, k)$.
if $t(S) \in \mathcal{P}(|L|, |R|, m)$ **then return** ("YES");
return ("NO");

mapped to point p. If arc $\{w_{i-1}, w_i\}$ which is drawn as line-segment (q, p) is upward, then we can combine the UPSE for $T_{s,w_{i-1}}$ with the restricted UPSE of T_{w_i} in order to obtain an UPSE of T_{s,w_i} into point set $S_{1..a,1..b}$. This idea which allows us to add points in $\mathcal{P}(a, b, i)$ is formally described by the following recurrence relation.

For any $1 < i \leqslant m$ we set

$$\begin{aligned}\mathcal{P}(a, b, i) = \big\{p \bigm| {} & \exists a_1, b_1 \in \mathbb{N} \cup \{0\}\colon a_1 + b_1 = |T_{w_i}| \\ & \textbf{and } p \in \mathcal{L}(T_{w_i}, L_{a-a_1+1..a}, R_{b-b_1+1..b}) \\ & \textbf{and } \exists q \in \mathcal{P}(a - a_1, b - b_1, i - 1) \\ & \textbf{and } \{p, q\} \text{ is upward}\big\}.\end{aligned} \tag{1}$$

Lemma 6. *Let T be an n-vertex rooted directed tree, S be a convex point set of size n, s and t be a source and a sink of T, respectively. Let $P_{s,t} = \{s = w_1, w_2, \ldots, w_m = t\}$ be the path connecting s and t in T and let T_{s,w_i}, $1 \leqslant i \leqslant m$ be the subtrees of T, as defined above. There exists an UPSE of T into S such that vertices s and t are mapped to points $b(S)$ and $t(S)$, respectively, if and only if $\mathcal{P}(|L|, |R|, m)$, computed by recurrence relation (1), contains point $t(S)$.*

Proof. We first prove that if $p \in \mathcal{P}(a, b, i)$ then T_{s,w_i} has an UPSE into point set $S_{1..a,1..b}$ with vertex w_i mapped to point p. From the boundary conditions it holds for $i = 1$. Assume that if $q \in \mathcal{P}(a - a_1, b - b_1, i - 1)$ then $T_{s,w_{i-1}}$ has an UPSE into $S_{1..a-a_1,1..b-b_1}$ with vertex w_{i-1} mapped to point q. Let now $p \in \mathcal{P}(a, b, i)$. Then by the definition of the recurrence relation we infer that: (1) there exist $a_1, b_1 \in \mathbb{N} \cup \{0\}$ so that $a_1 + b_1 = |T_{w_i}|$, (2) $p \in \mathcal{L}(T_{w_i}, L_{a-a_1+1..a}, R_{b-b_1+1..b})$, which, by definition of $\mathcal{L}$, means that there exists a restricted UPSE of T_{w_i} into $L_{a-a_1+1..a}, R_{b-b_1+1..b}$ with w_i mapped to p, (3) $\exists q \in \mathcal{P}(a - a_1, b - b_1, i - 1)$, and thus, by the induction hypothesis, $T_{s,w_{i-1}}$ has an UPSE into $S_{1..a-a_1,1..b-b_1}$, and, finally, (4) edge $\{p, q\}$ is upward. Then we combine the UPSE for $T_{s,w_{i-1}}$ with the restricted UPSE for T_{w_i} and the edge $\{p, q\}$ in order to get an UPSE of T_{s,w_i} into point set $S_{1..a,1..b}$. By Observation 1, we have that edge (p, q) does not create any crossing with the tree T_{w_i} and, therefore, the combined drawing is planar.

For the reversed statement we also work by induction. From the boundary conditions we know that if $T_{s,w_1} = T_{w_1}$ has an UPSE into a point set $S_{1..a,1..b}$ then $b(S) \in \mathcal{P}(a, b, 1)$, where $a + b = |T_{w_1}|$. Assume that the statement is true for $T_{s,w_{i-1}}$, i.e., if $T_{s,w_{i-1}}$ has an UPSE into a point set $S_{1..a,1..b}$ with vertex w_{i-1} mapped to q then $q \in \mathcal{P}(a, b, i - 1)$. Assume also that T_{s,w_i} has an UPSE into a point set $S_{1..a,1..b}$ with vertices s and w_i mapped to points $b(S)$ and p, respectively. By the discussion above we know that in every such embedding $T_{s,w_{i-1}}$ is mapped to consecutive points of $S_{1..a,1..b}$ that contains $b(S)$. Therefore there exist two numbers a_1 and b_1, so that $a_1 + b_1 = |T_{w_i}|$ and subtree T_{w_i} is mapped to the point set $S_{a-a_1+1..a,b-b_1+1..b}$, with vertex w_i mapped to some point p, $p \in S_{a-a_1+1..a,b-b_1+1..b}$. Moreover, by the induction hypothesis, there exists $q \in \mathcal{P}(a - a_1, b - b_1, i - 1)$. So, since the edge connecting p and q is upward, by the definition of recurrence relation we infer that $p \in \mathcal{P}(a, b, i)$.

For $i = m$ we infer that an UPSE of T into S such that source s and sink t are mapped to $b(S)$ and $t(S)$, respectively, exists if and only if $t(S) \in \mathcal{P}(|L|, |R|, m)$. □

Values $\mathcal{P}(a, b, k)$, when $0 \leqslant a \leqslant |L|$, $0 \leqslant b \leqslant |R|$, $1 \leqslant k \leqslant m$ are calculated by Algorithm 1.

Theorem 6. *Let T be an n-vertex rooted directed tree, S be a convex point set of size n, s be a source of T and t be a sink of T. It can be decided in time $O(n^5)$ whether T has an UPSE into S such that s is mapped to $b(S)$ and t is mapped to $t(S)$. Moreover, if such an UPSE exists, it can also be constructed within the same time bound.*

Proof. Algorithm 1 calculates the values $\mathcal{P}(a,b,k)$, when $0 \leqslant a \leqslant |L|$, $0 \leqslant b \leqslant |R|$, $1 \leqslant k \leqslant m$. Thus, by Lemma 6 we infer that Algorithm 1 decides whether T has an UPSE into S such that s and t are mapped to $b(S)$ and $t(S)$, respectively.

A naive analysis of Algorithm 1 yields an $O(n^7)$ time complexity. The analysis assumes that (i) the left and the right side of S have both size $O(n)$, (ii) the path from s to t has length $O(n)$, (iii) each tree T_{w_i} has size $O(n)$ and (iv) each $\mathcal{L}$-list containing the solution of a restricted UPSE problem is computed in $O(n^3)$ time. However, based on the following two observations, the total time complexity can be reduced to $O(n^5)$.

- A factor of n can be saved by realizing that in our dynamic programming we can maintain a list $\mathcal{P}'(a,i)$ which uses only one parameter for the left side of the convex set (in contrast with $\mathcal{P}(a,b,i)$ which uses a parameter for each side of S). The number of points on the right side of S is implied since the size of each tree T_{s,w_i} is fixed. For simplicity, we have decided to use notation $\mathcal{P}(a,b,i)$.
- Another factor of n can be saved by observing that the computation of list $\mathcal{L}$ is actually $O(d(w_i)n^2)$. Thus, summing over all i gives $O(n^3)$ in total, and not $O(n^4)$.

The UPSE of T into S can be recovered easily by modifying Algorithm 1 so that it stores for each point $p \in \mathcal{P}(a,b,k)$ the point q where vertex w_{i-1} is mapped to as well as the point set that hosts tree $T_{s,w_{i-1}}$ (i.e., its top points on the left and the right side of S). □

By applying Algorithm 1 on all $\langle source, sink\rangle$ pairs of T we can decide whether tree T has an UPSE on a convex point set S, as the main next theorem indicates.

Theorem 7. *Let T be an n-vertex rooted directed tree and S be a convex point set of size n. It can be decided in time $O(n^6)$ whether T has an UPSE into S. Moreover, if such an UPSE exists, it can also be constructed within the same time bound.*

Proof. Note that a naive application of the idea leads to the algorithm with time complexity $O(n^7)$, since there are $O(n^2)$ distinct pairs of sources and sinks. Next we explain how the overall time complexity can be reduced to $O(n^6)$. Let $P_{s,t}$ be a path from s to t, passing through m vertices, and let t' be the j-th vertex of $P_{s,t}$ that is also a sink of G. During the computation of $\mathcal{P}(a,b,m)$ corresponding to path $P_{s,t}$ we also compute $\mathcal{P}(a,b,j)$ and thus we can immediately answer whether there exists an UPSE of G into S so that s and t' are mapped to $b(S)$ and $t(S)$, respectively. Next consider a sink $\tilde{t}$ that does not belong to path $P_{s,t}$. Consider the path $P_{s,\tilde{t}}$. Assume that the last common vertex of $P_{s,t}$ and $P_{s,\tilde{t}}$ is the j-th vertex of $P_{s,t}$. In order to compute whether there exists an UPSE of G into S so that s and $\tilde{t}$ are mapped to $b(S)$ and $t(S)$, respectively, we can start the computations of Algorithm 1 determined by variable k from the $j+1$-th step (see line 4 of the algorithm). Thus, for a single source s and all possible sinks variable k changes at most n times. Since the number of different sources is $O(n)$ we conclude that the whole algorithm runs in time $O(n^6)$. □

5. Generalization of the decision algorithm to outerplanar digraphs

In this section we extend our decision algorithm for directed trees to the class of outerplanar digraphs. For better understanding, we keep the new definitions and the line of arguments on outerplanar digraphs as close as possible to already existing definitions and arguments on trees.

5.1. Definitions and preliminary results

Let G be acyclic outerplanar digraph (*outerplanar DAG* for short). We say that G is a *rooted outerplanar DAG* if one of its vertices, denoted by $r(G)$, is designated as its *root*. We then also say that G *is rooted at* vertex $r(G)$. A maximal biconnected subgraph of G is called a *block* of G.

5.1.1. Vertices

Switch: Recall that a vertex of G which is either a source or a sink is referred to as a *switch*.
Side vertex: Let B be a block of G. A vertex of B which is not a switch of B is called a *side vertex* of B.
Cut vertex: A *cut vertex* is any vertex of G whose removal increases the number of connected components.
(s,t)-Separating vertex: Given two vertices s and t of G, a cut vertex c of G is called (s,t)-*separating* if the removal of c leaves s and t in different connected components.

5.1.2. Split at a cut vertex

Let c be a cut vertex of G and let $G_1,\dots,G_k$ be the maximal connected components obtained after deletion of c and all the edges incident to it. Let G'_i be the digraph induced by the vertices of G_i and vertex c, $i=1,\dots,k$. The set of subgraphs $G'_1,\dots,G'_k$ of G is called a *split at vertex c*.

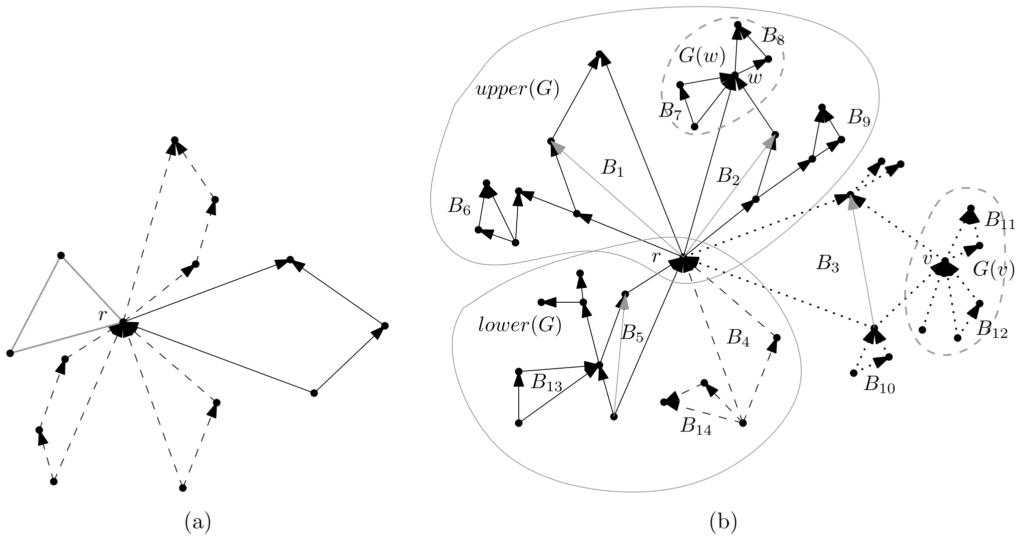

Fig. 10. (a) An example of extremal and side blocks. The extremal blocks are drawn by dashed lines. The rest are the side blocks. (b) An outerplanar DAG G. One extremal and one side subgraph of G are drawn by dashed and dotted lines, respectively. The lower and the upper subgraphs of G are surrounded by solid gray curves. The subgraphs $G(v)$ and $G(w)$ of graph G are surrounded by gray dashed curves.

5.1.3. Blocks

Let B be a block of G. Binucci et al. [14] proved the following lemma.

Lemma 7. *(See Binucci et al. [14].) Let G be an n-vertex DAG containing a k-vertex cycle-DAG C, for some $k \leqslant n$. Suppose that C has at least two vertices u and v that are sources in C. Then there exists a convex point set S of size n such that G has no upward straight-line embedding into S.*

This result trivially extends to an arbitrary convex point set:

Lemma 8. *Let G be an n-vertex outerplanar DAG, let B be a block of G and S be a convex point set of size n. If B contains a cycle-DAG C which has either two sources or two sinks then G does not admit an UPSE into S.*

By Lemma 8, we are only interested in digraphs so that the boundary of each block contains exactly one source and one sink. Note also that if there exists an UPSE of an outerplanar DAG G into a convex point set S, then it is outerplane and thus in the following we consider only outerplane embeddings of G. Let B be a block of G and let s be the source and t be the sink of the boundary of B.

