Upward Pointset Embeddings of Planar st-Graphs

Carlos Alegría
Roma Tre University, Rome, Italy

Susanna Caroppo
Roma Tre University, Rome, Italy

Giordano Da Lozzo
Roma Tre University, Rome, Italy

Marco D'Elia
Roma Tre University, Rome, Italy

Giuseppe Di Battista
Roma Tre University, Rome, Italy

Fabrizio Frati
Roma Tre University, Rome, Italy

Fabrizio Grosso
Roma Tre University, Rome, Italy

Maurizio Patrignani
Roma Tre University, Rome, Italy

Abstract

We study upward pointset embeddings (UPSEs) of planar st-graphs. Let G be a planar st-graph and let $S \subset \mathbb{R}^2$ be a pointset with $|S| = |V(G)|$. An *UPSE* of G on S is an upward planar straight-line drawing of G that maps the vertices of G to the points of S. We consider both the problem of testing the existence of an UPSE of G on S (UPSE Testing) and the problem of enumerating all UPSEs of G on S. We prove that UPSE Testing is NP-complete even for st-graphs that consist of a set of directed st-paths sharing only s and t. On the other hand, for n-vertex planar st-graphs whose maximum st-cutset has size k, we prove that it is possible to solve UPSE Testing in $\mathcal{O}(n^{4k})$ time with $\mathcal{O}(n^{3k})$ space, and to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time. Moreover, for an n-vertex st-graph whose underlying graph is a cycle, we provide a necessary and sufficient condition for the existence of an UPSE on a given poinset, which can be tested in $\mathcal{O}(n \log n)$ time. Related to this result, we give an algorithm that, for a set S of n points, enumerates all the non-crossing monotone Hamiltonian cycles on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(n^2)$ space, after $\mathcal{O}(n^2)$ set-up time.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Upward pointset embeddings, planar st-graphs, st-cutset, non-crossing monotone Hamiltonian cycles, graph drawing enumeration

Digital Object Identifier 10.4230/LIPIcs.GD.2024.24

Related Version *Full Version*: `https://arxiv.org/abs/2408.17369` [1]

Funding This research was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND).

1 Introduction

Given an n-vertex upward planar graph G and a set S of n points in the plane, an *upward pointset embedding* (UPSE) of G on S is an upward planar drawing of G where the vertices are mapped to the points of S and the edges are represented as straight-line segments. The Upward Pointset Embeddability Testing Problem (UPSE Testing) asks whether an upward planar graph G has an UPSE on a given pointset S.

Pointset embedding problems are classic challenges in Graph Drawing and have been considered for both undirected and directed graphs. For an undirected graph, a *pointset embedding* (PSE) has the same definition of an UPSE, except that the drawing must be planar, rather than upward planar. The Pointset Embeddability Testing Problem (PSE Testing) asks whether a planar graph has a PSE on a given pointset S. Pointset embeddings have been studied by several authors. It is known that a graph admits a PSE on *every* pointset in general position if and only if it is outerplanar [12, 26]; such a PSE can be constructed efficiently [7, 8, 9, 10]. PSE Testing is, in general, NP-complete [11], however it is polynomial-time solvable if the input graph is a planar 3-tree [35]. More in general, a polynomial-time algorithm for PSE Testing exists if the input graph has a fixed embedding, bounded treewidth, and bounded face size [5]. PSE becomes NP-complete if one of the latter two conditions does not hold. PSEs have been studied also for dynamic graphs [15, 16].

The literature on UPSEs is not any less rich than the one on PSEs. From a combinatorial perspective, the directed graphs with an UPSE on a one-sided convex pointset have been characterized [6, 27]; all directed trees are among them. Conversely, there exist directed trees that admit no UPSE on certain convex pointsets [6]. Directed graphs that admit an UPSE on any convex pointset, but not on any pointset in general position, exist [3]. It is still unknown whether every digraph whose underlying graph is a path admits an UPSE on every pointset in general position, see, e.g., [33]. UPSEs where bends along the edges are allowed have been studied in [6, 24, 25, 30, 31]. From the computational complexity point of view [28, 29], it is known that UPSE Testing is NP-hard, even for planar st-graphs and 2-convex pointsets, and that UPSE Testing can be solved in polynomial time if the given poinset is convex.

Our contributions. We tackle UPSE Testing for planar st-graphs. Let G be an n-vertex planar st-graph and S be a set of n points in the plane. We adopt the common assumption in the context of upward pointset embeddability, see e.g. [3, 6, 28, 29], that no two points of S lie on the same horizontal line. Our results are the following:

- In Section 3, we show that UPSE Testing is NP-hard even if G consists of a set of internally-disjoint st-paths (Theorem 1). A similar proof shows that UPSE Testing is NP-hard for directed trees consisting of a set of directed root-to-leaf paths (Theorem 2). This answers an open question from [4] and strengthens a result therein, which shows NP-hardness for directed trees with multiple sources and with a prescribed mapping for a vertex.
- In Section 4, we show that UPSE Testing can be solved in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space, where k is the size of the largest st-cutset of G (Theorem 7). This parameter measures the "fatness" of the digraph and coincides with the length of the longest directed path in the dual [18]. By leveraging on the techniques developed for the testing algorithm, we also show how to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time (Theorem 8).
- In Section 5, we provide a simple characterization of the pointsets in general position that allow for an UPSE of G, if G consists of two (internally-disjoint) st-paths. Based on that, we provide an $\mathcal{O}(n \log n)$ testing algorithm for this case (Theorem 9).
- Finally, in Section 6, inspired by the fact that an UPSE of a planar st-graph composed of two st-paths defines a non-crossing monotone Hamiltonian cycle on S, we provide an algorithm that enumerates all the non-crossing monotone Hamiltonian cycles on a given pointset with $\mathcal{O}(n)$ worst-case delay, and $\mathcal{O}(n^2)$ space usage and set-up time (Theorem 10).

Concerning our last result, we remark that a large body of research has considered problems related to enumerating and counting non-crossing structures on a given pointset [2, 13, 22, 32, 36]. Despite this effort, the complexity of counting the non-crossing Hamiltonian cycles, often called *polygonalizations*, remains open [20, 32, 34]. However, it is possible to enumerate all polygonalizations of a given pointset in singly-exponential time [37, 38]. Recently, an algorithm has been shown [21] to enumerate all polygonalizations of a given pointset in time polynomial in the output size, i.e., bounded by a polynomial in the number of solutions. However, an enumeration algorithm with polynomial (in the input size) delay is not yet known, neither in the worst-case nor in the average-case acception. Our enumeration algorithm achieves this goal for the case of monotone polygonalizations.

Because of space limitations, some proofs are sketched or omitted. They can be found in the full version of the paper [1].

2 Preliminaries

We use standard terminology in graph theory [19] and graph drawing [17]. For an integer $k > 0$, let $[k]$ denote the set $\{1, \ldots, k\}$. A *permutation with repetitions* of k elements from U is an arrangement of any k elements of a set U, where repetitions are allowed.

We denote by $\mathcal{CH}(S)$ the convex hull of a set S of points and by $\mathcal{B}(S)$ its boundary. The points of S with lowest and highest y-coordinates are the *south* and *north extreme* of S, respectively; together, they are the *extremes* of S. The *left envelope* of S is the subpath $\mathcal{E}_L(S)$ of $\mathcal{B}(S)$ to the left of the line through the extremes of S (including the extremes of S). The *right envelope* $\mathcal{E}_R(S)$ of S is defined analogously. We denote the subset of S in $\mathcal{E}_L(S)$ and $\mathcal{E}_R(S)$ by $\mathcal{H}_\mathrm{L}(S)$ and $\mathcal{H}_\mathrm{R}(S)$, respectively. A ray is *upward* if it passes through points whose y-coordinate is larger than the one of the starting point of the ray.

A polyline $(p_1, \ldots, p_k)$ is *y-monotone* if $y(p_i) < y(p_{i+1})$, for $i = 1, \ldots, k-1$. A *monotone path* on a pointset S is a y-monotone polyline $(p_1, \ldots, p_k)$ such that the points $p_1, \ldots, p_k$ belong to S. A *monotone cycle* on S consists of two monotone paths on S that share their endpoints. A *monotone Hamiltonian cycle* $(p_1, \ldots, p_k, p_1)$ on S is a monotone cycle on S such that each point of S is a point p_i (and vice versa).

A path $(v_1, \ldots, v_k)$ is *directed* if, for $i = 1, \ldots, k-1$, the edge (v_i, v_{i+1}) is directed from v_i to v_{i+1}; the vertices $v_2, \ldots, v_{k-1}$ are *internal*. A *planar st-graph* is an acyclic digraph with one source s and one sink t, which admits a planar embedding in which s and t are on the boundary of the outer face. An *st-path* in a planar st-graph is a directed path from s to t. A drawing of a directed graph is *straight-line* if each edge is represented by a straight-line segment, it is *planar* if no two edges cross, and it is *upward* if every edge is represented by a Jordan arc monotonically increasing in the y-direction from the tail to the head. A digraph that admits an upward planar drawing is an *upward planar graph*. Every upward planar graph admits an upward planar straight-line drawing [18]. An *Upward Pointset Embedding* (*UPSE*, for short) of an upward planar graph G on a pointset S is an upward planar straight-line drawing of G that maps each vertex of G to a point in S. In this paper, we study the following problem.

Upward Pointset Embeddability Testing Problem (UPSE Testing)

Input:	An n-vertex upward planar graph G and a pointset $S \subset \mathbb{R}^2$ with $|S| = n$.
Question:	Does there exist an UPSE of G on S?

