Characterization of Unlabeled Level Planar Graphs

J. Joseph Fowler* and Stephen G. Kobourov*

Department of Computer Science, University of Arizona
{jfowler,kobourov}@cs.arizona.edu

Technical Report 06-04
September 14, 2007

Abstract. We present the set of planar graphs that always have a simultaneous geometric embedding with a strictly monotone path on the same set of n vertices, for any of the $n!$ possible mappings. These graphs are equivalent to the set of unlabeled level planar (ULP) graphs that are level planar over all possible labelings. Our contributions are twofold. First, we provide linear time drawing algorithms for ULP graphs. Second, we provide a complete characterization of ULP graphs by showing that any other graph must contain a subgraph homeomorphic to one of seven forbidden graphs.

1 Introduction

Simultaneous embedding enables the visualization of multiple graphs on the same set of vertices. In order to preserve the "mental map," graphs are overlaid so that corresponding vertices have the same location. The mapping between vertices may be fixed, or may not be given, or may change and dynamically evolve as in the case of colored simultaneous embeddings [1]. To accommodate this, we consider all possible 1-1 mappings between graphs. Embeddings that use no edge bends and in which no pair of edges of the same graph cross are known as simultaneous geometric embeddings [2].

Determining which graphs share a simultaneous geometric embedding has proved difficult. While Geyer *et al.* [6] have shown this cannot always be done for tree-tree pairs, the question remains open for tree-path pairs. Estrella *et al.* [5] partially answer this question by characterizing the set of trees that have a simultaneous geometric embedding with a strictly monotone path. We now extend those results by characterizing the set of all planar graphs that have a simultaneous geometric embedding with a strictly monotone path. The importance of this result lies in the fact that all positive results showing that certain pairs of graphs allow simultaneous geometric embeddings rely on reducing at least one of the graphs in the pair under consideration to a path which is realized in strictly monotone fashion. Thus, our result captures the largest possible class of graphs that can be embedded using this technique.

* This work is supported in part by NSF grants CCF-0545743 and ACR-0222920.

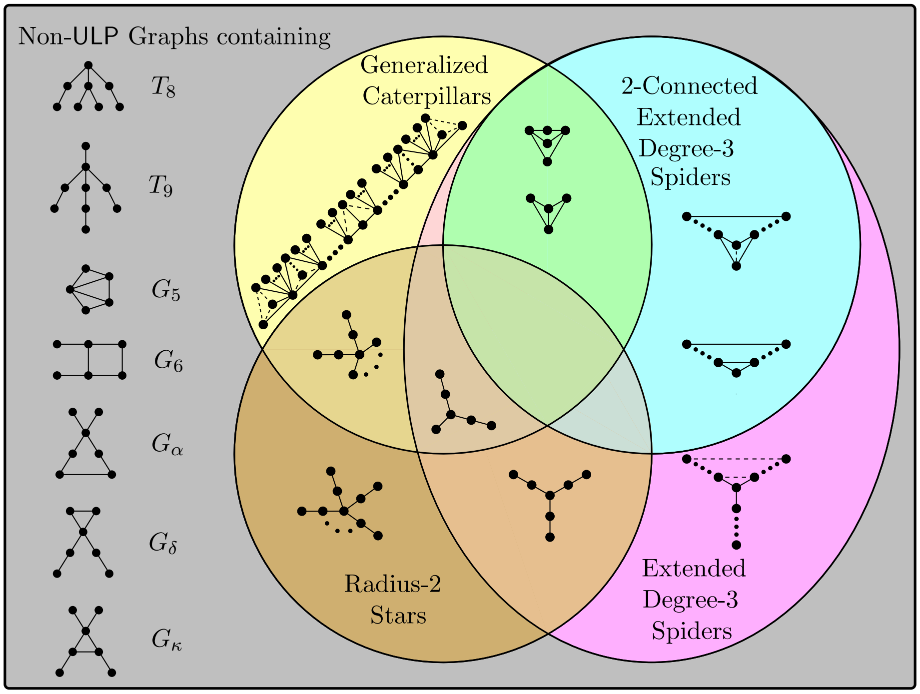


Fig. 1. A Venn diagram of the set of graphs characterized by the seven forbidden graphs T_8, T_9, G_5, G_6, G_α, G_δ, and G_κ in $\mathcal{F}$. Graphs that do not contain a subgraph homeomorphic to of any of these are generalized caterpillars, radius-2 stars, and extended degree-3 spiders with the subcategory of 2-connected extended degree-3 spiders.

Rotating or stretching a drawing along a single direction does not affect crossings. As a result, we assume that the path will be drawn in a zig-zag fashion with a difference of $+1$ between the y-coordinates of two successive vertices. This allows us to frame the problem of drawing the planar graph in terms of placing the vertices along a set of parallel horizontal lines, called tracks, with one vertex per track. For an n-vertex planar graph, we label the vertices from 1 to n in which the label is the y-coordinate. If a planar graph has a straight-line drawing without crossings for all $n!$ permutations of the labels, then it has a simultaneous geometric embedding with a strictly monotone path for any mapping.

A related problem is that of level planarity [8]. Our labeling forms a partition of vertices into levels with one vertex per level. If we consider a graph in which the y-coordinate of each level is distinct and all the edges are y-monotone, then we have a level drawing. If the drawing is planar, then the graph is level planar for that labeling. If this holds for each of the $n!$ labelings, then the graph is unlabeled level planar (ULP). ULP graphs are precisely those that have a simultaneous geometric embedding with strictly monotone paths for any labeling. Hence, we can also phrase our problem in terms of level planarity.

Any graph for which this cannot be done must have some subgraph homeomorphic to a forbidden graph, or obstruction, that will induce a crossing when drawn on tracks for a particular labeling. In this paper we show that ULP graphs fall into three categories: *radius*-2 *stars*, *generalized caterpillars*, and *extended degree*-3 *spiders*. Furthermore, we show how to simultaneously embed any ULP graph with a monotone path in linear time. Finally, we complete the characterization in terms of a minimal set of seven forbidden graphs, $\mathcal{F} := \{T_8, T_9, G_5, G_6, G_\alpha, G_\delta, G_\kappa\}$; see Fig. 1.