Sides of a block and a digraph: An outerplane embedding of block B is bounded by two vertex-disjoint paths connecting t and s, called *sides* of B. By $L(B)$ and $R(B)$ we denote the side vertices of two different sides of block B. When block B of graph G is specified, by $L(G, B)$ (resp. $R(G, B)$) we denote all the vertices of $L(B)$ (resp. $R(B)$) plus the vertices of the components incident to the vertices of $L(B)$ (resp. $R(B)$) and refer to $L(G, B)$ and $B(G, B)$ as to the *sides* of G with respect to B.

One-sided/two-sided block: If one of the sides of a block B is a single edge, then B is called a *one-sided block*, otherwise it is called a *two-sided block*.

Trivial block: The edges of G that do not belong to any biconnected component are called the *trivial blocks* of G. Thus the two end vertices of each of such edges are a source and a sink of the corresponding trivial block.

Extremal/side block: Let v be a cut vertex of G. The blocks of G that have v as their switch are called *extremal blocks* of v, the remaining blocks incident to v are called *side blocks* of v (see Fig. 10.a). By $b^-(v)$ (resp., $b^+(v)$) we denote the number of extremal blocks of v that have v as their sink (resp., source). By $b(v)$ we denote the total number of extremal blocks that contain v, i.e., $b(v) = b^-(v) + b^+(v)$.

5.1.4. Subgraphs

Extremal subgraph: Let G be a rooted digraph having cut vertex r as its root. Let $G_1^l, \ldots, G_{b^-(r)}^l$, and $G_1^h, \ldots, G_{b^+(r)}^h$ be the rooted subgraphs of G obtained by a split at r and having r as their sink or source, respectively. They are called the *extremal subgraphs of* G (see for example the dashed subgraph in Fig. 10.b). Note that the superscripts "l" and "h" indicate whether a particular subgraph of G has r as its sink or as its source, respectively.

Side subgraph: Let G be a rooted digraph having cut vertex r as its root. A subgraph of G, obtained by a split at r and having r as its side vertex is called a *side subgraph* of G (see for example the dotted subgraph in Fig. 10.b).

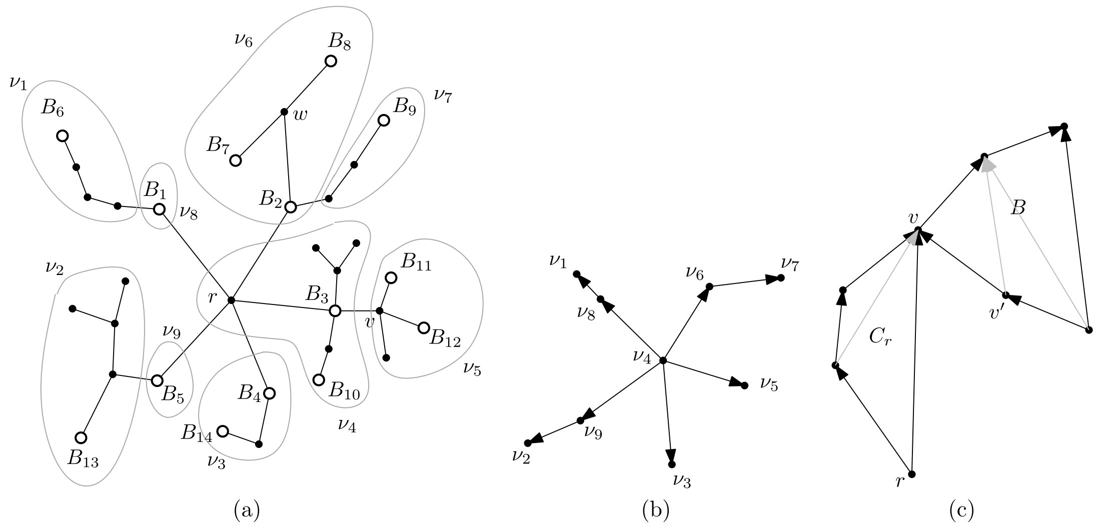

Fig. 11. (a) BC-tree of the graph depicted in Fig. 10.b. The subtrees surrounded by gray curves are merged to single vertices during the construction of auxiliary tree. (b) Auxiliary tree of the graph depicted in Fig. 10.b, constructed from the BC-tree of Fig. 11.a. (c) Proof of Lemma 9.

Lower/upper subgraph: Let G be a rooted digraph having cut vertex r as its root. The rooted subgraph of G consisting of r, together with $G^l_1, \dots, G^l_{b^-(r)}$ is called the *lower subgraph of* G. The lower subgraph of G is denoted by *lower*(G) (Fig. 10.b). Similarly, the rooted subgraph of G consisting of G's root, r, together with $G^h_1, \dots, G^h_{b^+(r)}$ is called the *upper subgraph of* G and is also rooted at r. The upper subgraph of G is denoted by *upper*(G).

Subgraph $G(v)$: Consider a block of G that contains the root r of G. Let v be a vertex of this block, different from r, which represents a cut vertex of G. Consider a split at v. By $G(v)$ we denote the union of the connected components that do not contain r (Fig. 10.b). Subgraph $G(v)$ is considered to be rooted at v.

5.1.5. Helpful data structures

Block-cut vertex tree: The *block-cut vertex tree*, or *BC-tree*, of a connected graph G is a tree with a node for each block and each cut vertex of G. Edges in the BC-tree connect each node ν, corresponding to block B_ν, to all the BC-tree nodes associated with the cut vertices in the block B_ν (refer to Fig. 11.a).

Auxiliary tree: Let G be a connected DAG. Assume that every cut vertex of G which is shared by two blocks is extremal for at least one of them. Let $\mathcal{T}(G)$ be the BC-tree of G. By exploring $\mathcal{T}(G)$, we construct an *auxiliary tree* of G and denote it by $\mathcal{T}'(G)$. Tree $\mathcal{T}'(G)$ contains a node μ for each connected subtree T of $\mathcal{T}(G)$ which is maximal with respect to the following property: "a cut vertex c that belongs to T and is shared by two blocks of G is a switch vertex for both of them". An edge of $\mathcal{T}'(G)$, directed from μ to ν, corresponds to a cut vertex which is a side vertex for a block associated with μ and a switch vertex for a block associated with ν. Since $\mathcal{T}'(G)$ is constructed from tree $\mathcal{T}(G)$ by contracting subtrees to vertices, we infer that $\mathcal{T}'(G)$ is also a tree. Fig. 11.b depicts the auxiliary tree for the graph of Fig. 10.b.

5.1.6. Embeddability

Let G be an outerplanar DAG rooted at a source (a sink) r.

One-side embeddable: We say that G is *one-side embeddable* iff G admits an UPSE into a one-sided convex point set of size $|G|$ so that r is mapped to its lowest (resp. highest) point.

The next theorem, proved by Binucci et al. [14], presents necessary and sufficient conditions for an outerplanar graph to have an UPSE into any one-sided point set without a restriction that its root has to be mapped to an extremal point of the point set.

Theorem 8. *(See Binucci et al. [14].) An n-vertex connected DAG G admits an UPSE into every one-sided convex point set of size n if and only if the following conditions are satisfied*:

Condition 1: *Every block of G is one-sided.*
Condition 2: *Every cut vertex shared by two blocks is extremal for at least one of them.*
Condition 3: *Every node of $\mathcal{T}'(G)$ has at most one incoming edge.*[2]

[2] Without defining the tree $\mathcal{T}'(G)$ this condition states that: "Each block has at most one cut vertex which is a side vertex for a different block."

Next we extend this theorem to the case where it is required to map the root of the given digraph to the extremal point of the point set.

Lemma 9. *Let G be an outerplanar DAG rooted at a switch vertex r. DAG G is one-side embeddable iff the conditions of Theorem 8 hold and moreover r belongs to a vertex of $\mathcal{T}'(G)$ with no incoming edges.*

Proof. We prove the lemma only for the case when vertex r is a source, the case when r is a sink is symmetrical.

In the proof of Theorem 8 it is shown that if r is a source vertex of G and belongs to a vertex of $\mathcal{T}'(G)$ without incoming edges then G has an UPSE into any one-sided point set so that r is mapped to its lowest point. We next show that this is also a necessary condition.

Let B_r be the block of G which contains vertex r. In the tree $\mathcal{T}'(G)$ block B_r is associated with a single vertex, call it t_r. Let C_r be the set of blocks of G that are represented in the tree $\mathcal{T}'(G)$ by vertex t_r.

For the sake of contradiction, assume that vertex t_r of $\mathcal{T}'(G)$ has a positive in-degree. Then, there is a block B in G, which is incident to C_r by a vertex v, so that v is a side vertex of B. Let v' be a vertex of B, connected to v by edge (v', v) (see Fig. 11.c). Let T be the spanning tree of G that is produced from G by the removal of all its transitive edges. The application of Lemma 1 to tree T and vertex v, infer that set of blocks C_r has to be mapped to consecutive points of a convex point set. Note that C_r has to be mapped below vertex v, because otherwise the resulting drawing is not upward. Moreover, if C_r is mapped to points lower than vertex v', a crossing of C_r and B is introduced. Thus, in any UPSE of G into a one-sided point set, the vertices of C_r are mapped to the points that are between the points where vertices v and v' are mapped to, which contradicts to the assumption that G, rooted at r, is one-side embeddable. Thus the lemma follows. □

Note that the construction of the auxiliary tree $\mathcal{T}'(G)$ of a graph G, checking of Conditions 1–3, as well as checking whether r belongs to a vertex of $\mathcal{T}'(G)$ with zero in-degree can be done in linear time. Moreover, in the proof of Theorem 8 the authors, given a switch r of G with the above property, present a linear time algorithm to construct an UPSE of G into S with r at the extremal point of S [14]. Thus, we get the following lemma.

Lemma 10. *Let G be an outerplanar DAG rooted at r. It can be tested in linear time whether G is one-side embeddable. In case it is, such an UPSE can be constructed within the same time bound.*

5.2. Properties of outerplanar graphs that may admit an UPSE into a convex point set

In this section we present farther necessary conditions for the structure of an outerplanar DAG to admit an UPSE into a convex point set.

Let S be a set of n points in convex position. Let G be an n-vertex outerplanar DAG and let s and t be a source and a sink of G. Let $P_{s,t}$ be a path from s to t and let $P^c_{s,t} = (s = w_1, \dots, w_m = t)$ be the subset of vertices of $P_{s,t}$ that contains only the cut vertices which are also the (s,t)-separating vertices (see Fig. 12.a) together with s and t. We call $P^c_{s,t}$ an *(s,t)-separating subset of* $P_{s,t}$.

Path component $\boldsymbol{G_{w_i,w_{i+1}}}$***:*** Note that every two consecutive vertices w_i, w_{i+1} in $P^c_{s,t}$ belong to the same block of G, that can be also a trivial block, i.e., an edge. By $G_{w_i,w_{i+1}}$ we denote the graph that is induced by this block and all vertices connected to it by paths not passing through w_i or w_{i+1}. We assume that $G_{w_i,w_{i+1}}$ is rooted at w_i. Graphs $G_{w_1,w_2}, \dots, G_{w_{m-1},w_m}$ are called the *path components* of G *defined by* the path $P_{s,t}$. In Fig. 12.a the path components $G_{w_i,w_{i+1}}$, $i = 1, \dots, 3$, are surrounded by gray curves.

Cut vertex component $\boldsymbol{G_{w_i}}$***:*** Consider vertex w_i and the subgraph of G that contains w_i and is produced by the deletion of the edges of G_{w_{i-1},w_i} and $G_{w_i,w_{i+1}}$ that are incident to w_i. Denote this graph by G_{w_i} and call it *cut vertex component* at vertex w_i *defined by* the path $P_{s,t}$. In Fig. 12.a the cut vertex components G_{w_i}, $i = 1, \dots, 4$, are surrounded by gray curves.

Subgraph $\boldsymbol{G_{s,w_i}}$***:*** Finally, let G_{s,w_i} be the subgraph of G induced by the vertices of cut vertex components $G_{w_1}, \dots, G_{w_i}$ and path components $G_{s,w_2}, \dots, G_{w_{i-1},w_i}$ (see again Fig. 12.a, see also Fig. 14.a for the general structure). We also say that *subgraph* G_{s,w_i} *is defined by* $P_{s,t}$ and w_i. By definition, $G_{s,w_m} = G$ and $G_{w_1} = G_{s,w_1}$.

The following lemma presents various properties of G that are necessary in order to admit an UPSE into a convex point set, as well as some properties of an UPSE of G into a convex point set. To facilitate the reader, we present the proof of each property immediately after its statement.