In the remainder, we assume that not all points in S lie on the same line, as otherwise there is an UPSE if and only if the input is a directed path. Recall that no two points in S have the same y-coordinate. Unless otherwise specified, we do not require points to be in *general position*, i.e., we allow three or more points to lie on the same line.

3 NP-Completeness of UPSE Testing

In this section we prove that UPSE Testing is NP-complete. The membership in NP is obvious, as one can non-deterministically assign the vertices of the input graph G to the points of the input pointset S and then test in polynomial time whether the assignment results in an upward planar straight-line drawing of G. In the remainder of the section, we prove that UPSE Testing is NP-hard even in very restricted cases.

We first show a reduction from 3-Partition to instances of UPSE in which the input is a planar st-graph composed of a set of internally-disjoint st-paths. An instance of 3-Partition consists of a set $A = \{a_1, \dots, a_{3b}\}$ of $3b$ integers, where $\sum_{i=1}^{3b} a_i = bB$ and $B/4 \leq a_i \leq B/2$, for $i = 1, \dots, 3b$. The 3-Partition problem asks whether A can be partitioned into b subsets $A_1, \dots, A_b$, each with three integers, so that the sum of the integers in each set A_i is B. Since 3-Partition is strongly NP-hard [23], we may assume that B is bounded by a polynomial function of b. Given an instance A of 3-Partition, we show how to construct in polynomial time, precisely $\mathcal{O}(b \cdot B)$, an equivalent instance (G, S) of UPSE Testing.

The n-vertex planar st-graph G is composed of $4b+1$ internally-disjoint st-paths. Namely, for $i = 1, \dots, 3b$, we have that G contains an *a_i-path*, i.e., a path with a_i internal vertices, and $b+1$ additional k-paths, where $k = 2B + 1$. Note that $n = 2 + (b+1)k + \sum_{i=1}^{3b} a_i = 2 + (b+1)k + bB$.

The points of S lie on the plane as follows (see Figure 1a):

- p_1 is the origin, with coordinates $(0, 0)$.
- Consider $b + 1$ upward rays $\rho_1, \dots, \rho_{b+1}$, whose starting point is p_1, such that the angles $\alpha_1, \dots, \alpha_{b+1}$ that they respectively form with the x-axis satisfy $3\pi/4 > \alpha_1 > \dots > \alpha_{b+1} > \pi/4$. Let ℓ be a line intersecting all the rays, with a positive slope smaller than $\pi/4$. For $j = 1, \dots, b+1$, place k points $p_{j,1}, \dots, p_{j,k}$ (in this order from bottom to top) along ρ_j, so that $p_{j,k}$ is on ℓ and no two points share the same y-coordinate. Observe that $p_{b+1,k}$ is the highest point placed so far.
- Place p_n at coordinates $(0, 10 \cdot y(p_{b+1,k}))$.
- Finally, for $j = 1, \dots, b$, place B points along a non-horizontal segment s_j in such a way that: (i) s_j is entirely contained in the triangle with vertices $p_{j,k}$, $p_{j+1,k}$, and p_n, (ii) for any point p on s_j, the polygonal line $\overline{p_1 p} \cup \overline{p p_n}$ is contained in the region R_j delimited by the polygon $\overline{p_1 p_{j,k}} \cup \overline{p_{j,k} p_n} \cup \overline{p_n p_{j+1,k}} \cup \overline{p_{j+1,k} p_1}$, and (iii) no two distinct points on any two segments s_i and s_j share the same y-coordinate.

Note that S has $2 + (b+1)k + bB = n$ points. The described reduction is the main ingredient for the proof of the following theorem.

▶ **Theorem 1.** UPSE Testing *is* NP*-hard even for planar st-graphs consisting of a set of directed internally-disjoint st-paths.*

Proof. First, the construction of G and S takes polynomial time. In particular, the coordinates of the points in S can be encoded with a polylogarithmic number of bits. In order to prove the NP-hardness, it remains to show that the constructed instance (G, S) of UPSE Testing is equivalent to the given instance A of 3-Partition. Refer to Figure 1b.

Suppose first that A is a positive instance of 3-Partition, that is, there exist sets $A_1, \dots, A_b$, each with three integers, such that the sum of the integers in each set A_j is B. We construct an UPSE of G on S as follows. We map s to p_1 and t to p_n. For $j = 1, \dots, b+1$, we map the k internal vertices of a k-path to the points $p_{j,1}, \dots p_{j,k}$, so that vertices that come first in the directed path have smaller y-coordinates. Furthermore, for $j = 1, \dots, b$, let $A_j = \{a_{j_1}, a_{j_2}, a_{j_3}\}$. Then we map the a_{j_1} internal vertices of an a_{j_1}-path, the a_{j_2} internal vertices of an a_{j_2}-path, and the a_{j_3} internal vertices of an a_{j_3}-path to the set of B points in the triangle with vertices $p_{j,k}$, $p_{j+1,k}$, and p_n, so that vertices that come first in the directed

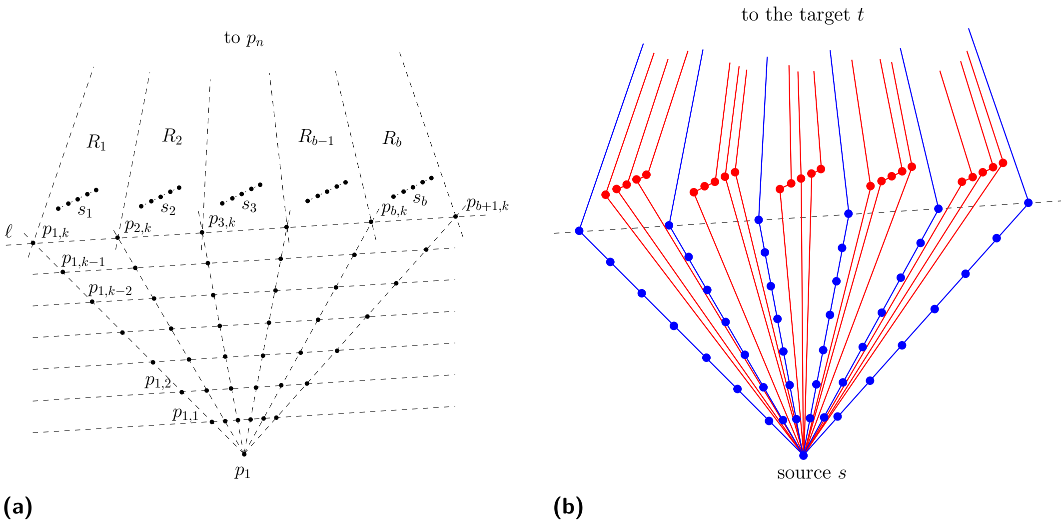


Figure 1 Illustration for the proof of Theorem 1. (a) The pointset S. (b) The UPSE of G on S.

paths have smaller y-coordinates and so that the internal vertices of the a_{j_1}-path have smaller y-coordinates than the internal vertices of the a_{j_2}-path, which have smaller y-coordinates than the internal vertices of the a_{j_3}-path. This results in an UPSE of G on S.

Suppose next that (G, S) is a positive instance of UPSE TESTING. Trivially, in any UPSE of G on S, we have that s is drawn on p_1 and t on p_n. Consider the points $p_{1,1}, \dots p_{b+1,1}$. The paths using them use all the $(b+1)k$ points $p_{j,i}$, with $j = 1, \dots, b+1$ and $i = 1, \dots, k$. Indeed, if these paths left one of such points unused, no other path could reach it from s without passing through $p_{1,1}, \dots p_{b+1,1}$, because of the collinearity of the points along the rays $\rho_1, \dots, \rho_{b+1}$. Hence, there are at most $b+1$ paths that use $(b+1)k$ points. Since the a_i-paths have less than k internal vertices, these $b+1$ paths must all be k-paths. Let $P_1, \dots, P_{b+1}$ be the left-to-right order of the k-paths around p_1. For $j = 1, \dots, b+1$, path P_j uses all points $p_{j,i}$ on ρ_j, as if P_j used a point $p_{h,i}$ with $h > j$, then two among $P_j, \dots, P_{b+1}$ would cross each other. Note that, after using $p_{j,k}$, path P_j ends with the segment $\overline{p_{j,k}p_n}$. Hence, for $j = 1, \dots, b$, the region R_j is bounded by P_j and P_{j+1}; recall that R_j contains the segment s_j. The a_i-paths must then use the points on $s_1, \dots, s_b$. Since $B/4 < a_i < B/2$, no two a_i-paths can use all the B points in one region and no four a_i-paths can lie in the same region. Hence, three a_i-paths use the B points in each region, and this provides a solution to the given 3-PARTITION instance. ◀

A similar reduction, illustrated in Figures 2a and 2b, allows us to state the following.

▶ **Theorem 2.** UPSE TESTING *is* NP*-hard even for directed trees consisting of a set of directed root-to-leaf paths.*

4 Testing and Enumeration Algorithms for Planar st-Graphs with Maximum st-Cutset of Bounded Size

An *st-cutset* of a planar st-graph $G = (V, E)$ is a subset W of E such that:

- removing W from E results in a graph consisting of exactly two connected components C_s and C_t,
- s belongs to C_s and t belongs to C_t, and
- any edge in W has its tail in C_s and its head in C_t.