2 Preliminaries

Two planar n-vertex graphs $G_1(V, E_1)$ and $G_2(V, E_2)$ have a *simultaneous embedding with mapping* if they can be drawn in the xy-plane with bijection $f : V \mapsto V$ in which v and $f(v)$ have the same xy-coordinates while maintaining the planarity of each graph. If this can be done for some bijection f, then G_1 and G_2 are *simultaneously embeddable*. If edges of both E_1 and E_2 are drawn with straight-line edges, then G_1 and G_2 have a *simultaneous geometric embedding*.

Let an n-vertex graph $G(V, E)$ have a labeling $\phi : V \mapsto [1..n]$ in which $\phi(u) \neq \phi(v)$ for all $(u, v) \in E$. A horizontal line $\ell_j = \{(x, j) \,|\, x \in \mathbb{R}\}$ for some $j \in [1..n]$ is *track* j. In a *realization* of G, each vertex $v \in V$ is placed along track $\phi(v)$ and each edge (u, v) is strictly y-monotone. Edge bends $b_1, b_2, \ldots, b_k$ may naturally occur at any point edge (u, v) intersects a track provided $\phi(u) < \phi(b_1) < \cdots < \phi(b_k) < \phi(v)$ or $\phi(u) > \phi(b_1) > \cdots > \phi(b_k) > \phi(v)$ in which b_1 is adjacent to u, b_k is adjacent to v, and b_i lies between b_{i-1} and b_{i+1} for $1 < i < k$.

A realization without crossings is a *planar realization* of G. A planar realization with one straight-line segment for each edge (u, v) is a *straight-line planar realization* of G. While any planar realization with bends can be "stretched out" in the x-direction to form a straight-line planar realization in $O(n)$ time as shown by Eades *et al.* [4], the area of the realization can become exponential.

A *level graph* $G(V, E, \phi)$ is a *directed* graph with *leveling* $\phi : V \mapsto [1..k]$ that assigns every vertex to one of k levels so that $\phi(u) < \phi(v)$ for every edge (u, v). In a *level drawing* all vertices in a level have the same y-coordinate and each edge is y-monotone. If the level drawing can be drawn without crossings, then G is *level planar*. The level planarity of G for a given leveling is independent of

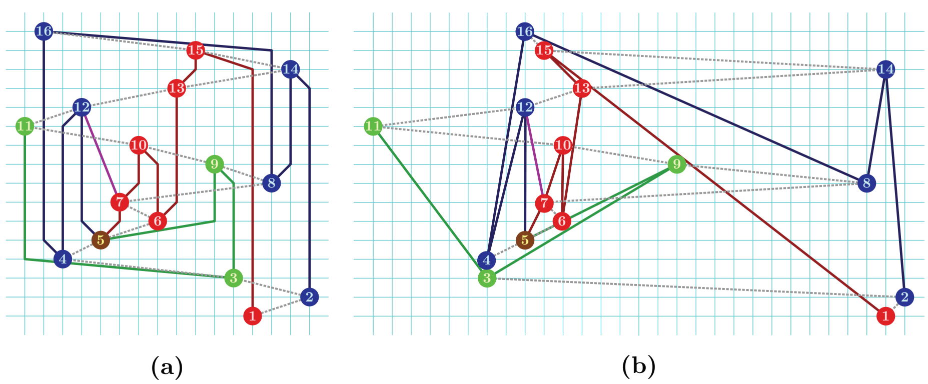


Fig. 2. Simultaneous embeddings of a path and a ULP tree with and without bends.

its orientation: First take an n-vertex undirected graph G. Then label G with labeling $\phi : V \mapsto [1..n]$. Next orient each edge (u, v) of G so that $\phi(u) < \phi(v)$ to form the level graph $\tilde{G}(V, \tilde{E}, \phi)$ with the leveling ϕ on n levels with one vertex per level. Then ask is $\tilde{G}$ level planar? If yes, repeat this process for all other labelings of G. If one never encounters a level nonplanar graph, the graph G is called *unlabeled level planar* (ULP). Hence, a ULP graph has a simultaneous embedding with a strictly y-monotone path for any labeling ϕ; see Fig 2.

The vertices placed along a track correspond to the levels in a level graph. An undirected graph with a labeling ϕ has a "planar realization" if and only if the corresponding level graph is "level planar". These two terms are interchangeable only if edge bends do not matter. If we need a simultaneous geometric embedding we use the more restrictive term "straight-line planar realization".

A *chain* C of G is a simple path denoted $v_1 – v_2 – \cdots – v_t$. The vertices of C are denoted $V(C)$. A vertex v of C is *ϕ-minimal* (or *ϕ-maximal*) if it has a minimal (or maximal) track number of all the vertices of $V(C)$. Such a vertex is *ϕ-extreme* if it is ϕ-minimal or ϕ-maximal.

In a graph $G(V, E)$, *subdividing* an edge $(u, v) \in E$ replaces edge (u, v) with the pair of edges (u, w) and (w, v) in E by adding w to V. A *subdivision* of G is a graph obtained by performing a series of subdivisions of G. A graph $G(V, E)$ is *isomorphic* to a graph $\tilde{G}(\tilde{V}, \tilde{E})$ if there exists a bijection $f : V \mapsto \tilde{V}$ such that $(u, v) \in E$ if and only if $\big(f(u), f(v)\big) \in \tilde{E}$. A graph $G(V, E)$ is *homeomorphic* to a graph $\tilde{G}(\tilde{V}, \tilde{E})$ if a subdivision of G is isomorphic to a subdivision of $\tilde{G}$. The *distance* between vertices u and v in a graph is the length of the shortest path from u to v. The *eccentricity* of a vertex v is the greatest distance to any other vertex. The *radius* of a graph is the minimum eccentricity of any vertex.

A *leaf vertex* is any degree-1 vertex. A *caterpillar* is a tree in which the removal of all leaf vertices yields a path (the empty graph is a special case of a path). The remaining path forms the *spine*. A *lobster* is a tree in which the removal of all leaf vertices yields a caterpillar. A *claw* is a $K_{1,3}$, whereas, a *star* is a $K_{1,k}$ for some $k \geq 3$. A *double star* is a star in which each edge has been subdivided once. A *radius-2 star* (R-2S) is any subgraph of a double star with radius 2. A *degree-3 spider* is an arbitrarily subdivided claw. The following six

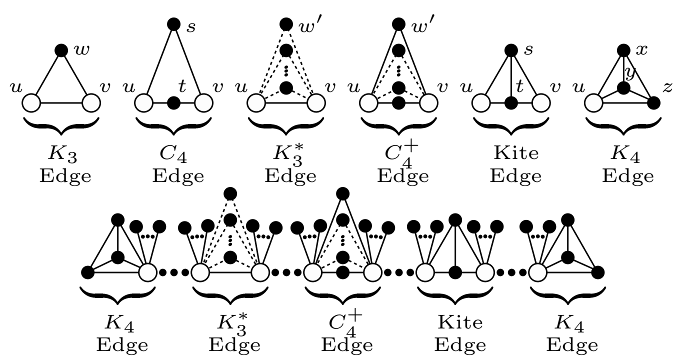


Fig. 3. The six types of H edges used to from a GC on the second line.

types of "edges" in Fig. 3 allow us to generalize a caterpillar and to extend a degree-3 spider to include cycles.