Lemma 11. *Let G be an n-vertex outerplanar DAG, let S be a convex point set of size n and let Γ_G be an UPSE of G into S, so that a source s and a sink t of G are mapped to $b(S)$ and $t(S)$, respectively. Let $P_{s,t}$ be a path from s to t, let $P^c_{s,t} = (s = w_1, \dots, w_m = t)$ be the (s,t)-separating subset of $P_{s,t}$. Let $G_{w_i,w_{i+1}}$ be one of the path components defined by the path $P_{s,t}$. Let B_i be the block of $G_{w_i,w_{i+1}}$ that contains both w_i and w_{i+1}, and let s_i and t_i be the source and the sink of B_i, respectively. Let also G_{w_i} be the cut vertex component at cut vertex w_i and let G_{s,w_i} be the subgraph of G defined by $P_{s,t}$ and w_i. The following statements hold for G and for embedding Γ_G:*

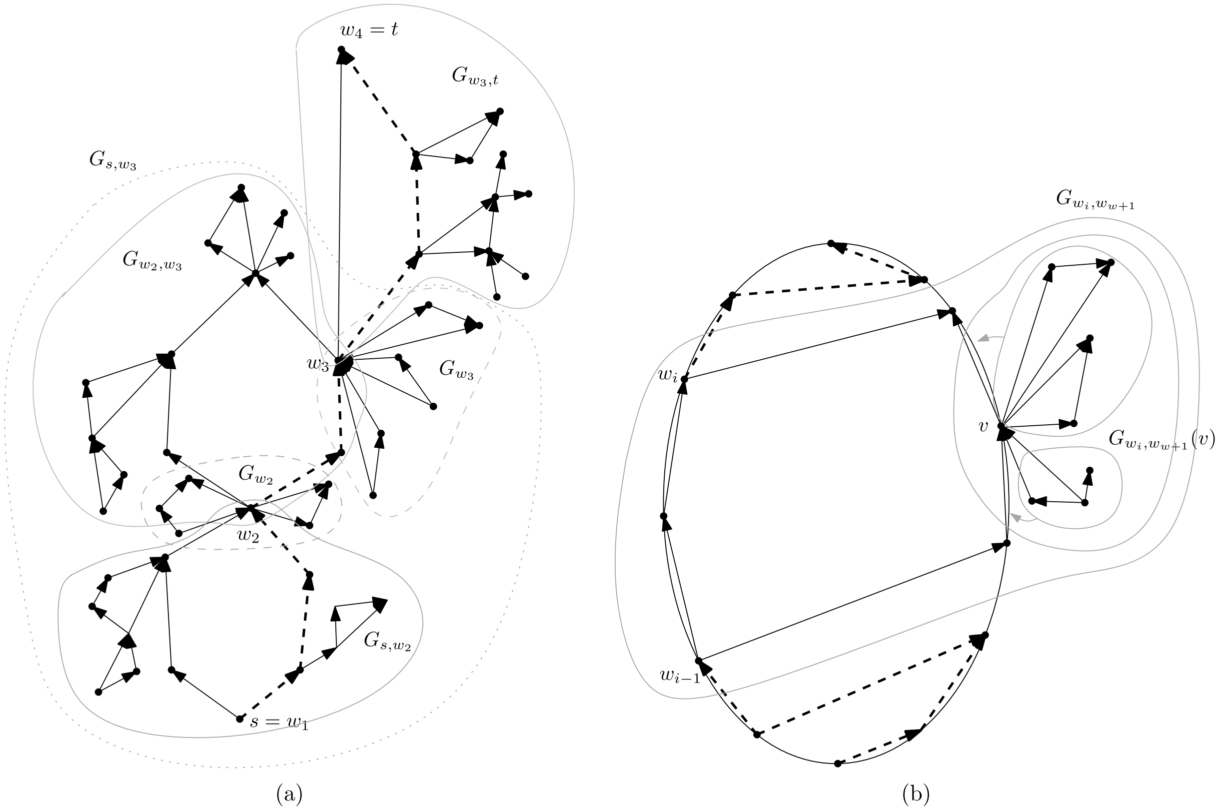

Fig. 12. (a) A path from s to t is depicted by dashed lines. The subset $P^c_{s,t} = \{s = w_1, w_2, w_3, w_4 = t\}$ contains the cut vertices that are also (s,t)-separating vertices of G. The path components $G_{w_i,w_{i+1}}$, $i = 1, \ldots, 3$ are surrounded by solid gray curves. The cut vertex components G_{w_i}, $i = 1, \ldots, 4$ are surrounded by dashed gray curves. Subgraph G_{s,w_3} defined by path $P_{s,t}$ and vertex w_3 is depicted by a dotted gray curve. (b) An UPSE of G_{w_{i-1},w_i} and of path $P_{s,t}$ on some points of S. Path $P_{s,t}$ is denoted by dashed lines.

(1) *For each vertex v of G, there exists at most one block B such that v is a side vertex of B.*

Proof. For the sake of contradiction assume that there are two blocks B_1 and B_2 so that v is a side vertex for both B_1 and B_2 (see Fig. 13.a). Since S is convex, in any UPSE of G into S, the points to which the vertices of B_1 and B_2 are mapped determine a convex point set. On the other hand, in any upward planar drawing of B_1 and B_2 so that all the vertices lie on the boundary of the drawing, vertex v has to be drawn inside the convex hull of the remaining vertices of B_1 and B_2, which is a contradiction. □

(2) *Let B' be a two-sided block of G. The vertices of $L(B')$ and $R(B')$ of block B' are mapped in Γ_G to opposite sides of S.*

Proof. Note that if B' is a two-sided block then any drawing which does not map $L(B')$ and $R(B')$ to different sides of S is either not upward or not planar. □

(3) *Vertex w_i (resp. w_{i+1}) of block B_i either coincides with the source s_i (resp. sink t_i) of B_i or is adjacent to it.*

Proof. For the sake of contradiction, assume that vertex w_i neither coincides with s_i nor is adjacent to it (the case of w_{i+1} and t_i is symmetrical) (see Figs. 13.b–c). Let a and b be the vertices adjacent to s_i. By assumption, the vertices s and t are mapped to points $b(S)$ and $t(S)$, respectively. In any UPSE Γ_G of G into S, path $P_{s,t}$ travels along the points of S from point $b(S)$ to point $t(S)$. The remaining consecutive free points of S represent one-sided point sets. By Statement (2), the vertices a and b have to be mapped to different sides of S. Thus, we get a crossing with path $P_{s,t}$ and a contradiction to the planarity of Γ_G. □

(4) *If vertex w_i (resp. vertex w_{i+1}) of block B_i is adjacent to the source s_i (resp. to the sink t_i) of B_i, then s_i (resp. t_i) is mapped in Γ_G to the side of S opposite to which w_i (resp. w_{i+1}) is mapped.*

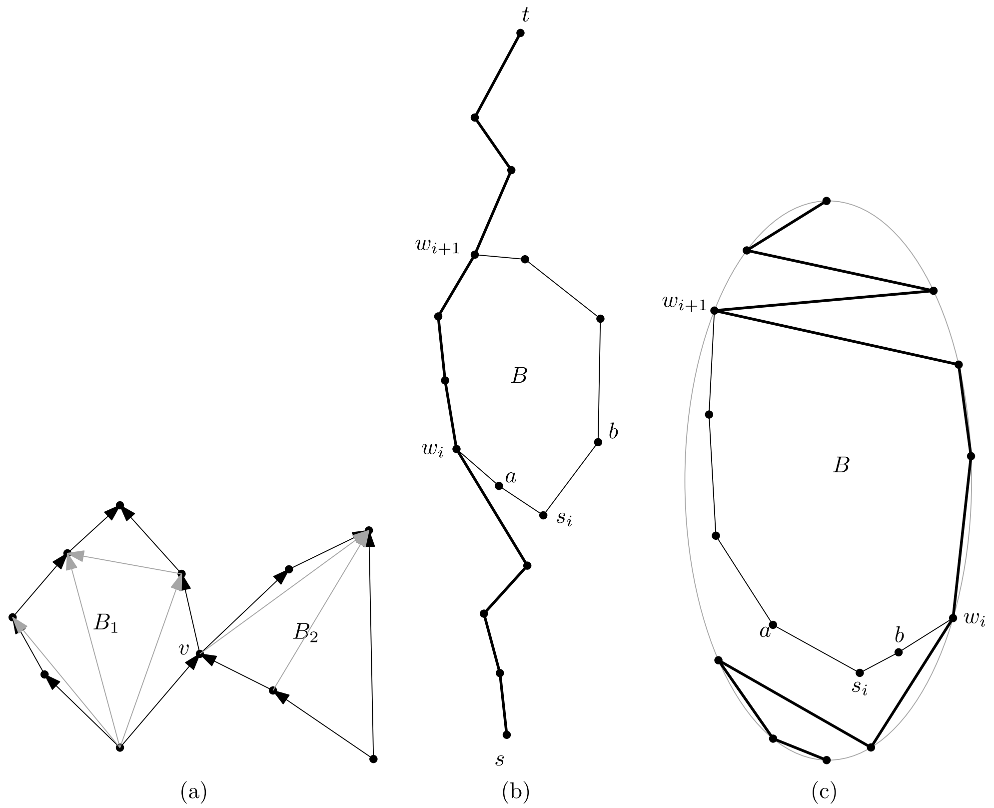

Fig. 13. Illustration for the proof of Lemma 11. (a) Statement (1). (b–c) Statement (3), path $P_{s,t}$ is denoted by black bold ink.

Proof. Consider path $P_{s,t}$ which does not contain s_i (resp. t_i) and connects w_i (resp. w_{i+1}) to $b(S)$ (resp. $t(S)$). If s_i is mapped to the same side of w_i, then path $P_{s,t}$ crosses the path on the boundary of B_i connecting s_i and t_i and not containing w_i. □

(5) *For $1 \leqslant i \leqslant m$, the vertices of G_{s,w_i} are mapped in Γ_G to consecutive points of S containing $b(S)$ (see Figs. 14.a–b).*

Proof. We first construct a spanning tree T of G with specific properties (see Fig. 14.a): For each w_i, $1 \leqslant i \leqslant m$, T contains exactly one edge of G_{w_{i-1},w_i} incident to w_i and exactly one edge of $G_{w_i,w_{i+1}}$ incident to w_i. For simplicity of presentation we call T_{s,w_i}, T_{w_i}, and $T_{w_i,w_{i+1}}$ the subtrees of T that are the spanning trees for G_{s,w_i}, G_{w_i}, and $G_{w_i,w_{i+1}}$, respectively. Call p_i and n_i the vertices of T_{w_{i-1},w_i} and $T_{w_i,w_{i+1}}$, respectively, that are adjacent to w_i. Since Γ_G is an UPSE of G, it contains an UPSE of T, say Γ_T. Consider the subtrees of T created by the removal of n_i. One of these subtrees is T_{s,w_i}. By Lemma 1, the subtree T_{s,w_i} is mapped in Γ_T on consecutive points of S, moreover, this point set contains $b(S)$, since by the requirements of the lemma, s is mapped to $b(S)$. Since, G_{s,w_i} is the subgraph of G induced by the vertices of T_{s,w_i}, we infer that the vertices of G_{s,w_i} are mapped in Γ_G to consecutive points of S containing $b(S)$. □

(6) *For $1 \leqslant i \leqslant m$, the vertices of G_{w_i} are mapped in Γ_G on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side. Moreover, for every $u, v \in G_{s,w_i}$ so that $u \notin G_{w_i}$ and $v \in G_{w_i}$ and such that, u and v are mapped to the same side of S, v is mapped higher than u (see Fig. 14.b).*

Proof. By the previous statement G_{s,w_i} is mapped to consecutive points of S containing $b(S)$. Consider spanning tree T_{s,w_i} of G_{s,w_i} as it was defined in the proof of the previous statement. Recall that p_i is the single vertex of G_{w_{i-1},w_i}, incident to w_i in T_{s,w_i}. Consider subtree T_{w_i} which is one of the subtrees created by removing vertex p_i from tree T_{s,w_i}. By Lemma 1, T_{w_i} is mapped to consecutive points of the point set where G_{s,w_i} is mapped and therefore on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side. Therefore the same holds for G_{w_i}.

By Lemma 1 and by assuming that vertex s is mapped in Γ_G to point $b(S)$, the subtree of T obtained by the removal of vertex w_i which contains s, is embedded on a set of consecutive points of S including $b(S)$. Any other vertex of G, and hence also those of G_{w_i}, are higher than the vertices of G_{s,w_i}. □

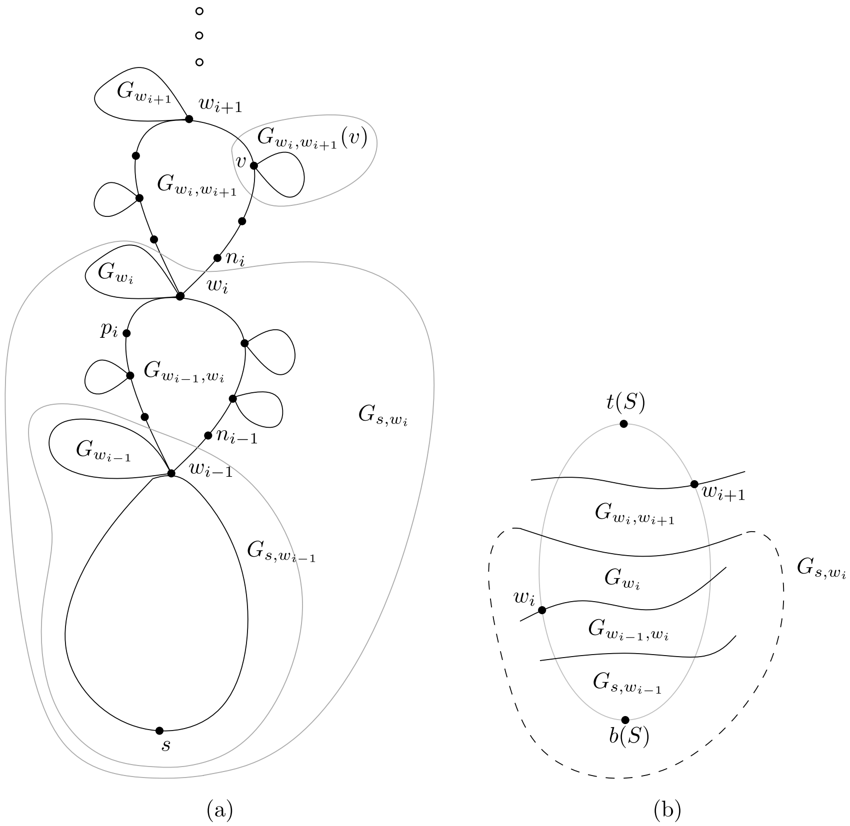

Fig. 14. (a) The structure of G, illustration of G_{s,w_i}, G_{w_i}, $G_{w_i,w_{i+1}}$, $G_{w_i,w_{i+1}}(v)$. (b) Illustration for Lemma 11, Statements (5)–(7).