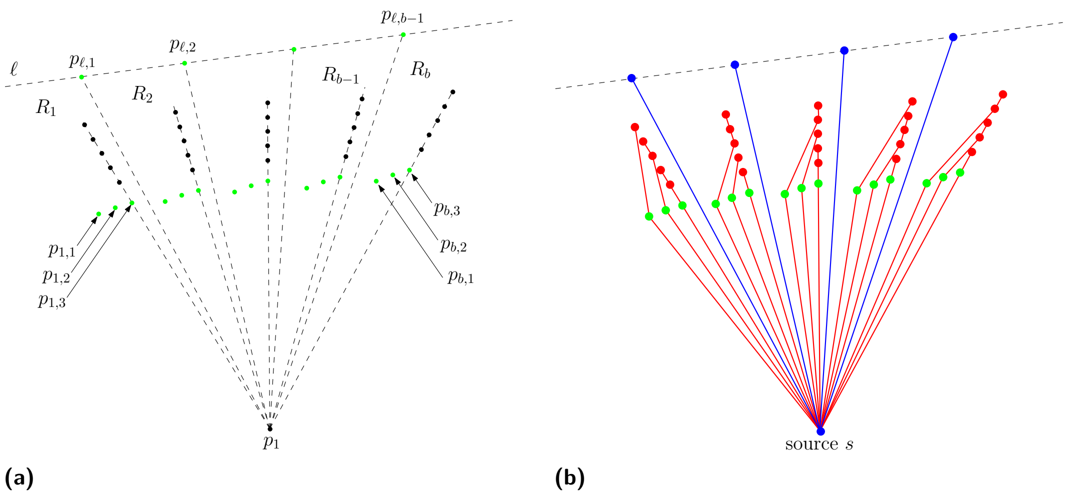


Figure 2 Illustration for the proof of Theorem 2. (a) The pointset S. The points of S visible from p_1 (green points) are as many as the children of the root of the tree T. (b) The UPSE of T on S corresponding to a solution to the original instance 3-PARTITION (red vertices).

In this section, we consider instances (G, S) where G is a planar *st*-graph, whose maximum *st*-cutset has bounded size k. In Theorem 7, we show that UPSE TESTING can be solved in polynomial time for such instances (G, S). Moreover, in Theorem 8, we show how to enumerate all UPSEs of (G, S) with linear delay. The algorithm for Theorem 7 is based on a dynamic programming approach. It exploits the property that, for an *st*-cutset W defining the connected components C_s and C_t, the extensibility of an UPSE Γ' of $C_s \cup W$ on a subset S' of S to an UPSE of G on S only depends on the drawing of the edges of W, and not on the embedding of the remaining vertices of C_s, provided that in Γ' there exists an horizontal line that crosses all the edges of W. The algorithm for Theorem 8 leverages a variation of the dynamic programming table computed by the former algorithm to efficiently test the extensibility of an UPSE of $C_s \cup W$ (in which there exists a horizontal line that crosses all the edges of W) on a subset S' of S to an UPSE of G on S.

The proofs of Theorems 7 and 8 exploit two dynamic programming tables T and Q defined as follows. Each entry of T and Q is indexed by a *key* that consists of a set of $h \leq k$ triplets $\langle e_i, p_i, q_i \rangle$, where, for any $i = 1, \dots, h$, it holds that $e_i \in E(G)$, $p_i, q_i \in S$, and $y(p_i) < y(q_i)$. Moreover, each key $\chi = \bigcup_{i=1}^{h} \langle e_i, p_i, q_i \rangle$ satisfies the following constraints:

- the set $E(\chi) = \bigcup_{i=1}^{h} e_i$ is an *st*-cutset of G and, for every i, j, with $i \neq j$, it holds true that $e_i \neq e_j$ (that is, $|E(\chi)| = h$);
- for every i, j, with $i \neq j$, it holds true that $p_i = p_j$ (resp. that $q_i = q_j$) if and only if e_i and e_j have the same tail (resp. the same head); and
- let ℓ_χ be the horizontal line passing through the tail with largest y-coordinate among the edges in $E(\chi)$, i.e., $\ell_\chi := y = y(p_i)$ s.t. $y(p_j) \leq y(p_i)$ for any $\langle e_j, p_j, q_j \rangle \in \chi$; then ℓ_χ intersects all the segments $\overline{p_j q_j}$, possibly at an endpoint.

For brevity, we sometimes say that the edge e_i has its tail (resp. its head) *mapped by* χ on p_i (resp. on q_i). We also say that e_i is *drawn as in* χ if its drawing is the segment $\overline{p_i q_i}$.

Let $\chi = \bigcup_{i=1}^{h} \langle e_i, p_i, q_i \rangle$ be a key of T and of Q; see Figure 3a. Let G_χ be the connected component containing s of the graph obtained from G by removing the edge set $E(\chi)$.

The entry $T[\chi]$ contains a Boolean value such that $T[\chi] = \texttt{True}$ if and only if there exists an UPSE of $G_\chi^+ = G_\chi \cup E(\chi)$ on some subset $S' \subset S$ with $|S'| = |V(G_\chi^+)|$ such that:

- the lowest point p_s of S belongs to S' and s lies on it, and
- for $i = 1, \dots, h$, the edge e_i is drawn as in χ.

If $T[\chi] = \texttt{False}$, the entry $Q[\chi]$ contains the empty set $\emptyset$. If $T[\chi] = \texttt{True}$ and $E(\chi)$ coincides with the set of edges incident to s, then $Q[\chi]$ stores the set $\{\perp\}$. If $T[\chi] = \texttt{True}$ and $E(\chi)$ does not coincide with the set of edges incident to s, $Q[\chi]$ stores the set Φ of keys with the following properties. Let e_τ be any edge whose tail v_τ has maximum y-coordinate among the edges in $E(\chi)$, i.e., $\langle e_\tau, p_\tau, q_\tau \rangle$ is such that $y(p_\tau) \geq y(p_j)$ for any $\langle e_j, p_j, q_j \rangle \in \chi$. For each $\varphi \in \Phi$, we have that:

- $T[\varphi] = \texttt{True}$;
- $E(\chi) \cap E(\varphi)$ contains all and only the edges in $E(\chi)$ whose tail is not v_τ, and each edge $e_i \in E(\chi) \cap E(\varphi)$ is drawn in φ as it is drawn in χ; and
- all the edges in $E(\varphi) \setminus E(\chi)$ have v_τ as their head.

Additionally, we store a list Λ of the keys σ such that $T[\sigma] = \texttt{True}$ and $E(\sigma)$ is the set of edges incident to t. Note that each edge in $E(\sigma)$ has its head mapped by σ to the point $p_t \in S$ with largest y-coordinate.

We use dynamic programming to compute the entries of T and Q in increasing order of $|V(G_\chi)|$. By the definition of T, we have that G admits an UPSE on S if and only if $\Lambda \neq \emptyset$.

First, we initialize all entries of T to $\texttt{False}$ and all entries of Q to $\emptyset$.

If $|V(G_\chi)| = 1$, then G_χ only consists of s. We set $T[\chi] = \texttt{True}$ and $Q[\chi] = \{\perp\}$ for every key $\chi = \bigcup_{i=1}^{h} \langle e_i, p_i, q_i \rangle$ such that:

- $e_1, \dots, e_h$ are the edges incident to s;
- $p_1 = \dots = p_h = p_s$; and
- for every distinct i and j in $\{1, \dots, h\}$, we have that p_s, q_i, and q_j are not aligned.

If $|V(G_\chi)| > 1$, we compute $T[\chi]$ and $Q[\chi]$ as follows, see Figure 3b. If two segments $\overline{p_i q_i}$ and $\overline{p_j q_j}$, with $i \neq j$, cross (that is, they share a point that is internal for at least one of the segments), then we leave $T[\chi]$ and $Q[\chi]$ unchanged; in particular, $T[\chi] = \texttt{False}$ and $Q[\chi] = \emptyset$. Otherwise, we proceed as follows. Let e_τ be any edge whose tail v_τ has maximum y-coordinate among the edges in $E(\chi)$. Let H^- be the set of edges obtained from $E(\chi)$ by removing all the edges having v_τ as their tail, and let H^+ be the set of edges of G having v_τ as their head. We define the set $H := H^- \cup H^+$. We have the following claim, which is illustrated in Figure 4.

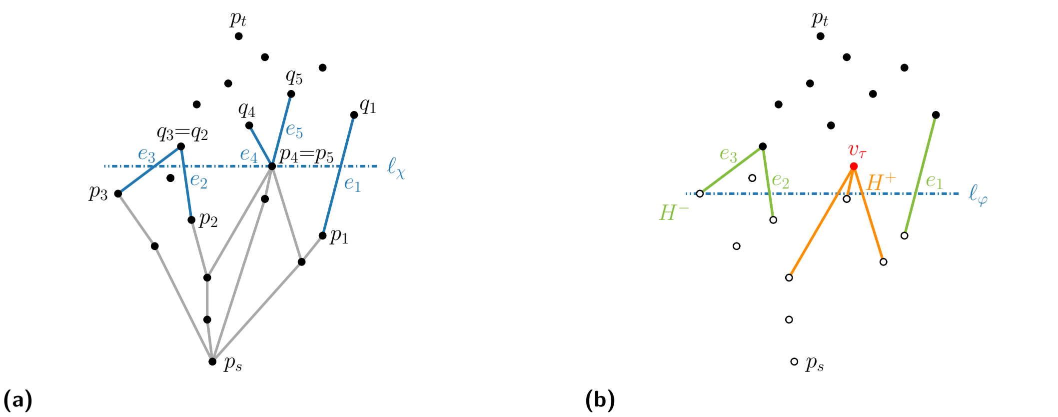


Figure 3 (a) An entry $\chi = \bigcup_{i=1}^{5} \langle e_i, p_i, q_i \rangle$ with $T[\chi] = \texttt{True}$ and a corresponding UPSE of G_χ on a subset of S that includes p_s. (b) An entry φ from which χ stems; the points in $S_\downarrow$ are filled white.