Definition 1

(a) A K_3 edge *is the cycle* u–v–w–u *on vertices* $\{u, v, w\}$
(b) A C_4 edge *is the cycle* u–s–v–t–u *on vertices* $\{u, v, s, t\}$.
(c) A kite edge *is the cycle* u–s–v–t–u *with edge* s–t *on vertices* $\{u, v, s, t\}$.
(d) A K_3^* edge *is set of cycles* u–v–w'–u *with edge* u–v *on vertices* $\{u, v\} \cup W$ *where* $w' \in W$ *for some possibly empty vertex set* W.
(e) A C_4^+ edge *is set of cycles* u–w–v–w'–u *on vertices* $\{u, v, w\} \cup W$ *where* $w' \in W$ *for some non-empty vertex set* W.
(f) A K_4 edge *is the complete graph on the vertices* $\{u, x, y, z\}$.

Definition 2 *A* generalized caterpillar *(GC) is a caterpillar in which each edge* u'–v' *along the* spine *can be replaced with a* K_3^*, C_4^+, *or kite edge (and the two edges at the end of the spine can also be replaced by a* K_4 *edge) in which vertex* u *(and* v *if present) replaces vertex* u' *(and* v'*); see Fig. 3.*

Definition 3

(a) A 1-connected extended degree-3 spider *(1-C E3-S) is a degree-3 spider with two optional edges connecting*
  (i) two of three vertices adjacent to the degree-3 vertex and
  (ii) two of the three leaf vertices; see Fig. 4(a).
(b) A 2-connected extended degree-3 spider *(2-C E3-S) is a cycle or a cycle with one* K_3, C_4 *or kite edge, see Fig. 4(b).*
(c) A extended degree-3 spider *(E3-S) is either a 1-connected extended degree-3 spider or a 2-connected extended degree-3 spider.*

These definitions allows us to make the following observation.

Observation 4 *Every spanning tree of a GC is a caterpillar. Every spanning tree of a E3-S is a degree-3 spider or a path.*

3 Graphs with Planar Realizations on Tracks

In this section we show that radius-2 stars (R-2S), generalized caterpillars (GC), and extended degree-3 spiders (E3-S) are level planar for any labeling. We do this by presenting linear time algorithms for straight-line, crossings-free drawing of any such graph on the tracks determined by its labeling. More formally, we show that $\mathcal{P} = \{G : G$ is a R-2S, GC, or E3-S$\}$ is ULP.

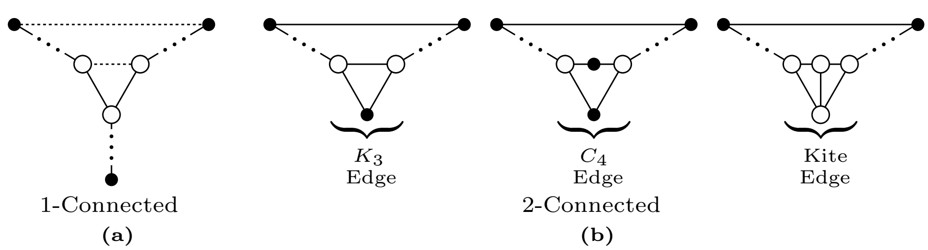


Fig. 4. A extended degree-3 spider is either (a) a 1-C E3-S or (b) a 2-C E3-S.

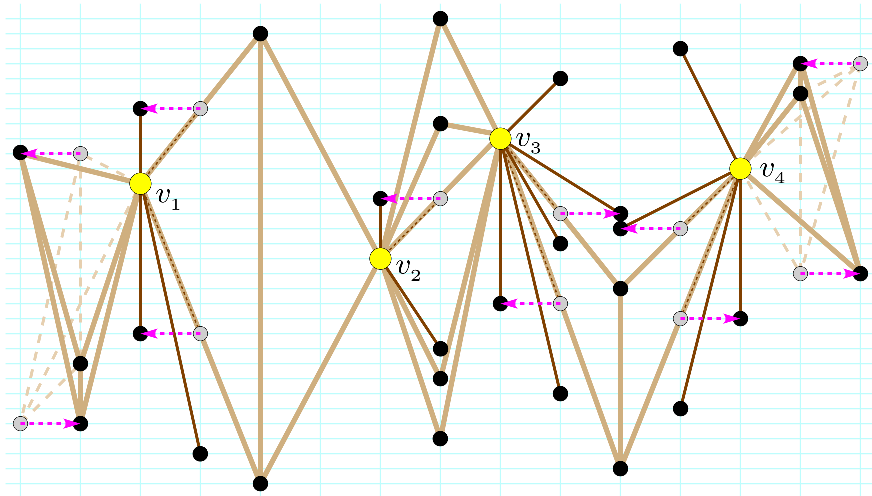


Fig. 5. The gray vertices are initial locations of vertices in a straight-line planar realization of a GC on a 14×32 grid. The arrows avoid crossings or overlapping edges. The K_4 edges incident to v_1 and v_4 show initial locations with dashed edges leading to crossings that are eliminated by switching the location of the two incident vertices.

The next lemma from [5] shows this for a R-2S.