(7) *For $1 \leqslant i \leqslant m-1$, the vertices of $G_{w_i,w_{i+1}}$, except for w_i and w_{i+1}, are mapped in Γ_G on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side. Moreover, for every $u \in G_{w_i}$, $v \in G_{w_i,w_{i+1}}$ such that u and v are mapped to the same side of S, v is mapped higher than u (see Fig. 14.b).*

Proof. By Statement (5), the vertices of G_{s,w_i} are mapped in Γ_G to consecutive points of S containing $b(S)$. Applying this statement for $i+1$ we get that the vertices of $G_{s,w_{i+1}}$ are mapped in Γ_G to consecutive points of S containing $b(S)$. Note that $G_{s,w_{i+1}} = G_{s,w_i} \cup (G_{w_i,w_{i+1}} \cup G_{w_{i+1}})$. Therefore we get that the vertices of the union of $G_{w_{i+1}}$ and $G_{w_i,w_{i+1}}$, except for vertex w_i, are mapped on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side. By Statement (6), the vertices of $G_{w_{i+1}}$ are also mapped to two consecutive point sets of S, one on its left and one on its right side. Moreover, all vertices of $G_{s,w_{i+1}}$ that do not belong to $G_{w_{i+1}}$ including those of $G_{w_i,w_{i+1}}$ are mapped lower than the vertices of $G_{w_{i+1}}$. Therefore, we infer that the vertices of $G_{w_i,w_{i+1}}$, except for vertices w_i and w_{i+1}, are mapped on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side. Let $u \in G_{w_i}$ and $v \in G_{w_i,w_{i+1}}$. By the definition of G_{w_i} and $G_{w_i,w_{i+1}}$ there exist vertex-disjoint paths $P_{s,u}$ and $P_{v,t}$ in G. Assuming that both u and v are mapped to the same side of S and that u is mapped higher than v, we infer that paths $P_{s,u}$ and $P_{v,t}$ have to cross, a contradiction. □

(8) *For each $v \in B_i$, $v \neq w_i$, w_{i+1}, $G_{w_i,w_{i+1}}(v)$ contains only extremal subgraphs.*

Proof. Consider an upward planar drawing of block B_i on some points of $L \cup R$. Assume for the sake of contradiction that $G_{w_i,w_{i+1}}(v)$ contains a side subgraph and let C be the side block of v that is contained in this side subgraph. If v is a side vertex of B_i then we get a contradiction by Statement (1), since v is a side vertex of both B_i and C. Otherwise, let v be a switch vertex of B_i, see Fig. 15. Let $v = t_i$, the case in which $v = s_i$ is symmetrical. Then, by Statement (3) and by fact that $v \neq w_{i+1}$, t_i is incident to w_{i+1}. By Statement (4), t_i is drawn on the side opposite to w_{i+1}. Let s_C be the source of C. In any upward planar drawing s_C is mapped to the opposite side of t_i and therefore to the same of w_{i+1}. This causes a crossing between block C and path $P_{s,t}$. Thus $G_{w_i,w_{i+1}}(v)$ contains only extremal subgraphs. □

(9) *For each $v \in B_i$, $v \neq w_i$, w_{i+1}, the vertices of $G_{w_i,w_{i+1}}(v)$ are mapped in Γ_G to the same side where v is mapped, on consecutive points around v (see Fig. 12.b).*

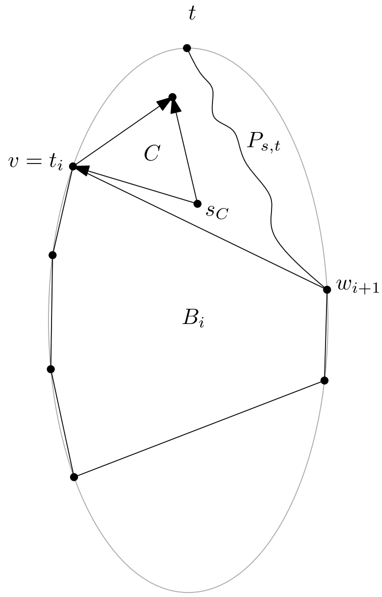

Fig. 15. Proof of Lemma 11, Statement (8).

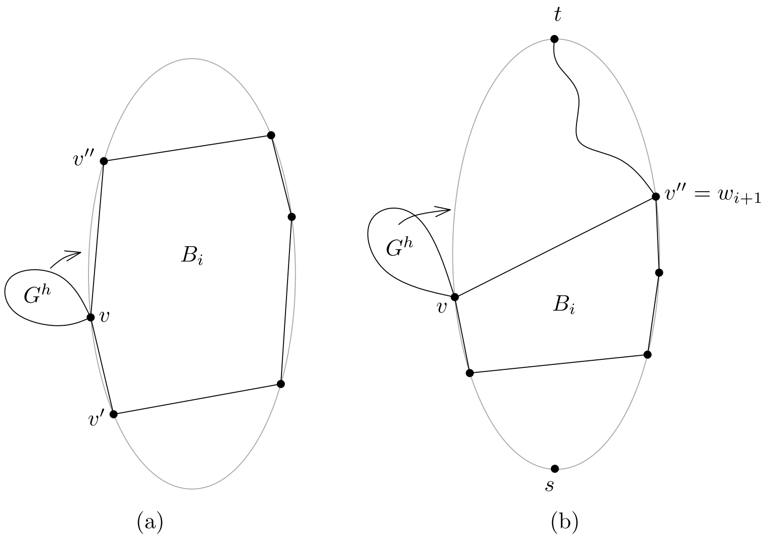

Fig. 16. (a)–(b) Proof of Lemma 11, Statement (9).

Proof. Let G^h be an extremal subgraph of $G_{w_i,w_{i+1}}(v)$, that has v as its source. The case when v is its sink is symmetrical. Let v' and v'' be two vertices of B_i, which are incident to v. Assume first that both v' and v'' are mapped to the same side of S where v is mapped (see Fig. 16.a). W.l.o.g. let v'' be mapped higher than v. Then G^h has to be drawn on the same side of S where v is drawn, below v'', otherwise a crossing among G^h and B_i is created.

Let now v'' be mapped to the other side of S (see Fig. 16.b). Since $v \neq w_i$ and $v \neq w_{i+1}$, we infer that v'' coincides either with w_i or with w_{i+1}, otherwise path $P_{s,t}$ crosses B_i. Therefore, there exists a path $P_{v'',t}$ that does not contain v. Thus, G^h has to be mapped to the same side where v is, otherwise a crossing among G^h and $P_{v'',t}$ is created. □

(10) *All the vertices of a single extremal or side subgraph of G_{w_i}, $1 \leqslant i \leqslant m$, except vertex w_i, are mapped in Γ_G to consecutive points of the same side of S. Moreover, the vertices of a side subgraph of G_{w_i} are mapped to the side of S opposite to that of w_i.*

Proof. By Statement (6), the vertices of G_{w_i} in Γ_G are mapped on a subset of S, call it S_i, comprised by two consecutive point sets of S, one on its left and one on its right side. Let T_{w_i} be the spanning tree of G_{w_i} with the following property: T_{w_i} contains only one edge incident to w_i from each of its extremal and side subgraphs. Let T_{w_i} be rooted at w_i. Let $\Gamma_{T_{w_i}}$ be the drawing of T_{w_i} on S_i that is induced by the drawing of G_{w_i} on S_i. By Lemma 1, each subtree of T_{w_i} is drawn on

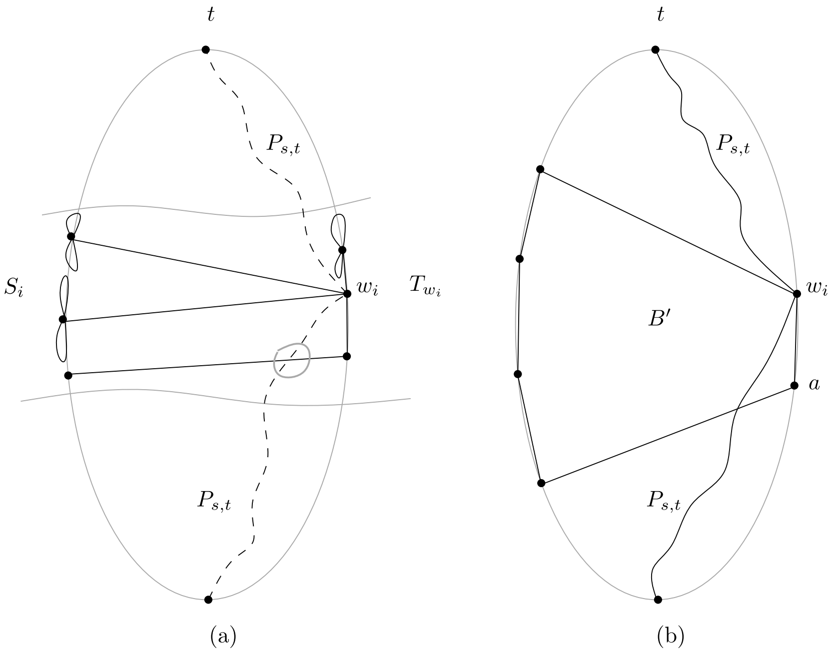

Fig. 17. (a) Proof of Lemma 11, Statement (10). (b) Proof of Lemma 11, Statement (11).

consecutive points of S_i. Moreover each of the subtrees of T_{w_i} should be drawn on the same side of S_i, otherwise a crossing with path $P_{s,t}$, passing through w_i, is produced (see Fig. 17.a). Hence the same is true for the extremal and side subgraph of G_{w_i}, except for vertex w_i.

Let now, $G^c_{w_i}$ be a side subgraph of G_{w_i} and B be the block of $G^c_{w_i}$ attached to w_i. I.e., B is a side block of w_i. Note that all vertices of B have to be mapped to the side of S opposite to w_i, otherwise we infer that either there is a crossing or a common vertex with $P_{s,t}$, which is impossible due to the planarity of Γ_G and the definition of G_{w_i}, respectively. Therefore all vertices of $G^c_{w_i}$ are mapped to the side of S opposite to w_i. □

(11) *A cut vertex component G_{w_i} contains at most one two-sided block B and if so then w_i is a side vertex of B. Moreover, if B' is a side block of w_i (one-sided or two-sided), then either $L(B') = \{w_i\}$ or $R(B') = \{w_i\}$, $1 \leqslant i \leqslant m$.*

Proof. By Statement (10) each extremal and side subgraph of G_{w_i}, except for vertex w_i, is mapped to one of the sides of S. Recall also, that by Statement (2) the sides of a two-sided block have to be mapped to different sides of S. Therefore, the only two-sided block of an extremal or side subgraph of G_{w_i} can be the one which is attached to w_i. For the sake of contradiction assume that the two-sided block B is contained in an extremal subgraph of G_{w_i}, therefore w_i is B's switch. By Statement (2) the sides of B have to be mapped to different sides of S, which still violates Statement (10). Thus, B can be only contained in a side subgraph of G_{w_i} and have w_i as its side vertex.

Let now, B' be a side block attached to w_i, either one or two-sided. Assume for the sake of contradiction that neither $L(B') = \{w_i\}$ or $R(B') = \{w_i\}$. W.l.o.g assume that $w_i \in L(B')$ and let $a \in L(B')$. Vertices w_i and a are connected by a directed path from a to w_i (resp. from w_i to a), see Fig. 17.b. Note that $a \notin P_{s,t}$, since by the definition of G_{w_i}, w_i is the only common vertex of $P_{s,t}$ and G_{w_i}. By Statement (2), vertices a and w_i are mapped to the same side of S. Therefore block B' crosses the path $P_{s,t}$. □

5.3. Two restricted UPSE problems for outerplanar DAGs

Analogously to the restricted UPSE for trees, in this section we study two restricted UPSE problems for outerplanar graphs. These problems are later on used by our main algorithm which decides whether there exists an UPSE of a given outerplanar DAG into a given convex point set.

5.3.1. Restricted UPSE for cut vertex components

Let G be an outerplanar DAG, let $P_{s,t}$ be a path in G from its source s to its sink t and let $P^c_{s,t} = \{s = w_1, \ldots, w_m = t\}$ be the (s,t)-separating subset of $P_{s,t}$. Let G_{w_i} be one of the cut vertex components defined by the path $P_{s,t}$. The restricted UPSE for cut vertex components deals with an UPSE of G_{w_i}. In Lemma 11, we proved that, if G admits an UPSE Γ_G into a convex point set S, then the following facts are true:

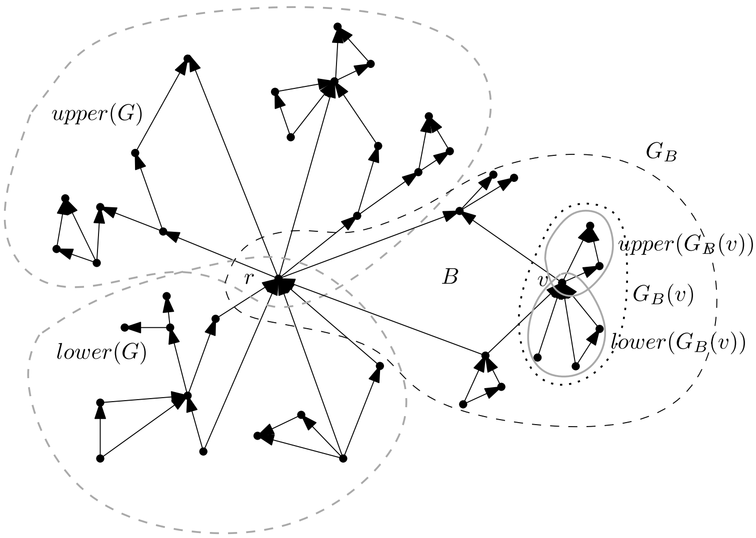

Fig. 18. Parts of the graph G treated by the proof of Theorem 9. Subgraphs *upper*(G) and *lower*(G) are indicated by gray dashed curves. Subgraph G_B is surrounded by black dashed curve. Subgraph $G_B(v)$ is indicated by a black dotted curve. Subgraphs *upper*$(G_B(v))$ and *lower*$(G_B(v))$ are indicated by gray solid curves.