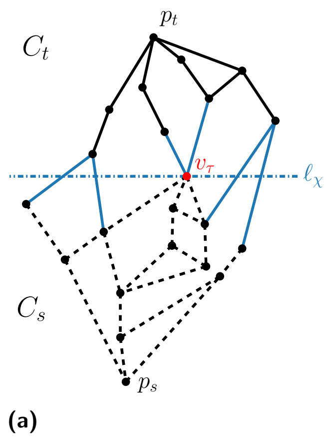


Figure 4 Illustrations for Claim 3. (a) The connected components C_s (dashed) and C_t (solid black) defined by the st-cuteset $E(\chi)$. (b) The connected components C'_s (dashed) and C'_t (solid black) defined by the st-cuteset H (blue and orange edges).

▷ Claim 3. H is an st-cutset of G.

Consider the set $S_\downarrow$ consisting of the points in S whose y-coordinates are smaller than $y(p_\tau)$. We have the following crucial observation.

▶ **Observation 4.** *$T[\chi] =$ **True** if and only if there exists some key φ, with $E(\varphi) = H$, such that $T[\varphi] =$ **True**, the edges in H^- are drawn in φ as in χ, the edges in H^+ have their heads mapped by φ on p_τ and their tails on a point in $S_\downarrow$.*

In view of Observation 4, we can now define a procedure to compute $T[\chi]$ and $Q[\chi]$. Assume that the edges $e_1, \dots, e_{|H^-|}, \dots, e_{|H|} \in H$ are ordered so that the edges of H^- precede those of H^+. By Observation 4, if $|S_\downarrow| < |H^+|$, then we leave $T[\chi]$ and $Q[\chi]$ unchanged, i.e., $T[\chi] =$ `False` and $Q[\chi] = \emptyset$. In fact, in this case, there are not enough points in $S_\downarrow$ to map the tails of the edges in H^+. Otherwise, let D be the set of all permutations with repetitions of $|H^+|$ points from $S_\downarrow$. We define a set Φ of keys that, for each $(d_1, \dots, d_{|H^+|}) \in D$, contains a key φ such that:

(i) $E(\varphi) = H$;
(ii) for any $i = 1, \dots, |H^-|$, the triple containing e_i in φ is the same as the triple containing e_i in χ (note that $e_i \in H^-$);
(iii) for any $j = |H^-|+1, \dots, |H|$, the triple containing e_j in φ has $q_j = p_\tau$, and $p_j = d_{j-|H^-|}$ (note that $e_j \in H^+$); and
(iv) for every $i = 1, \dots, |H^-|$ and $j = |H^-| + 1, \dots, |H|$, it holds $p_i = p_j$ if and only if e_i and e_j have the same tail.

Let $\Phi^{\mathtt{T}} = \{\varphi : \varphi \in \Phi \wedge T[\varphi] = \mathtt{True}\}$. By Observation 4, we have $T[\chi] =$ `True` if and only if $|\Phi^{\mathtt{T}}| \geq 1$. Thus, we set $T[\chi] = \bigvee_{\varphi \in \Phi} T[\varphi]$ and $Q[\chi] = \Phi^{\mathtt{T}}$. We say that χ *stems from* any key $\varphi \in \Phi$ with $T[\varphi] =$ `True`.

We now upper bound the sizes of T and Q and the time needed to compute them.

▷ Claim 5. Tables T and Q have size in $\mathcal{O}(n^{3k})$ and $\mathcal{O}(kn^{4k} \log n)$, respectively.

The proof of Claim 5 is based on the fact that the number of entries of T (and, thus, of Q) is bounded by $\binom{m}{k} \cdot n^k \cdot n^k \leq (mn^2)^k$. This is because an st-cutset $E(\chi)$ has size at most $h \leq k$ and because the number of permutations with repetitions of the points describing a mapping of the tails (or of the heads) of $E(\chi)$ on them is n^k. Further, $\binom{a}{b} \leq a^b$. Since each entry of T stores a single bit, we immediately have that T has $\mathcal{O}(n^{3k})$ size. Instead, each entry of Q stores at most $\mathcal{O}(n^k)$ keys of size $\mathcal{O}(k \log n)$; thus, Q has $\mathcal{O}(kn^{4k} \log n)$ size.

Computing $T[\chi]$ requires accessing the values of up to $|S_{\downarrow}|^{|H^+|} < n^k$ entries of T. Also, the time used to compute each entry $Q[\chi]$ is upper bounded by the time needed to write the $\mathcal{O}(n^k)$ keys in $Q[\chi]$, each of which has $\mathcal{O}(k \log n)$ size. Hence, we have the following.

▷ Claim 6. Tables T and Q can be computed in $\mathcal{O}(n^{4k})$ and $\mathcal{O}(kn^{4k} \log n)$ time, respectively.

Finally, recall that in order to verify whether G admits an UPSE on S, we need to check whether $\Lambda \neq \emptyset$. Computing the maximum size of an *st*-cutset of a planar *st*-graph G can be done in linear time, as it reduces to the problem of computing the length of a shortest path in the dual of any embedding of G (between the vertices representing the left and right outer faces of this embedding) [14, 18]. Therefore, the overall running time to test whether G admits an UPSE on S is dominated by the time needed to compute T, that is, $\mathcal{O}(n^{4k})$ time.

If the algorithm terminates with a positive answer, we obtain an UPSE Γ of G on S by exploiting table T. Let σ be a key in Λ. We initialize Γ to a drawing of the edges in $E(\sigma)$ as they are drawn in σ. Then we search in T a key χ with $T[\chi] = \texttt{True}$ such that σ stems from χ, and update Γ accordingly, until a key α is reached such that $T[\alpha] = \texttt{True}$ and $E(\alpha)$ is the set of edges incident to s. As the depth of the recursion is linear in the size of G and a key χ can be searched in $\mathcal{O}(n^k)$ time, we have the following.

▶ **Theorem 7.** *Let G be an n-vertex planar st-graph whose maximum st-cutset has size k and let S be a set of n points.* UPSE Testing *can be solved for (G, S) in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space; if an* UPSE *of G on S exists, it can be constructed within the same bounds.*

We describe the algorithm to enumerate all UPSEs of G on S that exploits table Q and set Λ. The algorithm defines and explores an acyclic digraph $\mathcal{D}$. The nodes of the digraph correspond to the keys χ of the table Q such that $Q[\chi] \neq \emptyset$, plus a source $n_{\mathcal{S}}$ and a sink $n_{\mathcal{T}}$. Let χ_i and χ_j be two keys of Q such that $Q[\chi_i] \neq \emptyset$ and $Q[\chi_j] \neq \emptyset$, and let $n(\chi_i)$ and $n(\chi_j)$ be the nodes corresponding to χ_i and χ_j in $\mathcal{D}$, respectively. There exists an edge from $n(\chi_i)$ to $n(\chi_j)$ in $\mathcal{D}$ if $\chi_j \in Q[\chi_i]$. Also, there exists an edge from $n_{\mathcal{S}}$ to each node $n(\sigma)$ such that $\sigma \in \Lambda$ and an edge to $n_{\mathcal{T}}$ from each node $n(\chi)$ such that $Q[\chi] = \{\perp\}$. Then $\mathcal{D}$ is an $n_{\mathcal{S}}n_{\mathcal{T}}$-graph since $n_{\mathcal{S}}$ is its unique source and $n_{\mathcal{T}}$ is its unique sink.

The algorithm performs a depth-first traversal of $\mathcal{D}$, in which every distinct $n_{\mathcal{S}}n_{\mathcal{T}}$-path corresponds to an UPSE of G on S. We initialize an UPSE Γ on S as $\Gamma = S$ (only S is drawn). When the visit traverses an edge $(n(\chi_i), n(\chi_j))$ of $\mathcal{D}$, it adds to Γ the edges in $E(\chi_j) \setminus E(\chi_i)$, drawn as in χ_j. When the traversal reaches $n_{\mathcal{T}}$, an UPSE Γ of G on S is produced. Backtracking to a node $n(\chi_i)$ along an edge $(n(\chi_i), n(\chi_j))$, the edges in $E(\chi_j) \setminus E(\chi_i)$ are removed from Γ .

To prove the correctness (see the full version of the paper [1] for a complete proof), we show that:

(i) Distinct paths from $n_{\mathcal{S}}$ to $n_{\mathcal{T}}$ in $\mathcal{D}$ correspond to different UPSEs of G on S.
(ii) For each UPSE of G on S, there exists in $\mathcal{D}$ a path corresponding to it.

Item i can be proved by contradiction: if two distinct $n_{\mathcal{S}}n_{\mathcal{T}}$-paths $\mathcal{P}_1$ and $\mathcal{P}_2$ yielded the same UPSE Γ, there would be a node $n(\chi_x)$ shared by $\mathcal{P}_1$ and $\mathcal{P}_2$ such that the nodes $n(\chi_1)$ and $n(\chi_2)$ of $\mathcal{P}_1$ and $\mathcal{P}_2$ following $n(\chi_x)$ are different. Since $n(\chi_x)$ is shared by $\mathcal{P}_1$ and $\mathcal{P}_2$, the keys χ_1 and χ_2 have the same edge-set $E(\chi_1) = E(\chi_2)$ but the tails of the edges in $E(\chi_x) \setminus E(\chi_1)$ are mapped differently, implying that the UPSEs yielded by $\mathcal{P}_1$ and $\mathcal{P}_2$ are different. To prove Item ii, we show that, if Γ is an UPSE of G on S, then there exists an

$n_{\mathcal{S}}n_{\mathcal{T}}$-path that yields Γ. For $i = 1, \dots, n$, let S_i be the set of the lowest i points of S. Also, for $i = 1, \dots, n-1$, let Γ_i be the restriction of Γ to the vertices of G mapped to S_i and to all their incident edges. We proceed by induction on i, showing that, for each Γ_i and Γ_{i+1} there exists an edge $(n(\chi_i), n(\chi_{+1}))$ in D that produces Γ_{i+1}. This involves finding a suitable st-cutset in Γ and proving that this corresponds to a key in D.