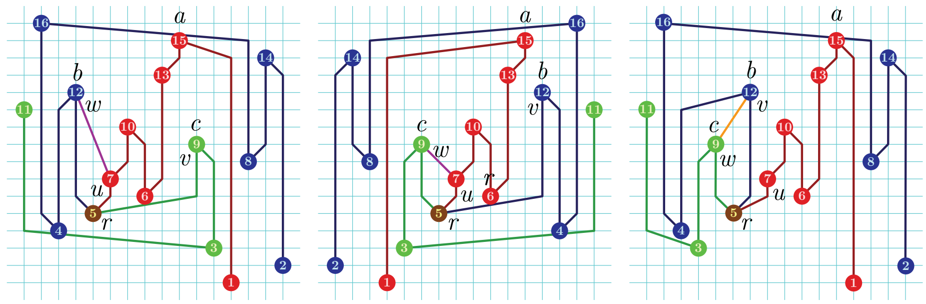
Lemma 5 *(**Lemma 4** of [5]) An n-vertex radius-2 star can be straight-line planarly realized in $O(n)$ time on a $(2n+1) \times n$ grid for any labeling.*

The following lemmas show how a GC and the two types of a E3-S also have compact planar realizations on tracks.

Lemma 6 *An n-vertex generalized caterpillar can be straight-line planarly realized in $O(n)$ time within an $n \times n$ grid for any labeling.*

Proof. We first obtain the cut vertices of the GC using the vertices of its spanning tree, which must be a caterpillar by Observation 4, as candidates. With these we can draw each incident K_3^*, C_4^+, kite, and K_4 spine edgeusing $2 \times n$ space for each one except a kite edge that requires $4 \times n$ space proceeding left to right along the spine as shown in Fig. 5. A K_4 edge on can be drawn without crossings by swapping vertices as in Fig. 5.

If we were not constrained to an integer grid, one could place all the incident edges with leaf vertices in a sufficiently narrow region above and below each cut vertex. Being restricted to integer coordinates, we shift the endpoint of a leaf vertex left or right by one space as needed to avoid overlapping edges. In general we attempt to draw the leaf vertices one to the left of the adjacent cut vertex except for the last vertex in which we drawn one to the right to allow for a final K_4 edge. While a C_4^+ cannot have an overlapping, a K_3^* edge can have one overlapping edge in which case the leaf is moved directly above or below the cut vertex. A kite edge is slightly more involved in that it can have two overlapping edge, and if both leaf vertices are above or below the adjacent cut vertex, then one of of the two leaf vertices can always be moved directly above or below the middle edge of the kite edge. □



(a) **(b)** **(c)**

Fig. 6. Examples of three 1-C E 3-Ss on 16×16 grids. The only difference is the edge between one pair of the three vertices adjacent to the root. If this edge is incident to u, the first vertex along chain with the vertex a, case (i) applies as in (a) and (b). Otherwise, case (ii) applies as in (c).

Lemma 7 *An n-vertex 1-connected extended degree-3 spider can be planarly realized in $O(n)$ time on an $n \times n$ grid for any labeling.*

Proof. We show how to draw G on tracks with at most one bend per edge for a labeling ϕ. We first draw a subgraph that is a degree-3 spider T with an extra edge in G between two of three vertices adjacent to the root vertex r (the unique degree-3 vertex) of T. Next, we accommodate an extra edge in G connecting two leaf vertices of T.

Let T' be portion of the T drawn so far. We maintain two invariants:

(1) two of the leaf vertices v_{min} and v_{max} of T' are ϕ-extreme and

(2) T' only intersects the track of the third leaf vertex v_{mid} either to the left or right of v_{mid}.

Provided these invariants hold, we keep placing the next vertex v adjacent to v_{mid} in $T - T'$ one space to the left or right of T' at x-coordinate v_x depending on which side of the track of v_{mid} that T' intersects. By (2), T' does not intersect one side of the track of v_{mid}. Whenever we draw from v to w (in this case $w = v_{\text{mid}}$), we bend the edge at $\big(v_x,\, \phi(w){-}1\big)$ if $\phi(v) < \phi(w)$ and at $\big(v_x,\, \phi(w){+}1\big)$ otherwise. We keep doing this until v becomes ϕ-extreme. Either v_{min} or v_{max} becomes v_{mid}. Since that vertex was previously ϕ-extreme by invariant (1), T' now only intersects its track either to the left or right, maintaining invariant (2).

We observe that we do not need to actually detect which side of a track that T' intersects. Rather we simply alternate directions each time we switch the chain being extended. For instance, if T' intersects the track of v_{mid} to the left, we continue extending the chain to the right until it intersects the track of v_{min} or v_{max} to the right. Then we can safely extend the next chain to the left, and so on. Once a chain is exhausted, we can safely extend the remaining two chains to the left and right respectively.

We start drawing T until both invariants hold for T'. Place r at $\big(0,\, \phi(r)\big)$. Let $\{u, v, w\}$ be the neighbors of r in T. Let v_{min}, v_{mid} and v_{max} be these vertices such that $\phi(v_{\text{min}}) < \phi(v_{\text{mid}}) < \phi(v_{\text{max}})$. If $\phi(v_{\text{min}}) < \phi(r) < \phi(v_{\text{max}})$, drawing edges from r to vertices at $\big(-1,\, \phi(v_{\text{min}})\big)$, $\big(1,\, \phi(v_{\text{max}})\big)$, and $\big(2,\, \phi(v_{\text{mid}})\big)$ satisfies both invariants. In this case, we can also add a straight-line edge between any one pair of $\{u, v, w\}$. Otherwise, suppose w.l.o.g that $\phi(r) < \phi(v_{\min})$. Let $\{a, b, c\}$ be the ϕ-maximal vertices of the portions of the chains in T from r to the point each chain crosses the track of r such that $\phi(a) > \phi(b) > \phi(c)$. Assume w.l.o.g. that u is first vertex of the chain with a. There are two cases:

(i) If edge (v, w) is not in G, assume w.l.o.g. edge (u, w) is in G. Extend the chain starting with u to the right of r until it reaches a becoming $v_{\max}$. Place v one right of a with an edge bend at $\big(v_x,\ \phi(r) + 1\big)$.

(ii) If edge (v, w) is in G, then assume w.l.o.g. v is the first vertex of the chain with b. Extend this chain to the right until it reaches b. Place u one right of b with an edge bend at $\big(u_x,\ \phi(r) + 1\big)$ and continue to extend the chain to the right until it reaches a becoming $v_{\max}$.

Place w at $\big(-1,\ \phi(w)\big)$ and extend the chain to the left until it becomes $v_{\min}$. Edge (u, w) or (v, w) can be drawn with a straight-line edge since u or v is one right of r. In both cases, invariants (1) and (2) hold; see Fig. 6.