1. The vertices of G_{w_i} are mapped in Γ_G on a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side (Statement (6)).
2. If G_{w_i} contains a side block B' attached to w_i, then w_i is a single side vertex of one of its sides. By Statement (1), B' is the only side block attached to w_i. Moreover there cannot be any two-sided block other than B' in G_{w_i} (Statement (11)).
3. All the vertices of a single extremal or side subgraph of G_{w_i}, $1 \leqslant i \leqslant m$, except for vertex w_i, are mapped to consecutive points of the same side of S, with the vertices of a side subgraph mapped to the side of S opposite to that of w_i (Statement (10)).

Taking into consideration these necessary conditions we define and solve the following problem.

Definition 2. In a *restricted cut vertex component UPSE* problem (RCVC-UPSE, for short) we are given an outerplanar DAG G rooted at a vertex r, so that if r is a side vertex of a block B of G, then: (1) r is the only side vertex on one of the sides of B, (2) block B is the only block of G that can be two-sided, and (3) B is a unique block having r as its side vertex (see Fig. 18 for a possible entry graph G of the problem). We are also given a convex point set S, and a designated point $p_r \in S$. We are asked to decide whether there exists an UPSE of G into S such that:

(i) the root r of G is mapped to point p_r; and
(ii) each extremal subgraph of G is mapped to consecutive points on the same side (either L or R) of S (excluding the vertex r);
(iii) all the vertices of the side subgraph of G containing block B (note that G can have at most one side subgraph by assumption (3) of the problem), except for vertex r, are mapped to the consecutive points of the side of S opposite to that where vertex r is mapped.

An UPSE of G into S that satisfies these requirements is called a RCVC-UPSE.

Theorem 9. *Let G be an n-vertex outerplanar DAG as described in RCVC-UPSE problem. Let L and R be left-sided and right-sided convex point sets, resp., such that $S = L \cup R$ is a convex point set of size n, and let p_r be a point of S. The RCVC-UPSE problem with input G, S, and p_r can be decided in $O(b(r)|G|)$ time. Moreover, if a RCVC-UPSE for G, S, and p_r exists, it can also be constructed in $O(b(r)|G|)$ time.*

Proof. Let G be a graph rooted at vertex r, as described in the RCVC-UPSE problem. Recall that by $b^-(r)$ (resp., $b^+(v)$) we denote the number of extremal blocks of r that have r as their sink (resp., source), and $b(v) = b^-(v) + b^+(v)$. Recall that r should be mapped to a pre-specified point p_r. Assume without loss of generality that $p_r \in L$ and let p_r be the k-th point of L. Graph G has at most one block, B, having r as its side vertex. Let G_B be the side subgraph of G that contains B (see Fig. 18). By the requirement of the problem, the vertices of G_B should be mapped to consecutive points of S, to the side opposite to r. Recall also that all the vertices of a single extremal subgraph of G have to be mapped to consecutive points of a single side

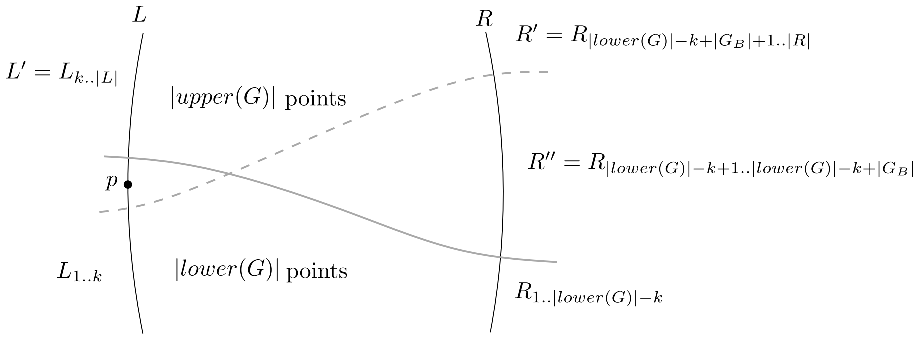

Fig. 19. (a) The partition of the point set described in the proof of Theorem 9.

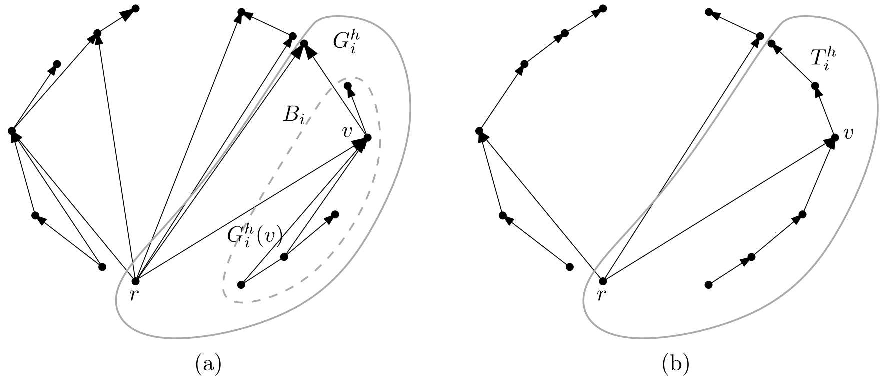

Fig. 20. (a) The graph *upper*(G), its extremal subgraph G_i^h and its subgraph $G_i^h(v)$. (b) The upward skeleton of *upper*(G). The subgraph *lower*$(G_i^h(v))$ is substituted by a path of length $|lower(G_i^h(v))| - 1$.

of S. Thus, we infer that the vertices of *lower*(G) have to be mapped to the points of S comprised by two consecutive subsets of S: (i) all the points of L below p_r, i.e., the point set $L_{1..k}$, and (ii) the $|lower(G)| - k$ lowest points of R, i.e., $R_{1..|lower(G)|-k}$ (see Fig. 19). The vertices of G_B have to be mapped to $\{p_r\} \cup R_{|lower(G)|-k+1..|lower(G)|-k+|G_B|}$. Finally, the vertices of *upper*(G) are mapped to the point set $L_{k..|L|} \cup R_{|lower(G)|-k+|G_B|+1..|R|}$. Let us denote $L_{k..|L|}$ by L', $R_{|lower(G)|-k+|G_B|+1..|R|}$ by R', and $R_{|lower(G)|-k+1..|lower(G)|-k+|G_B|}$ by R''. We first show how to test the existence of a RCVC-UPSE of *upper*(G) into point set $L' \cup R'$. Testing of the existence of a RCVC-UPSE of *lower*(G) into the point set $L_{1..k} \cup R_{1..|lower(G)|-k}$ is symmetrical.

We construct a tree $T_{upper(G)}$, called *upward skeleton* of *upper*(G) and we prove that *upper*(G) has a RCVC-UPSE into $L' \cup R'$, with r mapped to p_r, if and only if $T_{upper(G)}$ has a restricted UPSE into $L' \cup R'$, with r mapped to p_r. Let $G_1^h, \ldots, G_{b^+(r)}^h$ be the extremal subgraphs of *upper*(G), see Fig. 20.a. We construct tree $T_{upper(G)}$ to consist of a root r and subtrees $T_1^h, \ldots, T_{b^+(r)}^h$, rooted at vertices $r_1, \ldots, r_{b^+(r)}$, respectively, that are connected to r by the edges (r, r_i), $1 \leqslant i \leqslant b^+(r)$, respectively (see Fig. 20.b). Consider graph G_i^h and let B_i be the extremal block of r contained in G_i^h. By the requirements of the RCVC-UPSE, B_i is a one-sided block. Note that B_i can be also a trivial block. Thus, vertex r and the sink of B_i must be adjacent. Let v be a vertex of B_i adjacent to r and different from the sink of B_i, if such vertex exists, otherwise let v be the sink of B_i. Consider the graph $G_i^h(v)$ (see Fig. 20.a). Tree T_i^h consists of its root r_i, of a monotone path consisting of $|lower(G_i^h(v))|$ vertices, with sink at vertex r_i, and of a monotone path consisting of $|G_i^h| - |lower(G_i^h(v))| + 1$ vertices, with source at vertex r_i.

Lemma 12. *The outerplanar DAG upper(G), rooted at r, admits a RCVC-UPSE into $L' \cup R'$ with r mapped to p_r iff each extremal subgraph of upper(G), rooted at r, is one-side embeddable and the upward skeleton $T_{upper(G)}$ of upper(G) admits a restricted UPSE into $L' \cup R'$ with r mapped to p_r.*

Proof. Assume that *upper*(G), rooted at r, admits a RCVC-UPSE into $L' \cup R'$ with r mapped to p_r. Let Γ be such an embedding. We transform this embedding into a restricted UPSE of $T_{upper(G)}$ into $L' \cup R'$. We consider G_i^h, $i = 1, \ldots, b^+(r)$. Since Γ is a RCVC-UPSE, G_i^h (except possibly for vertex r) uses consecutive points of $L' \cup R'$. Let v be a vertex of B_i adjacent to r and different from the sink of B_i, if such vertex exists, otherwise let v be the sink of B_i. Consider $G_i^h(v)$, in tree T_i^h the *lower*$(G_i^h(v))$ is substituted by a directed path of $|lower(G_i^h(v))|$ vertices. We draw this path on the points where *lower*$(G_i^h(v))$ is drawn, using the points in the order of increasing y-coordinate. The rest of the vertices of T_i^h are embedded on the points where the vertices of $upper(G_i^h)$ are placed. Now consider a drawing of a single G_i^h, $i = 1, \dots, b^+(r)$. Subgraph G_i^h is rooted at r and r is mapped to p_r. Let $S_i \cup \{p_r\}$ be the points of $L' \cup R'$ occupied by the G_i^h. Consider a virtual point p_i that is below the points of S_i so that $S_i \cup \{p_i\}$ creates a one-sided convex point set. Place vertex r to point p_i. Note that the new drawing of G_i^h is upward and planar. Thus G_i^h is one-side embeddable.

Now assume that $T_{upper(G)}$ admits a restricted UPSE into $L' \cup R'$ with r mapped to p_r. Let T_i^h, $i = 1, \dots, b^+(r)$ be the subtrees of $T_{upper(G)}$ rooted at $r_1, \dots, r_{b^+(r)}$ respectively. Each of T_i^h, $i = 1, \dots, b^+(r)$ is mapped to consecutive points of $L' \cup R'$, call this point set S_i. Now consider G_i^h, $i = 1, \dots, b^+(r)$. Recall that G_i^h is one-side embeddable by assumption. Consider $S_i \cup \{p_i\}$, where p_i is a virtual point lower than all points of S_i, such that point set $S_i \cup \{p_i\}$ is one-sided. Map G_i^h to $S_i \cup \{p_i\}$ so that r is mapped to its lowest point, i.e. to p_i. Move r to the point p_r. Note that in any UPSE of G_i^h to $S_i \cup \{p_i\}$, vertex v is mapped higher than all vertices of $lower(G_i^h(v))$. Thus, since edge (r, v), connecting vertex r with tree T_i^h, is upward in the UPSE of $T_{upper(G)}$ it is also upward in the constructed drawing of G_i^h on point set $S_i \cup \{p\}$. □

By Lemma 12, $upper(G)$ admits a RCVC-UPSE into $L' \cup R'$ if and only if each extremal subgraph of $upper(G)$ rooted at r is one-side embeddable and the upward skeleton $T_{upper(G)}$ of $upper(G)$ admits a restricted UPSE into $L' \cup R'$ with r mapped to p_r. One-side embeddability of extremal subgraphs of $upper(G)$ can be tested in linear time by Lemma 10. Finally, the existence of a restricted UPSE of $T_{upper(G)}$ into $L' \cup R'$ can be tested in $O(b^+(r)|upper(G)|)$ time, by using the dynamic programming procedure explained in Section 4.1 (see also Theorem 4).

Next we explain how to test whether side subgraph G_B has an UPSE into points $\{p_r\} \cup R''$, where r is mapped to p_r. Note that the positions of the vertices of block B are predetermined by the sizes of the extremal graphs of the vertices of B. I.e. for any edge (v, v') of B, the vertices of $lower(G_B(v'))$ and $upper(G_B(v))$ have to be drawn above v and below v', respectively. Let Γ_B be a drawing of the vertices of B on points of $\{p_r\} \cup R''$ so that for each edge (v, v') of B, there are $|lower(G_B(v'))| + |upper(G_B(v))| - 2$ free points between the points to which v and v' are mapped. If Γ_B is upward and for every $v \in B$ each of the $lower(G_B(v))$ and $upper(G_B(v))$ is one-side embeddable, an UPSE of G_B on point set $\{p_r\} \cup R''$, so that r is mapped to p_r, can be constructed in time $O(|G_B|)$, by Lemma 10.