We now discuss the running time of the algorithm. Table Q can be constructed in $\mathcal{O}(kn^{4k} \log n)$ time, by Claim 6. Also, the digraph $\mathcal{D}$ can be constructed in linear time in the size of Q, which is $\mathcal{O}(kn^{4k} \log n)$ by Claim 5. Finally, we discuss the delay of our algorithm. Since the paths from $n_{\mathcal{S}}$ to $n_{\mathcal{T}}$ have $\mathcal{O}(n)$ size and since between an UPSE and the next one at most two paths are traversed, the delay of our algorithm is $\mathcal{O}(n)$. We get the following.

▶ **Theorem 8.** *Let G be a n-vertex planar st-graph whose maximum st-cut has size k and let S be a set of n points. It is possible to enumerate all* UPSE*s of G on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time.*

5 Planar st-Graphs Composed of Two st-Paths

In this section, we consider the special case of Theorem 7 in which the underlying graph of the given planar st-graph is an n-vertex cycle. Here, Theorem 7 would yield an $\mathcal{O}(n^8)$-time testing algorithm. We give a much faster algorithm based on a characterization of the positive instances, provided that the points are in general position.

▶ **Theorem 9.** *Let G be an n-vertex planar st-graph consisting of two st-paths P_L and P_R, and let S be a pointset with n points in general position. We have that G admits an* UPSE *on S with P_L to the left of P_R if and only if $|P_L| \geq |\mathcal{H}_{\mathrm{L}}(S)|$ and $|P_R| \geq |\mathcal{H}_{\mathrm{R}}(S)|$. Also, it can be tested in $\mathcal{O}(n \log n)$ time whether G admits an* UPSE *on S.*

Proof. Provided the characterization in the statement holds, we can test whether G admits an UPSE on S by computing the convex hull $\mathcal{CH}(S)$ of S (in $\mathcal{O}(n \log n)$ time), deriving the sets $\mathcal{H}_{\mathrm{L}}(S)$ and $\mathcal{H}_{\mathrm{R}}(S)$ (in $O(n)$ time, by scanning $\mathcal{CH}(S)$), and finally comparing their sizes with the ones of P_L and P_R (in $O(1)$ time). Thus, we focus on proving the characterization.

The necessity is obvious. In the remainder we prove the sufficiency by induction on the size of S (and, thus, of $V(G)$). We give some preliminary definitions; see Figures 5–7. Let p_s (p_t) be the south (north) extreme of S and let ℓ_{st} be the line through p_s and p_t. Let S_L (S_R) be the subset of S in the closed half-plane to the left (right) of ℓ_{st}, including p_s and p_t. Note that $\mathcal{H}_{\mathrm{L}}(S) \subseteq S_L$ and $\mathcal{H}_{\mathrm{R}}(S) \subseteq S_R$. Also, since S is in general position, $S_L \cap S_R = \{p_s, p_t\}$.

In the base case, it either holds that $S_L = \{p_s, p_t\}$ and $|\mathcal{H}_{\mathrm{R}}(S)| = |P_R|$, or $S_R = \{p_s, p_t\}$ and $|\mathcal{H}_{\mathrm{L}}(S)| = |P_L|$. We discuss the former case (see Figure 5), as the latter case is symmetric. In this case, an UPSE Γ of G on S can be constructed by drawing P_R as the right envelope $\mathcal{E}_R(S)$ and P_L as the y-monotone polyline connecting the point of $S_R \setminus \mathcal{H}_{\mathrm{R}}(S)$.

If the base case does not hold, we distinguish two cases based on whether both S_L and S_R contain a vertex different from p_s and p_t (**Case A**), or only one of them does (**Case B**).

If **Case A** holds, assume $|P_L| \geq |S_L|$; the case $|P_L| < |S_L|$ is symmetric, as in that case it holds true that $|P_R| \geq |S_R|$. Refer to Figure 6. Then $\mathcal{H}_{\mathrm{L}}(S)$ contains a point p different from p_s and p_t; see Figure 6a. Since by the statement $|P_L| \geq |\mathcal{H}_{\mathrm{L}}(S)|$ and $|\mathcal{H}_{\mathrm{L}}(S)| \geq 3$, we have that P_L contains at least one internal vertex. Let $S' = S \setminus \{p\}$, let P'_L be an st-path with $|P'_L| = |P_L| - 1$, and let G' be the st-graph $P'_L \cup P_R$. Since $|\mathcal{H}_{\mathrm{L}}(S')| \leq |S_L| - 1$ and $|S_L| \leq |P_L|$, we have that $|\mathcal{H}_{\mathrm{L}}(S')| \leq |P_L| - 1 = |P'_L|$. Thus, the graph G' and the pointset S' satisfy the conditions of the statement. Since $|S'| = |S| - 1$ (and $|V(G')| = |V(G)| - 1$), by induction we have that the graph G' admits an UPSE Γ' on S' (see Figure 6b). Figures 6b and 6c show how to modify Γ' to obtain an UPSE Γ of G on S.

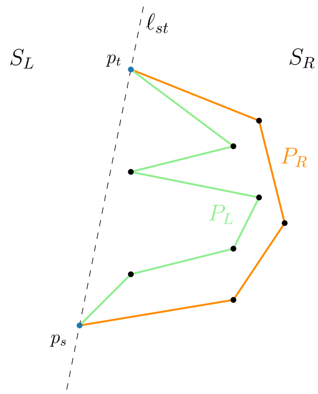
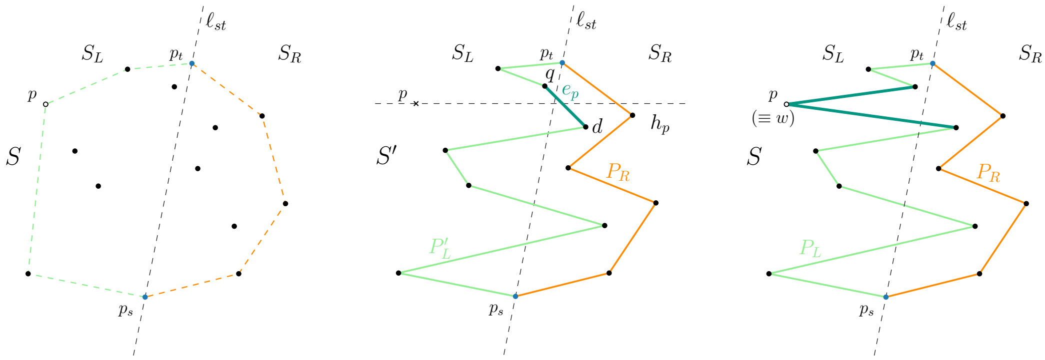
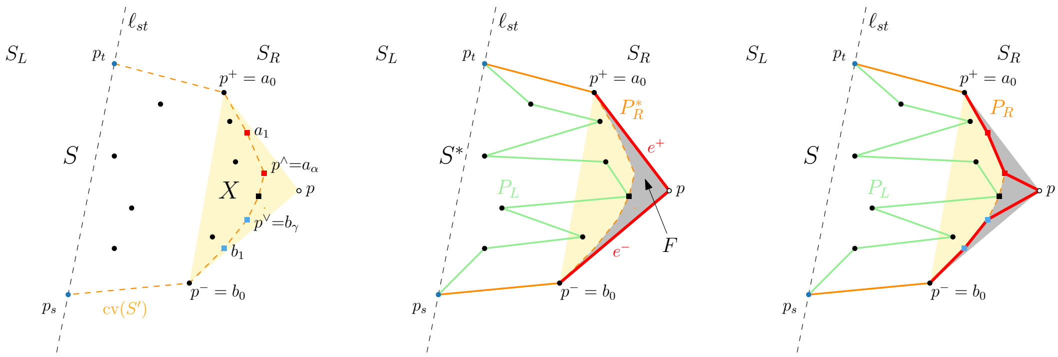


Figure 5 Illustration for the base case of Theorem 9.



(a) $\{p_s, p_t\} \subset S_L, S_R$. **(b)** UPSE Γ' of G' on S'. **(c)** UPSE Γ of G on S.

Figure 6 Illustrations for **Case A** in the proof of Theorem 9.



(a) $\{p_s, p_t\} \subset S_L, S_R$. **(b)** UPSE Γ^* of G^* on S^*. **(c)** UPSE Γ of G on S.

Figure 7 Illustrations for **Case B2** in the proof of Theorem 9.