If an edge connects two leaf vertices to form a cycle C in T, we first draw subtree $\tilde{T}$ in which two leaf vertices $c_{\min}$ and $c_{\max}$ of $\tilde{T}$ are the ϕ-extreme vertices of C. The above algorithm ensures the other chain of $\tilde{T}$ only intersects the tracks of $c_{\min}$ and $c_{\max}$ to the right or left, blocking one direction, but not both. Whichever $c_{\min}$ or $c_{\max}$ is leftmost or rightmost of $\tilde{T}$, say that $c_{\min}$ is rightmost, we extend the rest of C from $c_{\min}$ right until reaching v adjacent to $c_{\max}$. Then we draw an edge from v to $c_{\max}$ with a bend at $\big(v_x,\ \phi(c_{\max}) - 1\big)$. □

We next give a similar realization of a 2-C E 3-S with bends—the difference being that most edges are straight except for one or two edges that might require a bend.

Lemma 8 *An n-vertex 2-connected extended degree-3 spider can be planarly realized in $O(n)$ time on an $n \times n$ grid for any labeling.*

Proof. Let ϕ be a labeling of a 2-C E 3-S G. If G is merely a cycle C, then C can be planarly realized on an $n \times n$ grid with one edge bend. Begin with the

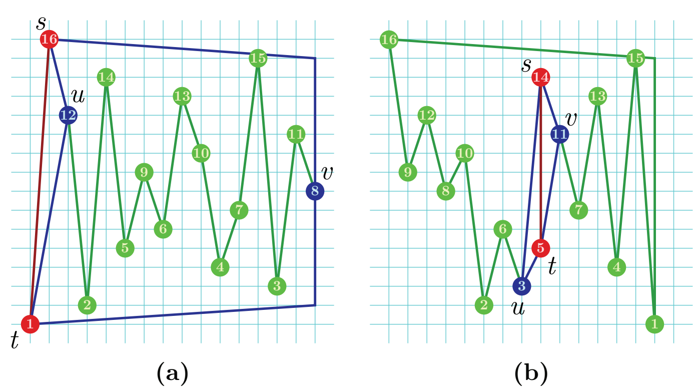


Fig. 7. Planar realizations of two 16-level 2-C E 3-Ss on 16×16 grids illustrating the two cases in which s and t are ϕ-extreme. If they are, the edge connecting s and t requires two bends as in (a). If not, only one edge bend is required as in (b).

ϕ-maximal vertex v_1 at the first position and proceed left to right placing each subsequent vertex in the cycle one to the right of the previous one until reaching the last vertex v_k that is also adjacent to v_1. The edge v_1–v_k requires only one bend directly above v_k routing the edge above all the other vertices.

By Definition 3, a 2-CE3-S is at worst a cycle with a kite edge between u and v with common neighbors $\{s, t\}$ connected by edge s–t such that $\phi(s) > \phi(t)$. If s and t are ϕ-extreme, then we can draw the cycle without t starting from s and ending with v as above and place t below s drawing the straight edges s–t and t–u. Then we draw t–v with a bend directly below v and route the edge below all the others; see Fig. 7(a). Otherwise, either s or t is not ϕ-extreme in which case the other ϕ-extreme one is used to draw the cycle so as to not end with u or v; see Fig. 7(b). Suppose that s is not ϕ-maximal, then t can be placed directly below s and the three additional edges can be added as straight edges. □

We can remove the bends on the edges by stretching the layout which yields the next corollary

Corollary 9 *An n-vertex 1-connected extended degree-3 spider with radius r can be straight-line planarly realized in $O(n)$ time on an $O(r!\,3^r) \times n$ grid for any labeling, whereas, an n-vertex 2-connected extended degree-3 spider can be straight-line planarly realized in $O(n)$ time on an $n^2 \times n$ grid for any labeling.*

Proof. The E3-S in Fig. 8 is a worst case for a degree-3 spider in terms of area. At each point in the algorithm of Lemma 7, there is only one choice when placing the next vertex in extending any chain forcing the three chains to form spirals. For each half spiral of a chain S when going from the lowest level of a vertex s at $\big(s_x, \phi(s)\big)$ to the higher level of a vertex t at $\big(t_x, \phi(t)\big)$ seen so far, the vertical distance between s and t increases by 3. Consider an adjacent chain W with its highest vertex v at $\big(v_x, \phi(v)\big)$ and lowest vertex u at $\big(u_x, \phi(v)\big)$ such that $\phi(v) = \phi(s) + 1$ and $\phi(u) = \phi(t) - 1$ extended so far. Then for edge v–u to clear t, v_x must satisfy

$$\begin{aligned} v_x &= (u_x - t_x) \cdot (\phi(v) - \phi(t)) - t_x \\ &= (u_x - t_x) \cdot (\phi(v) - \phi(t) - 1) \\ &= (u_x - t_x) \cdot (\phi(v) - \phi(u)). \end{aligned}$$

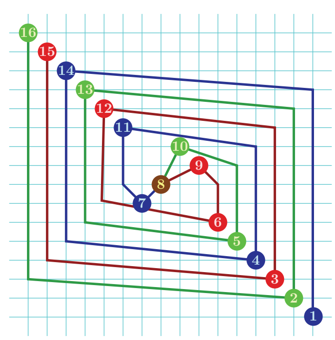


Fig. 8. A 16-level degree-3 spider on a 16×16 grid requires $O(n)$ bends.

Let w be the next vertex of W to be extended, such that $\phi(w) = \phi(u) - 3$. Then

$$\begin{aligned} w_x &= (s_x - v_x) \cdot (\phi(w) - \phi(s)) - t_x = (s_x - v_x) \cdot (\phi(w) - \phi(s) - 1) \\ &= (s_x - v_x) \cdot (\phi(w) - \phi(v)) \\ &= (s_x - [(u_x - t_x) \cdot (\phi(v) - \phi(u))]) \cdot (\phi(w) - \phi(v)). \end{aligned}$$

Hence, the dominating factor for a chain v_0–v_1–v_2–$\cdots$–v_r of length r, where $|\phi(v_{i+1}) - \phi(v_i)| = 3 + |\phi(v_i) - \phi(v_{i-1})|$ for $i \in [1..r-1]$, is $3 \times 6 \times 9 \times \cdots \times 3i \times \cdots \times 3r = r!\,3^r$. Let k be a constant greater than all of the initial factors for the three chains. In practice $k = 2$ suffices. Then, if the each v_i of the E3-S has coordinates $\big(x_{v_i}, \phi(v_i)\big)$, by moving v_i to $\big(-k|x_{v_i}|!\,3^{|x_{v_i}|}, \phi(x_i)\big)$ if $x_{v_i} < 0$ or $\big(kx_{v_i}!\,3^{x_{v_i}}, \phi(x_i)\big)$ if $x_{v_i} > 0$ is sufficiently far to the left or right so that it will clear all adjacent vertices so that straight edges can always be used.