Finally, since the size of G_B is bounded by $|G|$ the overall time complexity is $O(b^+(r)|G| + b^-(r)|G| + |G|) = O(b(r)|G|)$. □

Similarly to the case of trees, denote by $\mathcal{L}(G, L, R)$ the set of points $p_r \in L \cup R$ such that there exists a RCVC-UPSE of G into $L \cup R$ so that the root of G is mapped to p_r. The next theorem follows easily from Theorem 9 by testing each point of $L \cup R$ as a candidate host for $r(G)$.

Theorem 10. *Let G be an n-vertex outerplanar DAG rooted at vertex r, so that if r is a side vertex of a block B of G, then* (1) *r is the unique side vertex on one of the sides of B and,* (2) *block B is the unique block of G that can be two-sided. Let L and R be left-sided and right-sided convex point sets, resp., such that $L \cup R$ is a convex point set of size n. Then, set $\mathcal{L}(G, L, R)$ can be computed in $O(b(r)n^2)$ time.*

5.3.2. Restricted UPSE for path components

Let again G be an outerplanar DAG, S be a convex point set, let $P_{s,t}$ be a path from a source s to a sink t of G and let $P_{s,t}^c = \{s = w_1, \dots, w_m = t\}$ be the (s,t)-separating subset of $P_{s,t}$. Let $G_{w_i, w_{i+1}}$ be a path component of G defined by $P_{s,t}$. Let B_i be the block of $G_{w_i, w_{i+1}}$ that contains both w_i and w_{i+1}. If B_i is a trivial block, i.e., an edge then its drawing is determined uniquely by the positions of vertices w_i and w_{i+1}. Thus, in case of a tree, we only had to check whether the edge connecting w_i and w_{i+1} is drawn upward. Assume now that B_i is not a trivial block, then $G_{w_i, w_{i+1}}$ might have more drawings on a point set corresponding to it. In this section we present a restricted UPSE problem that deals with an UPSE of a path component.

We first recall that, by the statements of Lemma 11, if G admits an UPSE Γ_G into S with s and t mapped to $b(S)$ and $t(S)$, respectively, then the following facts are true for the structure of $G_{w_i, w_{i+1}}$ and its UPSE into a subset of S.

1. Vertex w_i (resp. w_{i+1}) of block B_i either coincides with the source s_i (resp. sink t_i) of B_i or is adjacent to it (Statement (3)).
2. If vertex w_i (resp. vertex w_{i+1}) of block B_i is incident to the source s_i (resp. the sink t_i) of B_i, then s_i (resp. t_i) is mapped in Γ_G to the opposite side to which w_i (resp. w_{i+1}) is mapped (Statement (4)).
3. The vertices of $G_{w_i, w_{i+1}}$, $1 \leqslant i \leqslant m-1$, except for w_i and w_{i+1} are mapped in Γ_G to a subset of S comprised by two consecutive point sets of S, one on its left and one on its right side (Statement (7)).
4. For each $v \in B_i$, $v \neq w_i, w_{i+1}$, the vertices of $G_{w_i, w_{i+1}}(v)$ are mapped in Γ_G to the same side where v is mapped on consecutive points around v (Statement (9)).

Based on these necessary conditions we define and solve the following problem:

Definition 3. Let G be an n-vertex outerplanar DAG G, let B be a block of G, and let s and t be the source and the sink of B. Let also w and w' be vertices of B such that w (resp. w') either coincides with s (resp. t) or is adjacent to it. We assume that G is rooted at w. Let S be a convex point set of n points and let q and p be two points of S. In a *restricted path component UPSE* problem (RPC-UPSE, for short) we are asked to determine whether there exists an UPSE of G into S such that:

(i) Vertices w and w' are mapped to the points q and p, respectively.
(ii) If w (resp. w') is incident to s (resp. t), then s (resp. t) is mapped to the side of S opposite to which w (resp. w') is mapped.
(iii) For each $v \in B$, $v \neq w, w'$, the vertices of $G(v)$ are mapped to the same side of S where v is mapped on consecutive points around v.

An UPSE of G into S that satisfies these requirements is called a RPC-UPSE.

Lemma 13. *Let G be an outerplanar DAG, B be a block of G, w and w' be vertices of B, S be a convex point set, and q and p be points of $S = L \cup R$, as defined in the RPC-UPSE problem. RPC-UPSE problem can be solved in $O(|G|)$ time. If a RPC-UPSE of G into S exists we can also construct it within the same time bound.*

Proof. If B is a trivial block then it has a unique drawing defined by the positions of w and w' and we only need to test the upwardness of the edge $\{w, w'\}$, thus the lemma holds. Assume that B is not a trivial block. Recall that, when a block B of graph G is specified, $L(G, B)$ (resp. $R(G, B)$) denotes all the vertices of $L(B)$ (resp. $R(B)$) plus the vertices of the components incident to the vertices of $L(B)$ (resp. $R(B)$), and we refer to $L(G, B)$ and $B(G, B)$ as to the sides of G with respect to block B.

By definition of RPC-UPSE, vertices s and t either coincide with w and w', or are adjacent to them, respectively. By Statement (2) of Lemma 11, in any UPSE of B into some points of S, all the vertices of $L(B)$ (resp. $R(B)$) are mapped to the same side of S. However, by RPC-UPSE again, if w (resp. w') is incident to s (resp. t) then s (resp. t) has to be mapped to the opposite side to which w (resp. w') is mapped. Thus, if at least one of w and w' is different from s and t, respectively, it is predetermined to which side of the point set the vertices of $\{L(B), R(B), s, t\}$ are mapped, since we know to which side of the point set vertices w and w' are mapped.

Based on these notes we define a *side mapping* of G. Side mapping of G is a function $M : V(G) \to \{\lambda, \rho\}$, so that:

1. For each $v \in G$, $M(v) = \lambda$ (resp. $M(v) = \rho$) if vertex v is mapped by M to the left (resp. right) side of the given point set.
2. $M(w)$ and $M(w')$ are already given since we know that w and w' should be mapped to points q and p, respectively.
3. If $w \neq s$ then $M(w) \neq M(s)$. For each $v \in G$ which belongs to the same side of G where w belong, set $M(v) = M(w)$, for the vertices of the other side of G, set $M(v) \neq M(w)$.
4. If $w' \neq t$ then $M(w') \neq M(t)$. For each $v \in G$ which belongs to the same side of G where w' belong, set $M(v) = M(w')$, for the vertices of the other side of G, set $M(v) \neq M(w')$.
5. If $w = s$ and $w' = t$ then we define two distinct side mappings: (i) for all $v \in L(G, B)$ set $M_1(v) = \lambda$ and for all $v \in R(G, B)$ set $M_1(v) = \rho$, (ii) for all $v \in L(G, B)$ set $M_2(v) = \rho$ and for all $v \in R(G, B)$ set $M_2(v) = \lambda$.

Note that a side mapping for G simply expresses the requirements of the RPC-UPSE problem. For the simplicity we introduce the following notation:

$$\lambda(M) := \left|\bigcup_{v \in G}\bigl(M(v) = \lambda\bigr)\right|, \qquad \rho(M) := \left|\bigcup_{v \in G}\bigl(M(v) = \rho\bigr)\right|.$$

Let M be a side mapping of G. Since the number of vertices of G that is mapped to the left (resp. right) side of the given point set should be $|L|$ (resp. $|R|$), we get the first necessary condition:

$$\lambda(M) = |L|, \qquad \rho(M) = |R|.$$

If the above equations hold we say that M is a *valid side mapping* of G.

Assume that there exists a valid side mapping of G. By requirement (iii) of RPC-UPSE, for each $v \in B$, $G(v)$ has to be drawn on the same side of S, where v is placed, on the consecutive points around v. Thus, both *upper*$(G(v))$ and *lower*$(G(v))$ should be one-side embeddable. This is the second necessary condition.

Finally, consider the following drawing Γ_B of B on S: let v be a vertex of B. Recall that a valid mapping M, maps vertex v to one of the two sides of the given point set. Leave $|upper(G(v))| - 1$ and $|lower(G(v))| - 1$ points free above and below the point where v is mapped. We call Γ_B the drawing of B *induced by mapping* M. It is clear that the drawing of B induced by M should be upward, this is the third necessary condition.

Next we overview the algorithm which tests the three necessary conditions and constructs the drawing if the conditions are fulfilled, i.e., we prove that these conditions are also sufficient. We also discuss the time complexity.

Algorithm 2: OUTERPLANAR-UPSE(G, S, s, t)

input : An outerplanar DAG G, a point set $S = L \cup R$, a source s and a sink t of G and a path $P_{s,t}$. Path $P^c_{s,t} = (s = w_1, \dots, w_m = t)$ is used to progressively build graph G from subgraphs G_{w_{i-1},w_i}, $2 \leqslant i \leqslant m$ and G_{w_i}, $1 \leqslant i \leqslant m$.
output : "YES" if G has an UPSE into S with s mapped to $b(S)$ and t mapped to $t(S)$, "NO" otherwise.

For $a = 0 \dots |L|$
  For $b = 0 \dots |R|$
    $\mathcal{P}(a, b, 1) = \mathcal{L}(G_{w_1}, L_{1..a}, R_{1..b}) \cap \{b(S)\}$
    For $k = 2 \dots m$ //Consider outerplanar DAG G_{w_k}
      $\mathcal{P}(a, b, k) = \emptyset$
      For $i = 0 \dots |G_{w_k}|$ //We consider the case where i vertices of G_{w_k} are placed to the left side of S
        if $(a - i \geqslant 0)$ **and** $(b - (|G_{w_k}| - i) \geqslant 0)$
          Let $\mathcal{L} = \mathcal{L}(G_{w_k}, L_{a-i+1..a}, R_{b-(|G_{w_k}|-i)+1..b})$
          //We consider all the possible placements of vertex w_k
          For each p in $\mathcal{L}$
            Let M be a side mapping of G_{w_{k-1},w_k}, when w_k is
            mapped to p and w_{k-1} is mapped to a point of L.
            //We consider all possible placements of w_{k-1}
            to the left side of the point set
            For each q in
            $\mathcal{P}(a - i - \lambda(M), b - (|G_{w_k}| - i) - \rho(M), k - 1) \cap L$
              if RPC-UPSE$(w_{k-1}, w_k, L_{a-i-\lambda(M)+1..a-i} \cup$
                $R_{b-(|G_{w_k}|-i)-\rho(M)+1..b-(|G_{w_k}|-i)} \cup \{p, q\}, q, p)$
                then add p to $\mathcal{P}(a, b, k)$.
            Let M be a side mapping of G_{w_{k-1},w_k}, when w_k is
            mapped to p and w_{k-1} is mapped to a point of R.
            //We consider all possible placements of w_{k-1}
            to the right side of the point set
            For each q in
            $\mathcal{P}(a - i - \lambda(M), b - (|G_{w_k}| - i) - \rho(M), k - 1) \cap R$
              if RPC-UPSE$(w_{k-1}, w_k, L_{a-i-\lambda(M)+1..a-i} \cup$
                $R_{b-(|G_{w_k}|-i)-\rho(M)+1..b-(|G_{w_k}|-i)} \cup \{p, q\}, q, p)$
                then add p to $\mathcal{P}(a, b, k)$.
if $t(S) \in \mathcal{P}(|L|, |R|, m)$ **then return** ("YES");
return ("NO");

1. Test whether there exists a valid side mapping M of G to S. This can be done in time proportional to the size of G.
2. Let Γ_B be drawing of B induced by mapping M. Test whether Γ_B is upward. The construction of Γ_B and the upwardness test can be done in time proportional to the size of G.
3. For each $v \in B$, test whether $upper(G(v))$ and $lower(G(v))$ are one-side embeddable. This can be done in time proportional to the size of the graph due to Lemma 10. The construction of the required RPC-UPSE of G into S is then completed by the construction of an UPSE of $upper(G(v))$ (resp. $lower(G(v))$) into a one-side convex point set of the same size, comprised by the consecutive points lying above (resp. below) v, which are left free by the definition of the drawing of B induced by a valid mapping. By Lemma 10 this can be done in time $O(|upper(G(v))|)$ (resp. $O(|lower(G(v))|)$).

This concludes the proof of the lemma. □

Given G, block B of G, vertices w, w' of B, convex point set S and points q, p of S, as in RPC-UPSE problem, the procedure RPC-UPSE(G, w, w', S, q, p) returns *TRUE* if there exists a RPC-UPSE of G into S with vertices w and w' mapped to points q and p, respectively.

5.4. Testing algorithm for outerplanar DAGs

Let G be an outerplanar DAG and s and t be a source and a sink of G, respectively. Let S be a convex point set. In this section we present a polynomial time algorithm that tests whether G admits an UPSE into S, such that s and t are mapped to $b(S)$ and $t(S)$, respectively.

Let $P_{s,t}$ be a path from s to t and let $P^c_{s,t} = \{s = w_1, \dots, w_m = t\}$ be the (s,t)-separating subset of $P_{s,t}$. Let G_{w_i}, $1 \leqslant i \leqslant m$, and $G_{w_i,w_{i+1}}$, $1 \leqslant i \leqslant m-1$, be the cut vertex components and the path components defined by $P_{s,t}$. Finally, let G_{s,w_i}, $1 \leqslant i \leqslant m$, be the subgraphs of G defined by $P_{s,t}$ and w_i.