If **Case B** holds, recall that $S_L = \{p_s, p_t\} \subset S_R$, and since the base case does not apply, we have that $|P_R| > |\mathcal{H}_{\mathrm{R}}(S)|$. Let p be any point in $\mathcal{H}_{\mathrm{R}}(S) \setminus \{p_s, p_t\}$ and $S' = S \setminus \{p\}$. By the conditions of **Case B**, the path P_R contains at least one internal vertex. Let P'_R be an st-path with $|P'_R| = |P_R| - 1$, and let G' be the st-graph $P_L \cup P'_R$. We distinguish two cases based on the size of $\mathcal{H}_{\mathrm{R}}(S')$. In **Case B1**, it holds $|P'_R| \geq |\mathcal{H}_{\mathrm{R}}(S')|$, whereas in **Case B2**, it holds $|P'_R| < |\mathcal{H}_{\mathrm{R}}(S')|$. In **Case B1**, we have that the pair (G', S') satisfies the conditions of the statement. In particular, the pair (G', S') either matches the conditions of the base case or again those of **Case B**. Thus, since $|S'| = |S| - 1$ (and $|V(G')| = |V(G)| - 1$), we can inductively construct an UPSE Γ' of G' on S', and obtain an UPSE of G on S with a redrawing similar to the one in Figure 6c. In **Case B2**, we proceed as follows; see Figure 7. Let p^+ (p^-) be the point of $\mathcal{H}_{\mathrm{R}}(S)$ with the smallest y-coordinate and above p (with the largest y-coordinate and below p). Let X be the set of points of S in the interior of the triangle Δp^+pp^- together with p^+ and p^- (but not p). Clearly, the right envelope of $\mathcal{CH}(X)$ forms a subpath of the right envelope of $\mathcal{CH}(S')$; see Figure 7a. The set $\mathcal{H}_{\mathrm{R}}(X)$ consists of p^-, p^+, and of k vertices not belonging to $\mathcal{H}_{\mathrm{R}}(S)$ (squares in Figure 7). We denote by $k^* = |P_R| - |\mathcal{H}_{\mathrm{R}}(S)|$ the number of points in the interior of $\mathcal{CH}(S)$ that need to be the image of a vertex of P_R in an UPSE of G on S. Note that $k > k^* > 0$ holds. Let $p^\wedge$ ($p^\vee$) be the point of $\mathcal{H}_{\mathrm{R}}(S')$ with the smallest y-coordinate and above p (with the largest y-coordinate and below p). Up to renaming, let $a_0 = p^+, a_1, \dots, a_\alpha = p^\wedge$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^+ to $p^\wedge$ (these points have decreasing y-coordinates). Similarly, let $b_0 = p^-, b_1, \dots, b_\gamma = p^\vee$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^- to $p^\vee$ (these points have increasing y-coordinates). We define the set $X^* \subset \mathcal{H}_{\mathrm{R}}(X)$ as follows. If $k^* \le \alpha$, then $X^* = \{a_i | 1 \le i \le k^*\}$, otherwise $X^* = \{a_i | 1 \le i \le \alpha\} \cup \{b_i | 1 \le i \le k^* - \alpha\}$. Observe that, $|X^*| = k^*$. Also, by the definition of k^*, the path P_R contains $\mathcal{H}_{\mathrm{R}}(S) - 2 + k^*$ internal vertices and since $\mathcal{H}_{\mathrm{R}}(S) \ge 3$ in **Case B**, we have that P_R contains at least $k^* + 1$ internal vertices. Let $S^* = S \setminus X^*$, let P^*_R be an st-path with $|P_R| - k^*$ vertices, and let G^* be the st-graph $P_L \cup P^*_R$. Clearly, the pair (G^*, S^*) satisfies the statement, and in particular the base case. In fact, $|P^*_R| = |P_R| - k^*$, and by the definition of k^*, we have that $|P_R| - k^* = |\mathcal{H}_{\mathrm{R}}(S)|$. Moreover, by construction, $\mathcal{H}_{\mathrm{R}}(S) = \mathcal{H}_{\mathrm{R}}(S^*)$, since the vertices of X^* lie in the interior of $\mathcal{CH}(S)$. Thus, since $|S^*| = |S| - k^*$, by induction G^* admits an UPSE Γ^* on S^*; see Figure 7b. Moreover, as the base case applies to (G^*, S^*), we have that the endpoints of the edges of P^*_R are consecutive along $\mathcal{E}_R(S)$. In particular, there exist two adjacent edges e^- and e^+ of P^*_R such that the tail of e^- is mapped to p^-, the head of e^- (i.e., the tail of e^+) is mapped to p, and the head of e^+ is mapped to p^+. Thus, it is possible to obtain an UPSE Γ of G on S from Γ^* (see Figure 7c) by replacing the drawing of the edges e^+ and e^- with a y-monotone polyline that passes through all the points in X^*. Such a polyline lies inside the region F (shaded gray in Figures 7b and 7c) obtained by subtracting from the triangle Δp^+pp^- (interpreted as a closed region) all the points of $\mathcal{CH}(X)$. In particular, observe that, in Γ^*, the region F is not traversed by any edge and that the only points of S^* that lie on the boundary of F are p and the points in $\mathcal{H}_{\mathrm{R}}(X) \setminus X^*$. ◀

6 Enumerating Non-crossing Monotone Hamiltonian Cycles

Theorem 9 allows us to test whether an n-vertex planar st-graph G composed of two st-paths can be embedded as a non-crossing monotone Hamiltonian cycle on a set S of n points. We now show an efficient algorithm for enumerating *all* the non-crossing monotone Hamiltonian cycles on S. Figure 8 shows two non-crossing monotone Hamiltonian cycles on a pointset.

▶ **Theorem 10.** *Let S be a set of n points. It is possible to enumerate all the non-crossing monotone Hamiltonian cycles on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(n^2)$ space, after $\mathcal{O}(n^2)$ set-up time.*

Let $p_1, \dots, p_n$ be the points of S, ordered by increasing y-coordinates. This order can be computed in $\mathcal{O}(n \log n)$ time. For $i \in [n]$, let $S_i = \{p_1, \dots, p_i\}$. A *bipath* B on S_i consists of two non-crossing monotone paths L and R on S_i, each of which might be a single point, such that L and R start at p_1, each point of S_i is the image of an endpoint of a segment of B, and if L and R both have at least one segment, then L is to the left of R (see Figure 9).

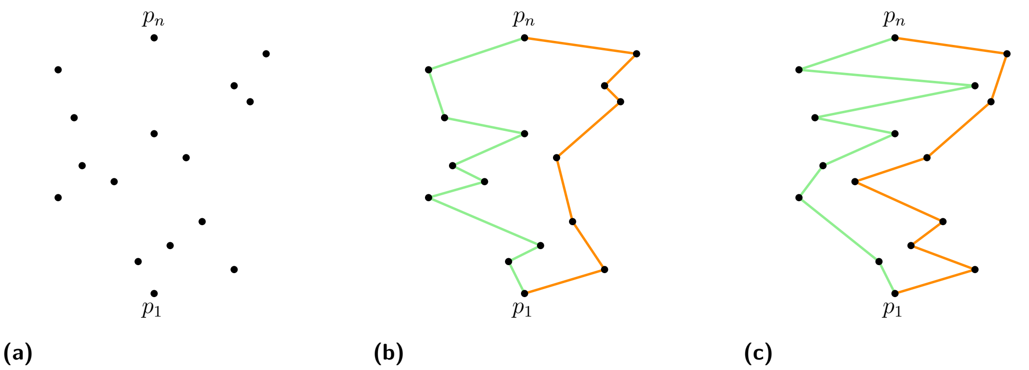


Figure 8 Two non-crossing monotone Hamiltonian cycles on the same pointset.

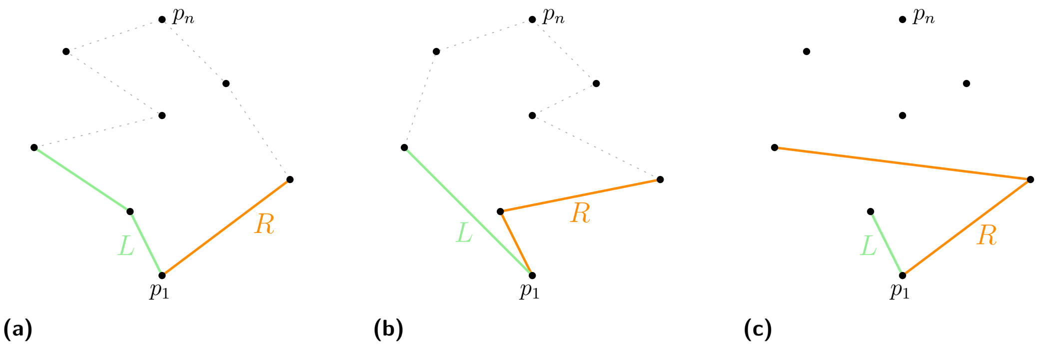


Figure 9 Three bipaths on S_4. The first two bipaths are extensible, while the third one is not.

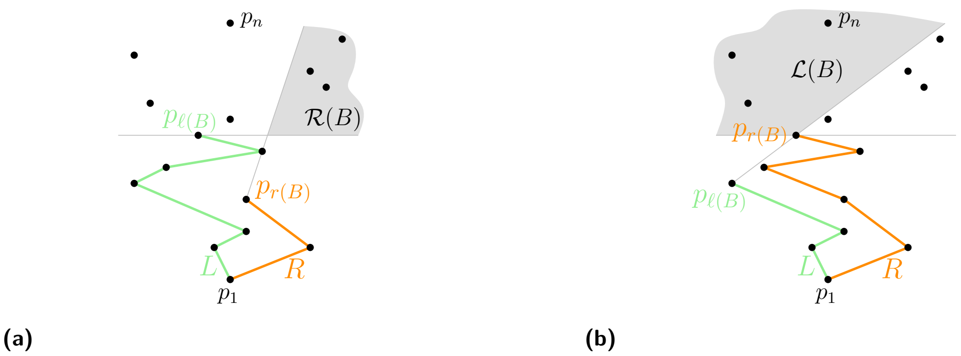


Figure 10 (a) Region $\mathcal{R}(B)$ for a bipath B. (b) Region $\mathcal{L}(B)$ for a bipath B.