Getting rid of the bends of a cycle in a 1-C E3-S requires stretching out the obscured vertices sufficiently far to the right. When drawing a cycle v_1–$\cdots$–v_n–v_1, a worst case in terms of space occurs when the vertex v_{n-1} placed at $\big(n-1, \phi(v_{n-1})\big)$ has level $\phi(v_{n-1}) = \phi(v_1) - 1$, and the vertex v_n placed at $\big(n, \phi(v_n)\big)$ has ϕ-minimum level $\phi(v_n) = 1$. In this case, in order for a straight edge v_1–v_n to clear v_{n-1}, v_n must be placed at $\big(n^2, \phi(v_n)\big)$ since the slope of the edge is $1/n$ taking $n^2 \times n$ space. □

Combining Lemmas 5, 6, 7, 8, and Corollary 9, we have our first theorem.

Theorem 10 *Any graph from $\mathcal{P}$ has a simultaneous geometric embedding with a strictly monotone path for any labeling.*

4 Forbidden Graphs

We give seven forbidden graphs $\mathcal{F} := \{T_8, T_9, G_5, G_6, G_\alpha, G_\delta, G_\kappa\}$ that do not always have a simultaneous geometric embedding with a strictly monotone path; see Fig. 9. For each we provide a labeling that forces self-crossings. As noted previously for a given labeling, a graph has a straight-line planar realization if and only if it also has a planar realization that allows edge bends provided the edges remain strictly monotone [4]. Hence, it suffices to only consider straight-line edges in this section.

Lemma 11 *There exist labelings that prevent each graph in $\mathcal{F}$ from having planar realizations on tracks.*

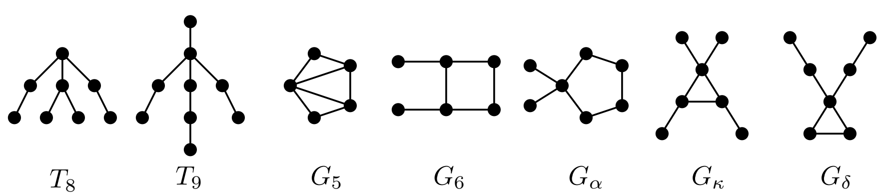


Fig. 9. The seven forbidden graphs of $\mathcal{F}$.

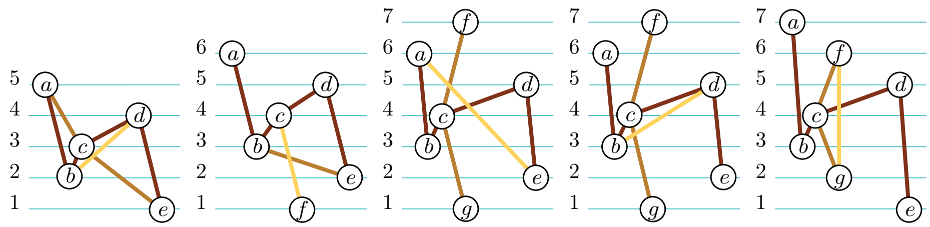


Fig. 10. Labelings that force self-crossings for G_5, G_6, G_α, G_κ, and G_δ.

Proof. The labelings of T_8 and T_9 were shown not to have planar realizations in [5]. We need to do the same for the labelings of the remaining five graphs in $\mathcal{F}$ given in Figure 10.

Let C denote the chain a–b–c–d–e, which is highlighted in each of the graphs in Figure 10. Observe that $\phi(a) > \phi(d) > \phi(c) > \phi(b) > \phi(e)$ in which C forms an backwards 'N'. If the rest of C intersects the track of c only on the left or right of c, then some part of the chain a–b–c must cross the chain c–d–e. Hence, we only need to consider embeddings in which c lies between the edge a–b and d–e, i.e., one of those edges intersect the track of c to the left, while the other intersects on the right. To avoid a self crossing of C, a–b must intersect the tracks of c and d on the same side of both vertices. The same goes for the d–e intersecting the tracks of b and c on the same side. So we can assume w.l.o.g. that a–b intersects the tracks of c and d to the their left while d–e intersects the tracks of b and c to the their right as is the case in all the figures.

For G_5, c and d being on the same side of a–b means that the edge b–d must also lie between the two edges. The only question is whether b–d intersects the track of c to the left or right. If it is to the left, then b–d must cross a–c, otherwise, it must cross c–e as in Fig. 10(a).

For G_6, from the assumptions, the edge c–f either crosses

(i) a–b if it intersects the track of b to the left since c is right of a–b,

(ii) d–e if it intersects the track of e to the right since c is left of d–e,

(iii) b–e otherwise since it must intersect the track of b to the right and e to the left as in Fig. 10(b).

In G_α, G_δ and G_κ for c–f and c–g to avoid crossing C, c–f must intersect the track of d to the left while c–g must intersect the track of b to the right. Since $\phi(f) > \phi(a) > \phi(e) > \phi(g)$ in G_α and G_κ, c–f must intersect the track of a to the right while c–g must intersect the track of e to the left. However, in G_δ $\phi(a) > \phi(f) > \phi(g) > \phi(e)$ so that a–b must intersect the track of f to the right while d–e must intersect the track of g to the left.

This means in G_α for a–e to avoid crossing C, as in Fig. 10(c), it must either intersect the track of d to the right in which case it must cross c–f or b to the left in which case it must cross c–g.

This also means in G_κ if b–d intersects the track of c to the right as in Fig. 10(d), it will cross c–g. Otherwise, b–d will cross c–f.