Similarly to the case of trees, the dynamic programming Algorithm 2 maintains a list of points $\mathcal{P}(a, b, k)$, $0 \leqslant a \leqslant |L|$, $0 \leqslant b \leqslant |R|$, $1 \leqslant k \leqslant m$, such that

$$p \in \mathcal{P}(a, b, k) \iff \begin{cases} G_{s,w_k} \text{ has an UPSE into point set } S_{1..a,1..b} \text{ with} \\ \text{vertices } s \text{ and } w_k \text{ mapped to points } b(S) \text{ and } p, \\ \text{respectively.} \end{cases}$$

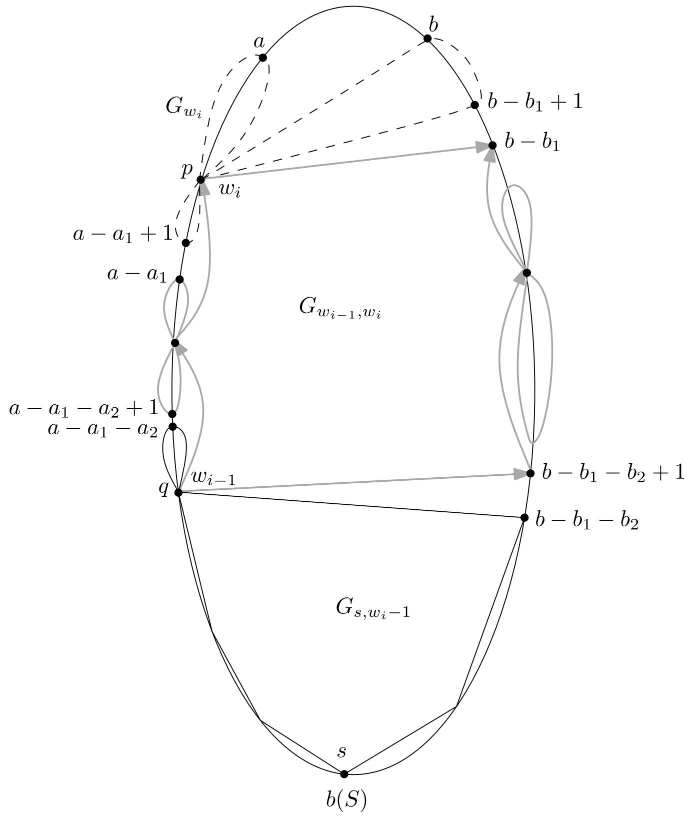

Fig. 21. Construction of an UPSE of G_{s,w_i} from an UPSE of $G_{s,w_{i-1}}$, a RPC-UPSE of G_{w_{i-1},w_i} and a RCVC-UPSE of G_{w_i}. Subgraph $G_{s,w_{i-1}}$ is drawn using black lines, path component G_{w_{i-1},w_i} using gray lines, and cut vertex component G_{w_i} using black dashed line.

For the boundary conditions of our dynamic programming we have that

$$\mathcal{P}(a,b,1) = \mathcal{L}(G_{w_1}, L_{1..a}, R_{1..b}) \cap \{b(S)\} \quad \text{where } a+b=|G_{w_1}|.$$

Note that since $w_1 = s$, and by assumption vertex s is mapped to point $b(S)$ we insist that $\mathcal{P}(a,b,1)$ is either $\{b(s)\}$ or $\emptyset$.

Assume that $G_{s,w_{i-1}}$ has an UPSE into point set $S_{1..a-a_1-a_2,1..b-b_1-b_2}$ with vertex w_{i-1} mapped to point q (see Fig. 21, black graph). Assume also that G_{w_i} has a RCVC-UPSE in $L_{a-a_1+1..a} \cup R_{b-b_1+1..b}$ with w_i mapped to p (Fig. 21, dashed graph). Assume finally, that G_{w_{i-1},w_i} has a RPC-UPSE into point set $L_{a-a_1-a_2+1..a-a_1} \cup R_{b-b_1-b_2+1..b-b_1} \cup \{p,q\}$ (Fig. 21, gray graph). Then we can combine the UPSE for $G_{s,w_{i-1}}$ with the RCVC-UPSE for G_{w_i} and with the RPC-UPSE for G_{w_{i-1},w_i}, in order to get an UPSE of G_{s,w_i} on point set $S_{1..a,1..b}$ with vertices s and w_i mapped to $b(S)$ and p, respectively. This procedure allows us to add points in $\mathcal{P}(a,b,i)$ and is described formally by the following recurrence relation:

$$\begin{aligned}\mathcal{P}(a,b,i) = \Big\{p \;\Big|\; & \exists a_1, b_1 \in \mathbb{N}\cup\{0\}\colon a_1+b_1 = |G_{w_i}| \\ & \exists a_2, b_2 \in \mathbb{N}\cup\{0\}\colon a_2+b_2 = |G_{w_{i-1},w_i}| - 2 \\ & \textbf{and } p \in \mathcal{L}(G_{w_i}, L_{a-a_1+1..a}, R_{b-b_1+1..b}) \\ & \textbf{and } \exists q \in \mathcal{P}(a-a_1-a_2, b-b_1-b_2, i-1) \\ & \textbf{and } \exists \text{RPC-UPSE of } G_{w_{i-1},w_i} \text{ into} \\ & L_{a-a_1-a_2+1..a-a_1} \cup R_{b-b_1-b_2+1..b-b_1} \cup \{p,q\}, \\ & \text{where } w_{i-1} \text{ and } w_i \text{ are mapped to} \\ & q \text{ and } p, \text{ respectively}\Big\}. \end{aligned} \tag{2}$$

Lemma 14. *Let G be an n-vertex outerplanar DAG, S be a convex point set of size n. Let s and t be a source and a sink of G, respectively, let $P_{s,t}$ be a path connecting s and t in G, let $P^c_{s,t} = (s = w_1, \ldots, w_m = t)$ be the (s,t)-separating subset of $P_{s,t}$ and let G_{s,w_i}, $1 \leqslant i \leqslant m$ be the subgraphs of G, defined by $P_{s,t}$ and w_i. There exists an UPSE of G into S such that s and t are mapped to $b(S)$ and $t(S)$, respectively, if and only if $\mathcal{P}(|L|,|R|,m)$, computed by the recurrence relation (2), contains $t(S)$.*

Proof. We first prove that if $p \in \mathcal{P}(a,b,i)$ then G_{s,w_i} has an UPSE into point set $S_{1..a,1..b}$ with vertex w_i mapped to point p. From the boundary conditions, this is true for $i=1$. Assume that if $q \in \mathcal{P}(a,b,i-1)$ then $G_{s,w_{i-1}}$ has an UPSE into $S_{1..a,1..b}$ with vertex w_{i-1} mapped to point q. Let now $p \in \mathcal{P}(a,b,i)$. Then by the definition of the recurrence relation we infer that:

1. There exist a_1 and $b_1 \in \mathbb{N} \cup \{0\}$ so that $a_1 + b_1 = |G_{w_i}|$.
2. There exist a_2 and $b_2 \in \mathbb{N} \cup \{0\}$ so that $a_2 + b_2 = |G_{w_{i-1},w_i}| - 2$.
3. Point $p \in \mathcal{L}(G_{w_i}, L_{a-a_1+1..a}, R_{b-b_1+1..b})$, which by the definition of $\mathcal{L}$, means that there exists a RCVC-UPSE of G_{w_i} into $L_{a-a_1+1..a} \cup R_{b-b_1+1..b}$ with w_i mapped to p.
4. There exists $q \in \mathcal{P}(a-a_1-a_2, b-b_1-b_2, i-1)$, thus, by induction hypothesis, $G_{s,w_{i-1}}$ has an UPSE into $L_{1..a-a_1-a_2} \cup R_{1..b-b_1-b_2}$.
5. There exists a RPC-UPSE of G_{w_{i-1},w_i} into $L_{a-a_1-a_2+1..a-a_1} \cup R_{b-b_1-b_2+1..b-b_1} \cup \{p,q\}$ where w_{i-1} and w_i are mapped to q and p respectively.

Then we combine the UPSE for $G_{s,w_{i-1}}$ into point set $S_{1..a-a_1-a_2,1..b-b_1-b_2}$ with the RCVC-UPSE for G_{w_i} into point set $S_{a-a_1+1..a,b-b_1+1..b}$ and with the RPC-UPSE of G_{w_{i-1},w_i} into point set $S_{a-a_1-a_2+1..a-a_1,b-b_1-b_2+1..b-b_1} \cup \{p,q\}$ in order to get an UPSE of G_{s,w_i} on point set $S_{1..a,1..b}$. Since each extremal and one side subgraph of G_{w_i} are drawn entirely on a single side of S we get that the resulting drawing is planar. It is upward as a combination of upward drawings.

For the reversed statement we also work by induction. From the boundary conditions we know that if $G_{s,w_1} = G_{w_1}$ has an UPSE into a point set $S_{1..a,1..b}$ then $b(S) \in \mathcal{P}(a,b,1)$, where $a+b = |G_{w_1}|$. Assume that the statement is true for $G_{s,w_{i-1}}$, i.e., if $G_{s,w_{i-1}}$ has an UPSE into a point set $S_{1..a,1..b}$ with vertex w_{i-1} mapped to q then $q \in \mathcal{P}(a,b,i-1)$. Assume also that G_{s,w_i} has an UPSE into a point set $S_{1..a,1..b}$ with vertices s and w_i mapped to points $b(S)$ and p, respectively. Recall Statements (5)–(7) of Lemma 11. By these statements, there exist numbers a_1, b_1, a_2, b_2, so that

1. $a_1 + b_1 = |G_{w_i}|$, and $p \in S_{a-a_1+1..a,b-b_1+1..b}$.
2. $a_2 + b_2 = |G_{w_{i-1},w_i}| - 2$.
3. The UPSE of G_{s,w_i} contains the UPSE of $G_{s,w_{i-1}}$ into point set $S_{1..a-a_1-a_2,1..b-b_1-b_2}$ with vertex w_{i-1} mapped to some point q of $S_{1..a-a_1-a_2,1..b-b_1-b_2}$. By induction hypothesis we infer that $q \in \mathcal{P}(a-a_1-a_2, b-b_1-b_2, i-1)$.
4. Cut vertex component G_{w_i} is mapped to the point set $S_{a-a_1+1..a,b-b_1+1..b}$, with vertex w_i mapped to point p. By the definition of $\mathcal{L}$ we get that $p \in \mathcal{L}(G_{w_i}, L_{a-a_1+1..a}, R_{b-b_1+1..b})$.
5. Path component G_{w_{i-1},w_i} is mapped to the point set $S_{a-a_1-a_2+1..a-a_1,b-b_1-b_2+1..b-b_1} \cup \{p,q\}$, with vertices w_{i-1}, w_i mapped to points q, p, respectively. Moreover, by Statement (9) of Lemma 11, if B_i is the block of G_{w_{i-1},w_i} that contains both w_{i-1} and w_i, then for each $v \in B_i$, $v \neq w_{i-1}, w_i$ the vertices of $G_{w_{i-1},w_i}(v)$ are mapped on the same side of S around v. Therefore, there exists a RPC-UPSE of G_{w_{i-1},w_i} into $S_{a-a_1-a_2+1..a-a_1,b-b_1-b_2+1..b-b_1} \cup \{p,q\}$, with vertices w_{i-1}, w_i mapped to points q, p, respectively.

Thus, by the definition of recurrence relation (2) we infer that $p \in \mathcal{P}(a,b,i)$.

Finally, for $i=m$ we infer that an UPSE of T into S such that source s and sink t are mapped to $b(S)$ and $t(S)$, respectively, exists if and only if $\mathcal{P}(|L|,|R|,m)$ contains $t(S)$. □

The following theorem concludes the section and is proved along the same lines as Theorem 6 and Theorem 7.

Theorem 11. *Let G be an n-vertex outerplanar DAG, S be a convex point set of size n, s be a source of G and t be a sink of G. It can be tested in $O(n^6)$ time whether G has an UPSE into S. Moreover, if such an UPSE exists, it can be constructed within the same time bound.*

Proof. Algorithm 2 calculates the values $\mathcal{P}(a,b,k)$, when $0 \leqslant a \leqslant |L|$, $0 \leqslant b \leqslant |R|$, $1 \leqslant k \leqslant m$, thus from Lemma 6, we infer that Algorithm 2 decides whether G has an UPSE into S such that s and t are mapped to $b(S)$ and $t(S)$, respectively.

We first prove that Algorithm 2 terminates in time $O(n^6)$ and then improve it to $O(n^5)$. Note the following:

1. By Theorem 10, the list $\mathcal{L}$ in line 8 can be computed in time $O(d(w_k)|G_{w_k}|^2)$ and contains at most $2|G_{w_k}|$ values.
2. In line 10, variable p runs over at most $2|G_{w_k}|$ values.
3. In lines 15 and 24, the variable q runs over at most $2|G_{w_{k-1}}|$ different values. This happens because by Statement (6) of Lemma 11, vertex q must lie on one of the $|G_{w_{k-1}}|$ highest points of the corresponding point set. The factor of 2 comes from the fact that q can lie either on the left or on the right side of the corresponding point set.
4. By Lemma 13, the test of lines 17–18 and 26–27 can be done in time $O(|G_{w_{k-1},w_k}|)$.

Thus, the complexity of the algorithm is limited by the following value:

$$\sum_{a=0}^{|L|}\sum_{b=0}^{|R|}\sum_{k=2}^{m}\sum_{i=0}^{|G_{w_k}|}\left(d(w_k)|G_{w_k}|^2 + |G_{w_{k-1}}| \cdot |G_{w_k}| \cdot |G_{w_{k-1},w_k}|\right) = O\left(n^6\right).$$

A factor of n can be saved by realizing that in our dynamic programming we can maintain a list $\mathcal{P}'(a,i)$ which uses only one parameter for the left side of the convex set (in contrast with $\mathcal{P}(a,b,i)$ which uses a parameter for each side of S). The number of points on the right side of S is implied since the size of graph G_{s,w_i} is fixed. For simplicity, we have decided to use notation $\mathcal{P}(a,b,i)$.