We say that a bipath B is *extensible* if there exists a non-crossing monotone Hamiltonian cycle on S whose restriction to S_i is B. Consider a bipath B on S_i with $1 < i < n$. Let $p_{\ell(B)}$ and $p_{r(B)}$ be the endpoints of L and R with the highest y-coordinate, respectively. Suppose first that $\ell(B) > r(B)$. Consider the rightmost ray $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$ starting at $p_{r(B)}$ through a point of $S_{\ell(B)} \setminus S_{r(B)}$. We denote by $\mathcal{R}(B)$ the open region of the plane strictly to the right of $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$ and strictly above the horizontal line through $p_{\ell(B)}$; see Figure 10a. Similarly, if $p_{r(B)}$ is higher than $p_{\ell(B)}$, then $\mathcal{L}(B)$ is the open region of the plane strictly to the left of the leftmost ray $\ell(p_{\ell(B)}, S_{r(B)} \setminus S_{\ell(B)})$ from $p_{\ell(B)}$ through a point of $S_{r(B)} \setminus S_{\ell(B)}$ and strictly above the horizontal line through $p_{r(B)}$; see Figure 10b.

For any $i \in [n-1]$, we say that a bipath B on S_i is *safe* if:

(i) $i = 1$; or
(ii) $i > 1$, $p_{\ell(B)}$ is higher than $p_{r(B)}$, and $|\mathcal{R}(B) \cap S| \geq 1$; or
(iii) $i > 1$, $p_{r(B)}$ is higher than $p_{\ell(B)}$, and $|\mathcal{L}(B) \cap S| \geq 1$.

We have the following lemma which is proved in the full version of the paper [1].

▶ **Lemma 11.** *A bipath B is extensible if and only it is safe.*

Our enumeration algorithm implicitly defines and explores a search tree T. Its leaves have level n and correspond to non-crossing monotone Hamiltonian cycles on S. The internal nodes at level i correspond to extensible bipaths on S_i and have at most two children each. The exploration of T performed by the algorithm is a depth-first traversal. When a node μ is visited, the number of its children is established. If μ has at least one child, the visit proceeds with any child of μ. Otherwise, μ is a leaf; then the visit proceeds with any unvisited child of the ancestor of μ that has largest level, among the ancestors of μ with unvisited children.

The algorithm starts at the root of T, which corresponds to the (unique) safe bipath on S_1. At each node μ at level $i \in [n-2]$, corresponding to a bipath $B(\mu)$, we construct either one or two bipaths on S_{i+1}, associated with either one or two children of μ, respectively. Let $L(\mu)$ and $R(\mu)$ be the left and right non-crossing monotone paths composing $B(\mu)$, respectively, and let $p_{\ell(B(\mu))}$ and $p_{r(B(\mu))}$ be the endpoints of $L(\mu)$ and $R(\mu)$ with the highest y-coordinate, respectively. If $\overline{p_{\ell(B(\mu))}p_{i+1}}$ does not cross $R(\mu)$, then let $B_L = B(\mu) \cup \overline{p_{\ell(B(\mu))}p_{i+1}}$. We test whether B_L is a safe bipath and, in the positive case, add to μ a child μ_L corresponding to B_L. Analogously, if $\overline{p_{r(B(\mu))}p_{i+1}}$ does not cross $L(\mu)$, we test whether $B_R = B(\mu) \cup \overline{p_{r(B(\mu))}p_{i+1}}$ is a safe bipath and, in the positive case, add to μ a child μ_R corresponding to B_R. Note that the algorithm guarantees that each non-leaf node of T is safe, and thus, by Lemma 11, extensible. Finally, at each node μ at level $n-1$, we add a leaf λ to μ corresponding to the non-crossing monotone Hamiltonian cycle $B(\mu) \cup \overline{p_{\ell(B(\mu))}p_n} \cup \overline{p_{r(B(\mu))}p_n}$. Since μ is extensible, such a cycle is indeed non-crossing.

In order to complete the proof of Theorem 10, we show what follows:
(i) Each node of T at level $i \neq n$ is internal.
(ii) Each leaf corresponds to a non-crossing monotone Hamiltonian cycle on S.
(iii) Distinct leaves correspond to different non-crossing monotone Hamiltonian cycles on S.
(iv) For each non-crossing monotone Hamiltonian cycle on S, there exists a leaf of T corresponding to it.
(v) Using $\mathcal{O}(n^2)$ pre-processing time and $\mathcal{O}(n^2)$ space, the algorithm enumerates each non-crossing monotone Hamiltonian cycle on S with $\mathcal{O}(n)$ delay.

- To prove Item i, we show that the leaves of T have all level n. Consider a node μ of T with level $i < n-1$, we prove that it has a child in T. Recall that $B(\mu)$ is safe, otherwise it would not had been added to T, and thus, by Lemma 11, it is extensible. Hence, there exists a non-crossing monotone Hamiltonian cycle C on S whose restriction to S_i is $B(\mu)$. Also, the restriction of C to S_{i+1} is a bipath $B'(\mu)$ on S_{i+1} which coincides with $B(\mu)$, except that it contains either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$. Since $B'(\mu)$ is the restriction of C to S_{i+1}, it is extensible and thus, by Lemma 11, it is safe. It follows that μ has a child corresponding to $B'(\mu)$, which is inserted in T when adding either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$ to $B(\mu)$. The proof that a node with level $n-1$ is not a leaf is analogous.
- To prove Item ii, consider a leaf λ and its parent μ in T. Note that μ is associated with a safe bipath $B(\mu)$ on S_{n-1}; by Lemma 11, we have that $B(\mu)$ is extensible. Since $B(\mu)$ is extensible, the (unique) monotone Hamiltonian cycle on S whose restriction to S_{n-1} is $B(\mu)$ is non-crossing. This cycle corresponds to λ and is added to T when visiting μ.

- To prove Item iii, suppose for a contradiction that there exist two leaves λ_1 and λ_2 associated with two monotone Hamiltonian cycles C_1 and C_2, respectively, with $C_1 = C_2$. Let μ be the lowest common ancestor of λ_1 and λ_2 in T. Let j be the level of μ. Denote by μ_i the child of μ leading to λ_i, with $i \in \{1, 2\}$. By the construction of T, we have that exactly one of the bipaths $B(\mu_1)$ and $B(\mu_2)$ contains the segment $\overline{p_{\ell(B(\mu))}p_{j+1}}$, while the other one contains the segment $\overline{p_{r(B(\mu))}p_{j+1}}$. This contradicts the fact that $C_1 = C_2$.
- To prove Item iv, let C be a non-crossing monotone Hamiltonian cycle on S. Consider the safe bipath B on S_{n-1} obtained by removing from C the point p_n, together with its two incident segments. It suffices to show that T contains a node μ such that $B = B(\mu)$. In fact, in this case, μ is an extensible node of level $n-1$ whose unique child in T is the leaf corresponding to C. To prove that T contains such a node μ, we prove by induction that, for every level $i = 1, \dots, n-1$, the tree T contains a node corresponding to the restriction B_i of B to S_i. The base case trivially holds. For the inductive case, suppose that T contains a node ν whose associated bipath $B(\nu)$ is B_{i-1}. Then B_i is obtained by adding either the segment $\overline{p_{\ell(B(\nu))}p_i}$ or the segment $\overline{p_{r(B(\nu))}p_i}$ to B_{i-1}. Since B_i is extensible, by Lemma 11 it is safe, and hence ν has a child in T corresponding to B_i.
- Finally, we discuss Item v. To this aim, we compute in $\mathcal{O}(n^2)$ time two tables C and D of $\mathcal{O}(n^2)$ size that allow us to test in $\mathcal{O}(1)$ time whether a bipath B on S_i, with $i \in \{2, \dots, n-1\}$, can be extended to a bipath on S_{i+1} and whether B is safe. The tables C and D are indexed by triples $\langle p_a, p_b, X\rangle$, where $p_a, p_b \in S$ with $a < b$ and $X \in \{L, R\}$. Each entry of C contains a Boolean value $C[p_a, p_b, X]$ that is set to `True` if and only if the segment $\overline{p_a p_{b+1}}$ does not cross any bipath B on S_b composed of two monotone st-paths L and R respectively ending at points p_a and p_b (if $X = L$) or respectively ending at points p_b and p_a (if $X = R$). Each entry of D contains a Boolean value $D[p_a, p_b, X]$ that is set to `True` if and only if the open region that is (i) strictly to the right of the rightmost (if $X = R$, or leftmost if $X = L$) ray starting at p_a and passing through a point in $S_b \setminus S_a$ and (ii) strictly above the horizontal line through p_b contains a point of S. For each fixed $a \in [n-2]$ and $X \in \{L, R\}$, we compute all the entries $C[p_a, p_b, X]$ and $D[p_a, p_b, X]$ with $b = a+1, a+2, \dots, n-1$ in overall $\mathcal{O}(n)$ time. This sums up to $\mathcal{O}(n^2)$ time over all the entries of C and of D. The query time of C and D, together with the fact that T has n levels, implies that the algorithm's delay is in $\mathcal{O}(n)$. More details can be found in the full version of the paper [1].

Items i–iv prove the correctness of the enumeration algorithm, while Item v proves its efficiency. This concludes the proof of Theorem 10.

7 Conclusions and Open Problems

We addressed basic pointset embeddability problems for upward planar graphs. We proved that UPSE testing is NP-hard even for planar st-graphs composed of internally-disjoint st-paths and for directed trees composed of directed root-to-leaf paths. For planar st-graphs, we showed that UPSE Testing can be solved in $O(n^{4k})$ time, where k is the maximum st-cutset of G, and we provided an algorithm to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay. We also showed how to enumerate all monotone polygonalizations of a given pointset with $\mathcal{O}(n)$ worst-case delay. We point out the following open problems.