Finally, in G_δ if f–g intersects the track of c to the right as in Fig. 10(e), it will cross c–d–e. Otherwise, f–g will cross a–b–c. □

Corollary 12 *A graph containing a subgraph homeomorphic to a graph in $\mathcal{F}$ does not have a simultaneous geometric embedding with a strictly monotone path for all labelings.*

Proof. We provide a labeling ϕ of a graph G containing a subgraph homeomorphic to a graph $\tilde{G} \in \mathcal{F}$. Let h be the homeomorphism that maps an edge in $\tilde{G}$ to a path in G and a vertex in $\tilde{G}$ to the endpoint of such a path in G. Label the vertices of $\tilde{G}$ using the appropriate labeling ϕ' from Lemma 11 that forces a self-crossing in $\tilde{G}$. We maintain the same relative ordering of the labels in G as in $\tilde{G}$. In particular, we want $\phi\big(h(u)\big) < \phi\big(h(v)\big)$ if and only if $\phi'(u) < \phi'(v)$ for each edge (u, v) in $\tilde{G}$. For each path $h\big((u, v)\big) = p_{(u,v)} = v_1\text{–}v_2\text{–}\cdots\text{–}v_k$ in G that corresponds to an edge (u, v) in $\tilde{G}$, we want $\phi(v_1) < \phi(v_2) < \cdots < \phi(v_k)$ if $\phi'(u) < \phi'(v)$. We can assign the other vertices of G not in the image of h arbitrary labels. Then every edge (u, v) in $\tilde{G}$ corresponds to a strictly monotone path $p_{(u,v)}$ in G preserving the nonplanarity of the realization of $\tilde{G}$. □

5 Completing the Characterization

The next lemma shows that the seven forbidden graphs of $\mathcal{F}$ are minimal; the removal of any edge from any of the seven yields a graph from $\mathcal{P}$.

Lemma 13 *Each forbidden graph is minimal, in that the removal of any edge yields one or more GCs, R-2Ss, or E3-Ss.*

Proof. Showing that the removal of any edge from T_8 or T_9 yielded a caterpillar, radius-2 star, or degree-3 spider, all members of $\mathcal{P}$, was done in [5]. For G_5 in which a–b–d–e–c–a, a–b–c–a, b–c–d–b, c–d–e–c all form cycles shown in Fig. 10(a), the removal of edges b–c or c–d forms a 2-CE3-S, while removing of any other edge forms a GC. For G_6 in which b–e–d–c forms a 4-cycle shown in Fig. 10(b), the removal of any edge leaves a GC. For G_α shown in Fig. 10(c), the removal of c–f or c–g leaves a E3-S. Removing any other edge yields a GC. For G_κ in which b–c–d–b forms a 3-cycle shown in Fig. 10(d), the removal of c–f or c–g leaves a 1-CE3-S, while removing any other edge leaves a GC. For G_δ in which c–f–g–c forms a 3-cycle shown in Fig. 10(e), the removal of c–b or c–d leaves a GC and a lone edge. Removing a–b, d–e, or f–g leaves a GC, and removing c–f or c–g leaves a degree-3 spider. □

Finally, the next theorem completes our characterization.

Theorem 14 *Every connected graph either contains a subgraph homeomorphic to one of the seven forbidden graphs of $\mathcal{F}$, or it is a generalized caterpillar, radius-2 star, or a extended degree-3 spider, which form the collection of graphs $\mathcal{P}$ that have simultaneous geometric embeddings with strictly monotone paths for any labelings, the set of ULP graphs.*

Proof. Let $G(V, E)$ be a connected graph and let $\mathcal{T}(G)$ denote the set of all spanning trees of G, and let $\mathcal{C}$, $\mathcal{R}$, $\mathcal{S}$ denote the sets of all caterpillars, radius-2 stars, and degree-3 spiders, resp. By Theorem 7 of [5], any $T \in \mathcal{T}(G)$ must either

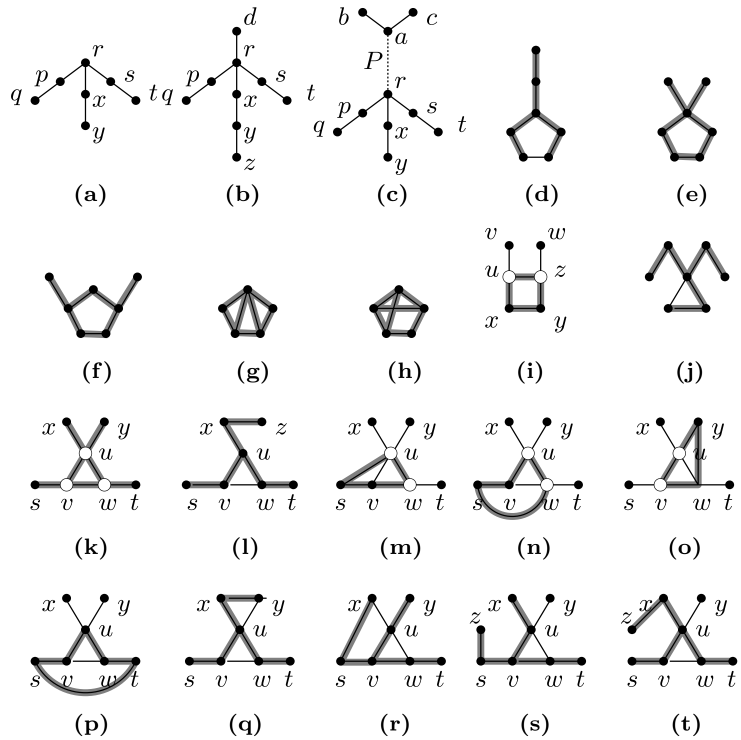


Fig. 11. Various cases of Theorem 14 in which shaded edges are subgraphs and white vertices are cut vertices in the proof.

have a subdivision of T_8 or T_9 or $T \in \mathcal{C} \cup \mathcal{R} \cup \mathcal{S}$. For G not to have a subdivision of T_8 or T_9, none of the spanning trees in $\mathcal{T}(G)$ can.

We cannot have a case in which $T_1, T_2 \in \mathcal{T}$ such that T_1 and T_2 are in separate categories with neither in the common intersection. For instance, suppose $T_1 \in \mathcal{R} - \mathcal{S}$, and $T_2 \in \mathcal{S} - \mathcal{R}$. Clearly both T_1 and T_2 contain a minimal lobster L (a $K_{1,3}$ in which each edge has been subdivided once). Label the three chains of L as r–p–q, r–s–t, r–x–y with root vertex r as in Fig. 11(a). Since $T_1 \in \mathcal{R} - \mathcal{S}$, r must have degree at least 4 with extra edge r–d. Since $T_2 \in \mathcal{S} - \mathcal{R}$, it must have radius of 3 or greater in which one of the chains has an incident edge, say that it is y–z. Then L, r–d, and y–z form a copy of T_9 in G as in Fig. 11(b). However, if $T_1 \in (\mathcal{R} \cup \mathcal{S}) - \mathcal{C}$, then T_1 must contain L (using the same labeling as above) preventing $T_1 \in \mathcal{C}$. On the other hand, if $T_2 \in \mathcal{C} - (\mathcal{R} \cup \mathcal{S})$, then T_2 must contain two vertices of degree > 2. One of these is r and suppose that the other is a. This gives a path $r \rightsquigarrow a$, call it P, having a pair of edges incident to v, namely a–b and a–c that are not on P. Then L and P along with edges a–b and a–c form a copy of T_8 as in Fig. 11(c).