By applying Algorithm 2 on all $\langle source, sink\rangle$ pairs of T we can decide whether outerplanar DAG G has an UPSE into a convex point set S. A naive application of this idea leads to the algorithm with time complexity $O(n^7)$, since there are $O(n^2)$ distinct pairs of sources and sinks. Next we explain how the overall time complexity can be reduced to $O(n^6)$. Let $P_{s,t}$ be a path from s to t, passing through m vertices, and let $P^c_{s,t}$ be the subset of $P_{s,t}$ that contain only the cut vertices of G that are also (s,t)-separating vertices. Let first t' be the j-th vertex of $P^c_{s,t}$ that is also a sink of G. During the computation of $\mathcal{P}(a,b,m)$ corresponding to path $P_{s,t}$ we also compute $\mathcal{P}(a,b,j)$ and thus we can immediately answer whether there exists an UPSE of G into S so that s and t' is mapped to $b(S)$ and $t(S)$, respectively. Next consider a sink $\tilde{t}$ that does not belong to path $P_{s,t}$. Consider a path $P_{s,\tilde{t}}$ and its subset $P^c_{s,\tilde{t}}$ that contains only the cut vertices that are also (s,t)-separating vertices. Assume that the last common vertex of $P^c_{s,t}$ and $P^c_{s,\tilde{t}}$ is the j-th vertex of $P^c_{s,t}$. In order to compute whether there is an UPSE of G into S so that s and $\tilde{t}$ are mapped to $b(S)$ and $t(S)$, respectively, we can start the computations of Algorithm 2 determined by variable k from the $j+1$-th step (see line 4 of the algorithm). Thus, for a single source s and all possible sinks variable k changes $O(n)$ times, since there are $O(n)$ cut vertices in G. Since the number of different sources is $O(n)$ we conclude that the overall time complexity of the algorithm is $O(n^6)$. □

6. Upward planar straight-line point set embeddability is $\mathcal{NP}$-complete

In this section we examine the complexity of testing whether a given n-vertex upward planar digraph G admits an UPSE into a point set S. We show that the problem is $\mathcal{NP}$-complete even for a single-source and single-sink upward planar digraph and a 2-convex point set. The problem is trivially in $\mathcal{NP}$. In order to prove the $\mathcal{NP}$-completeness, we construct a reduction from the 3-Partition problem. The proof is along the same lines as the proof by Cabello [4] for the undirected case and therefore is sketched.

Problem: 3-Partition
Input: A bound $B \in \mathbb{Z}^+$, and a set $A = \{a_1,\dots,a_{3m}\}$ with $a_i \in \mathbb{Z}^+$, $\frac{B}{4} < a_i < \frac{B}{2}$.
Output: m disjoint sets $A_1,\dots,A_m \subset A$ with $|A_i| = 3$ and $\sum_{a\in A_i} a = B$, $1 \leqslant i \leqslant m$.

Theorem 12. *Given an n-vertex upward planar digraph G and a planar point set S of size n in general position, the decision problem of whether there exists an UPSE of G into S is $\mathcal{NP}$-complete. The decision problem remains $\mathcal{NP}$-complete even when G has a single source and a single sink and S is a 2-convex point set.*

Sketch of proof. Given instance of 3-Partition, we construct the graph G as follows. Graph G consists of: (1) a source s and a sink t, (2) paths $B_1,\dots,B_{3m}$, so that path B_i has a_i vertices, $1 \leqslant i \leqslant 3m$ and is connected to s and t as indicated in Fig. 22, (3) triangles u_i, v_i, w_i, $1 \leqslant i \leqslant m-1$, which are also connected to s and t as indicated in Fig. 22.

The point set S is illustrated in Fig. 23. It contains two extremal points, $b(S)$ and $t(S)$. The remaining points lie on two curves $\mathcal{Q}_1$ and $\mathcal{Q}_2$, such that any set of points laying on these curves is in convex position. More specifically, curve $\mathcal{Q}_2$ contains $Bm + 2(m-1)$ points: $\bigcup_{i=1}^m C_i \cup \{p_i, q_i\colon\ 1 \leqslant i \leqslant m-1\}$. The remaining points $r_1,\dots,r_{m-1}$ are situated on the curve $\mathcal{Q}_1$ so that point r_i lies in the triangle defined by p_i, q_i, $t(S)$, $1 \leqslant i \leqslant m-1$.

It is easy to see that in any upward planar embedding of G into point set S it is true that: (i) vertices s, t are mapped to points $b(S)$ and $t(S)$, respectively, (ii) vertices v_i, w_i are mapped to points p_i, q_i, $1 \leqslant i \leqslant m-1$, up to enumeration, (iii) vertex u_i is mapped to point r_i, (iv) each path B_i, $1 \leqslant i \leqslant 3m$, is mapped to consecutive points of C_j, $1 \leqslant j \leqslant m-1$. Thus the paths which are mapped to a single set C_j contain exactly B vertices. This concludes the proof. □

We further note that the proof in the first version of this paper uses a different construction (see also [15]) which yields that the problem is $\mathcal{NP}$-complete even for digraphs with longest simple cycle of length 4. The proof sketched above was suggested to us by a reviewer. While the longest simple cycle of constructed graph has length 6, it has the advantage that it is biconnected. Moreover, the point set in the current construction is 2-convex, which makes the $\mathcal{NP}$-hardness proof a tight result in combination with the algorithm given in Section 5.

7. Conclusion

The main contribution of this paper is the study of the problem of testing whether an upward planar digraph admits an upward planar embedding into a given point set. We provide a complete picture for the complexity of the problem regarding the structure of the given point set, by proving that the problem can be solved in polynomial time for a convex point set while it is $\mathcal{NP}$-complete for a 2-convex point set.

Note that the complexity of the problem varies depending on the structure of digraphs under consideration. Thus, each single-source upward planar digraph with no cycle of length greater than three admits an upward planar embedding into

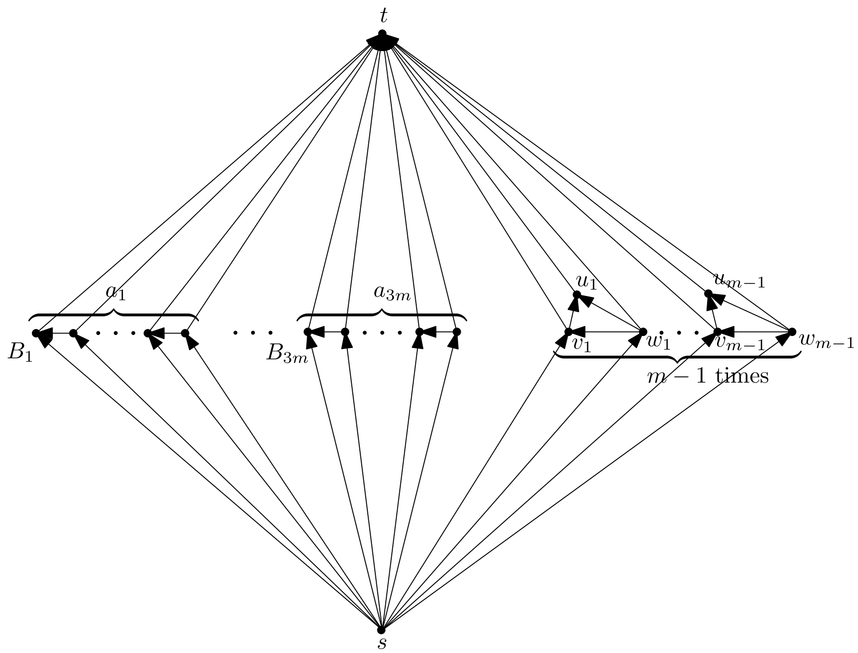

Fig. 22. The graph G of the construction used in the proof of $\mathcal{NP}$-completeness.

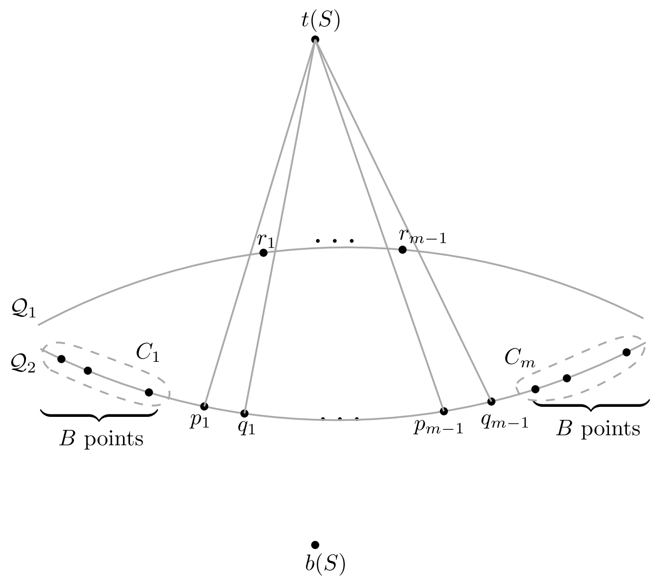

Fig. 23. The point set S of the construction used in the proof of $\mathcal{NP}$-completeness.

every point set in general position (see [13]). On the other hand, the problem is $\mathcal{NP}$-hard for a single-source digraph with the longest cycle of length four (see [15]) and for a single-source, single-sink digraph with a longest cycle of length six (see Theorem 12). Thus, it might be interesting to investigate the complexity of the problem for the class of single-source, single-sink upward planar digraphs with the longest cycle of length 5.

Acknowledgements

We thank Markus Geyer for the useful discussions during the work on this paper. We are also grateful to anonymous referees for their useful suggestions which helped us to significantly improve the results and the presentation of the paper.

References

[1] P. Gritzmann, B.M.J. Pach, R. Pollack, Embedding a planar triangulation with vertices at specified positions, Amer. Math. Monthly 98 (1991) 165–166.
[2] P. Bose, On embedding an outer-planar graph in a point set, Comput. Geom. 23 (2002) 303–312.
[3] P. Bose, M. McAllister, J. Snoeyink, Optimal algorithms to embed trees in a point set, J. Graph Algorithms Appl. 1 (1997) 1–15.
[4] S. Cabello, Planar embeddability of the vertices of a graph using a fixed point set is np-hard, J. Graph Algorithms Appl. 10 (2006) 353–363.
[5] M. Badent, E. Di Giacomo, G. Liotta, Drawing colored graphs on colored points, Theoret. Comput. Sci. 408 (2008) 129–142.
[6] E.D. Giacomo, W. Didimo, G. Liotta, H. Meijer, F. Trotta, S.K. Wismath, k-Colored point-set embeddability of outerplanar graphs, J. Graph Algorithms Appl. 12 (2008) 29–49.
[7] J. Halton, On the thickness of graphs of given degree, Inform. Sci. 54 (1991) 219–238.
[8] M. Kaufmann, R. Wiese, Embedding vertices at points: Few bends suffice for planar graphs, J. Graph Algorithms Appl. 6 (2002) 115–129.
[9] J. Pach, R. Wenger, Embedding planar graphs at fixed vertex locations, Graphs Combin. 17 (2001) 717–728.
[10] F. Giordano, G. Liotta, T. Mchedlidze, A. Symvonis, Computing upward topological book embeddings of upward planar digraphs, in: T. Tokuyama (Ed.), ISAAC, in: Lecture Notes in Comput. Sci., vol. 4835, Springer, 2007, pp. 172–183.
[11] F. Giordano, G. Liotta, S. Whitesides, Embeddability problems for upward planar digraphs, in: I. Tollis, M. Patrignani (Eds.), Graph Drawing, in: Lecture Notes in Comput. Sci., vol. 5417, Springer, Berlin/Heidelberg, 2009, pp. 242–253.
[12] T. Mchedlidze, A. Symvonis, On ρ-constrained upward topological book embeddings, in: D. Eppstein, E.R. Gansner (Eds.), Graph Drawing, in: Lecture Notes in Comput. Sci., vol. 5849, Springer, 2009, pp. 411–412.
[13] P. Angelini, F. Frati, M. Geyer, M. Kaufmann, T. Mchedlidze, A. Symvonis, Upward geometric graph embeddings into point sets, in: U. Brandes, S. Cornelsen (Eds.), Graph Drawing, in: Lecture Notes in Comput. Sci., vol. 6502, Springer, Berlin/Heidelberg, 2011, pp. 25–37.
[14] C. Binucci, E.D. Giacomo, W. Didimo, A. Estrella-Balderrama, F. Frati, S.G. Kobourov, G. Liotta, Upward straight-line embeddings of directed graphs into point sets, Comput. Geom. 43 (2010) 219–232.
[15] M. Geyer, M. Kaufmann, T. Mchedlidze, A. Symvonis, Upward point-set embeddability, in: I. Cerna, T. Gyimothy, J. Hromkovic, K. Jefferey, R. Kralovic, M. Vukolic, S. Wolf (Eds.), SOFSEM 2011: Theory and Practice of Computer Science, in: Lecture Notes in Comput. Sci., vol. 6543, Springer, Berlin/Heidelberg, 2011, pp. 272–283.
[16] M. Kaufmann, T. Mchedlidze, A. Symvonis, Upward point set embeddability for convex point sets is in P, in: M.J. van Kreveld, B. Speckmann (Eds.), Graph Drawing, in: Lecture Notes in Comput. Sci., vol. 7034, Springer, 2011, pp. 403–414.
[17] L.S. Heath, S.V. Pemmaraju, A.N. Trenk, Stack and queue layouts of directed acyclic graphs: Part I, SIAM J. Comput. 28 (1999) 1510–1539.