- Our NP-hardness proofs for UPSE testing use the fact that the points are not in general position. Given a directed tree T on n vertices and a set S of n points *in general position*, is it NP-hard to decide whether T has an UPSE on S?

- Can UPSE TESTING be solved in polynomial time or does it remain NP-hard if the input is a *maximal* planar st-graph?
- We proved that UPSE TESTING for a planar st-graph is in XP with respect to the size of the maximum st-cutset of G. Is the problem in FPT with respect to the same parameter? Are there other interesting parameterizations for the problem?
- Let S be a pointset and $\mathcal{P}$ be a non-crossing path on a subset of S. Is it possible to decide in polynomial time whether $\mathcal{P}$ can be extended to a polygonalization of S? A positive answer would imply an algorithm with polynomial delay for enumerating the polygonalizations of a pointset, with the same approach as the one we adopted in this paper for monotone polygonalizations.

References

1 Carlos Alegria, Susanna Caroppo, Giordano Da Lozzo, Marco D'Elia, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, and Maurizio Patrignani. Upward pointset embeddings of planar st-graphs. *CoRR*, abs/2408.17369, 2024. arXiv:2408.17369.

2 Victor Alvarez, Karl Bringmann, Radu Curticapean, and Saurabh Ray. Counting triangulations and other crossing-free structures via onion layers. *Discret. Comput. Geom.*, 53(4):675–690, 2015. doi:10.1007/S00454-015-9672-3.

3 Patrizio Angelini, Fabrizio Frati, Markus Geyer, Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. Upward geometric graph embeddings into point sets. In Ulrik Brandes and Sabine Cornelsen, editors, *18th International Symposium on Graph Drawing (GD 2010)*, volume 6502 of *LNCS*, pages 25–37. Springer, 2010. doi:10.1007/978-3-642-18469-7_3.

4 Elena Arseneva, Pilar Cano, Linda Kleist, Tamara Mchedlidze, Saeed Mehrabi, Irene Parada, and Pavel Valtr. Upward point set embeddings of paths and trees. In Ryuhei Uehara, Seok-Hee Hong, and Subhas C. Nandy, editors, *15th International Conference and Workshops on Algorithms and Computation (WALCOM 2021)*, volume 12635 of *LNCS*, pages 234–246. Springer, 2021. doi:10.1007/978-3-030-68211-8_19.

5 Therese Biedl and Martin Vatshelle. The point-set embeddability problem for plane graphs. In Tamal K. Dey and Sue Whitesides, editors, *28th ACM Symposium on Computational Geometry (SoCG 2012)*, pages 41–50. ACM, 2012. doi:10.1145/2261250.2261257.

6 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Alejandro Estrella-Balderrama, Fabrizio Frati, Stephen G. Kobourov, and Giuseppe Liotta. Upward straight-line embeddings of directed graphs into point sets. *Comput. Geom.*, 43(2):219–232, 2010. doi:10.1016/J.COMGEO.2009.07.002.

7 Prosenjit Bose. On embedding an outer-planar graph in a point set. In Giuseppe Di Battista, editor, *5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *LNCS*, pages 25–36. Springer, 1997. doi:10.1007/3-540-63938-1_47.

8 Prosenjit Bose. On embedding an outer-planar graph in a point set. *Comput. Geom.*, 23(3):303–312, 2002. doi:10.1016/S0925-7721(01)00069-4.

9 Prosenjit Bose, Michael McAllister, and Jack Snoeyink. Optimal algorithms to embed trees in a point set. In Franz-Josef Brandenburg, editor, *Symposium on Graph Drawing (GD '95)*, volume 1027 of *LNCS*, pages 64–75. Springer, 1995. doi:10.1007/BFB0021791.

10 Prosenjit Bose, Michael McAllister, and Jack Snoeyink. Optimal algorithms to embed trees in a point set. *J. Graph Algorithms Appl.*, 1(2):1–15, 1997. doi:10.7155/JGAA.00002.

11 Sergio Cabello. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. *J. Graph Algorithms Appl.*, 10(2):353–363, 2006. doi:10.7155/JGAA.00132.

12 Netzahualcoyotl Castañeda and Jorge Urrutia. Straight line embeddings of planar graphs on point sets. In Frank Fiala, Evangelos Kranakis, and Jörg-Rüdiger Sack, editors, *8th Canadian Conference on Computational Geometry (CCCG 1996)*, pages 312–318. Carleton University Press, 1996. URL: http://www.cccg.ca/proceedings/1996/cccg1996_0052.pdf.

13 Gi-Sang Cheon, Hong Joon Choi, Guillermo Esteban, and Minho Song. Enumeration of bipartite non-crossing geometric graphs. *Discret. Appl. Math.*, 317:86–100, 2022. doi:10.1016/J.DAM.2022.04.008.

14 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 4th Edition.* MIT Press, 2022. URL: https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/.

15 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. In Patrizio Angelini and Reinhard von Hanxleden, editors, *30th International Symposium on Graph Drawing and Network Visualization (GD 2022)*, volume 13764 of *LNCS*, pages 289–303. Springer, 2022. doi:10.1007/978-3-031-22203-0_21.

16 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. *J. Graph Algorithms Appl.*, 27(8):651–677, 2023. doi:10.7155/JGAA.00639.

17 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, 1999.

18 Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988. doi:10.1016/0304-3975(88)90123-5.

19 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

20 David Eppstein. Counting polygon triangulations is hard. *Discret. Comput. Geom.*, 64(4):1210–1234, 2020. doi:10.1007/S00454-020-00251-7.

21 David Eppstein. Non-crossing hamiltonian paths and cycles in output-polynomial time. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.SOCG.2023.29.

22 Philippe Flajolet and Marc Noy. Analytic combinatorics of non-crossing configurations. *Discret. Math.*, 204(1-3):203–229, 1999. doi:10.1016/S0012-365X(98)00372-0.

23 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, first edition edition, 1979.

24 Emilio Di Giacomo, Henry Förster, Daria Kokhovich, Tamara Mchedlidze, Fabrizio Montecchiani, Antonios Symvonis, and Anaïs Villedieu. On 1-bend upward point-set embeddings of st-digraphs. In José A. Soto and Andreas Wiese, editors, *16th Latin American Symposium on Theoretical Informatics (LATIN 2024), Part I*, volume 14578 of *LNCS*, pages 3–18. Springer, 2024. doi:10.1007/978-3-031-55598-5_1.

25 Francesco Giordano, Giuseppe Liotta, Tamara Mchedlidze, Antonios Symvonis, and Sue Whitesides. Computing upward topological book embeddings of upward planar digraphs. *J. Discrete Algorithms*, 30:45–69, 2015. doi:10.1016/J.JDA.2014.11.006.

26 Peter Gritzmann, Bojan Mohar, János Pach, and Richard Pollack. Embedding a planar triangulation with vertices at specified points. *The American Mathematical Monthly*, 98(2):165, 1991. doi:10.2307/2323956.

27 Lenwood S. Heath, Sriram V. Pemmaraju, and Ann N. Trenk. Stack and queue layouts of directed acyclic graphs: Part I. *SIAM J. Comput.*, 28(4):1510–1539, 1999. doi:10.1137/S0097539795280287.

28 Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. Upward point set embeddability for convex point sets is in P. In Marc J. van Kreveld and Bettina Speckmann, editors, *19th International Symposium on Graph Drawing (GD 2011)*, volume 7034 of *LNCS*, pages 403–414. Springer, 2011. doi:10.1007/978-3-642-25878-7_38.

29 Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. On upward point set embeddability. *Comput. Geom.*, 46(6):774–804, 2013. doi:10.1016/J.COMGEO.2012.11.008.

30 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. In Jan Kratochvíl, editor, *7th International Symposium on Graph Drawing (GD'99)*, volume 1731 of *LNCS*, pages 165–174. Springer, 1999. doi:10.1007/3-540-46648-7_17.

31 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *J. Graph Algorithms Appl.*, 6(1):115–129, 2002. doi:10.7155/JGAA.00046.

32 Dániel Marx and Tillmann Miltzow. Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *LIPIcs*, pages 52:1–52:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.SOCG.2016.52.

33 Tamara Mchedlidze. Upward planar embedding of an n-vertex oriented path on $O(n^2)$ points. *Comput. Geom.*, 46(8):1003–1008, 2013. doi:10.1016/J.COMGEO.2013.05.004.

34 Joseph S. B. Mitchell and Joseph O'Rourke. Computational geometry column 42. *Int. J. Comput. Geom. Appl.*, 11(5):573–582, 2001. doi:10.1142/S0218195901000651.

35 Rahnuma Islam Nishat, Debajyoti Mondal, and Md. Saidur Rahman. Point-set embeddings of plane 3-trees. *Comput. Geom.*, 45(3):88–98, 2012. doi:10.1016/J.COMGEO.2011.09.002.

36 Andreas Razen and Emo Welzl. Counting plane graphs with exponential speed-up. In Cristian S. Calude, Grzegorz Rozenberg, and Arto Salomaa, editors, *Rainbow of Computer Science - Dedicated to Hermann Maurer on the Occasion of His 70th Birthday*, volume 6570 of *LNCS*, pages 36–46. Springer, 2011. doi:10.1007/978-3-642-19391-0_3.

37 Manuel Wettstein. Counting and enumerating crossing-free geometric graphs. *J. Comput. Geom.*, 8(1):47–77, 2017. doi:10.20382/JOCG.V8I1A4.

38 Katsuhisa Yamanaka, David Avis, Takashi Horiyama, Yoshio Okamoto, Ryuhei Uehara, and Tanami Yamauchi. Algorithmic enumeration of surrounding polygons. *Discret. Appl. Math.*, 303:305–313, 2021. doi:10.1016/J.DAM.2020.03.034.