Hence, $G \in \mathcal{G}_{\mathcal{X}} = \{H : H \text{ is a graph such that } T \in \mathcal{X} \text{ for all } T \in \mathcal{T}(H)\}$, where $\mathcal{X}$ is either $\mathcal{C}$, $\mathcal{R}$, or $\mathcal{S}$. We argue that G must be a R-2S, GC, E3-S if one adds the restrictions of G not containing G_5, G_6, G_α, G_δ, or G_κ. If $G \in \mathcal{G}_{\mathcal{R}}$, then G can has at exactly one vertex of degree > 2, given G has a R-2S as a subgraph and if there were more than one vertex v of degree > 2, then there would exist a $T \in \mathcal{T}$ containing at least two such vertices. This implies that G can have at most one cycle with v as a vertex, which must be K_3 since otherwise there would a $T \in \mathcal{T}$ with radius > 3. Since G cannot contain G_δ, if G has K_3, v has at most one incident path of length 2 and any number of incident leaf edges, which forms a GC. Hence, G is either a R-2S or such a GC.

If $G \in \mathcal{G}_\mathcal{S}$, G could be a E3-S, but must have maximum degree 3. We need to show that this is equivalent to G not containing any of the forbidden graphs. Clearly, G cannot contain G_5, G_α, G_δ or G_κ since they have spanning trees with vertices of degree 4. Neither can G contain G_6 since it has a spanning tree with two vertices of degree 3. This shows that the condition of having maximum degree 3 to be sufficient. To show that it is necessary, suppose that G has a vertex v of degree 4. If $G \notin \mathcal{G}_\mathcal{C}$, then there exists a $T \in \mathcal{T}(G)$ containing lobster $L \notin \mathcal{C}$ with chains r–p–q, r–s–t, r–x–y and root vertex r as in Fig. 11(a). If v is r, this implies the existence of an incident edge r–d. We can assume that $G \notin \mathcal{R}$, since $G \in \mathcal{R}$ has already been considered. Hence, G must have radius greater than 2, implying w.l.o.g. the existence of the edge y–z creating a copy of T_9 as in Fig. 11(b), or another vertex a of degree 3 with incident edges a–b and a–c not on the path $r \rightsquigarrow a$. Then this path and these edges along with L create a copy of T_8 as in Fig. 11(c).

If $G \in \mathcal{G}_\mathcal{C}$, then G could be a GC. However, a GC has three extra conditions: (1) all cycles of length at most 4, (2) no three pairwise adjacent cut vertices, and (3) no adjacent pair of cut vertices in the same 4-cycle. The forbidden graphs G_5, G_6 and G_α combine to impose condition (1): Aside from the two special cases of a C_5 with one incident edge or a C_5 with a chord, which are both a E3-S, no other graphs with a cycle of length greater than 4 contain either L or a subdivision of G_5, G_6, or G_α. We see this in that G has a copy of L as soon as there is a path of length two incident the cycle as in Fig. 11(d). This means that only chords or incident edges can be added to C_5. If more than one incident edge is added to the same vertex, then G has a copy of G_α as in Fig. 11(e). If more than two vertices have incident edges added, then a subdivision of G_6 is created; see Fig. 11(f). Finally, if more than one chord is added, then a copy of G_5 is created regardless if the chords are incident or not as in Figs. 11(g–h). The forbidden graph G_κ imposes condition (2) limiting the type of K_3's found in a GC. Aside from the special case of a lone K_3 of three pairwise cut vertices u, v, w with one incident edge each, namely u–x, v–s, and w–t, that is a E3-S, any graph having a K_3 on three pairwise adjacent cut vertices u, v, w either contains a copy of L or G_κ. As soon as we add another incident edge to u, v, or w, say that it is u–y, we create a copy of G_κ as in Fig. 11(k). Otherwise, adding an incident edge to x, s, or t, say that it is x–z, then one has a copy of L as in Fig. 11(l). Adding an edge between any of the six vertices, stops u, v, and w from being three pairwise cut vertices so that condition (2) would no longer apply. Finally, the forbidden graph G_6 directly imposes condition (3) in that as soon as there is a 4-cycle u–x–y–z–u with adjacent cut vertices u and x, implying incident edges u–v and x–w, there is a copy of G_6 in G as in Fig. 11(i).

This shows that the forbidden graphs collectively impose the three additional conditions on a GC, showing that they are necessary. However, what is left to show is that a GC cannot contain any of the forbidden subgraphs, i.e., to show that the three conditions are also sufficient. Condition (1) immediately prohibits the existence of either G_5 or G_α in G. Also G cannot contain G_δ since it contains L as proper subgraph excluding G from $\mathcal{G}_\mathcal{C}$ as in Fig. 11(j). If G contains G_6 with the 4-cycle u–x–y–z–u and incident edges u–v and x–w, in order to prevent the creation of a cycle of length greater than 4 violating condition (1), no edges or paths can be added between these six vertices, which means that u and x would be cut vertices violating condition (3).

Suppose that G contains G_κ with the K_3 u–v–w–u with incident edges u–x, u–y, v–s, and w–t. We consider all the non-isomorphic ways in which an edge can be added to G_κ There are six non-isomorphic edges that can be added to G_κ without introducing another vertex, namely, s–t, s–u, s–w, x–s, y–w, and x–y. Adding s–u, s–w, or y–w has either v and w or u and w as adjacent cut vertices of the cycles, s–u–w–v–s, s–w–u–v–s, or y–w–v–u–y, respectively, violating condition (3) as in Figs. 11(m–o). Adding s–t, creates a 5-cycle, s–t–w–u–v–s, violating condition (1) as in Fig. 11(p). Adding x–y or x–s creates a copy of L, namely u–x–y, u–v–s, and u–w–t with u as the root vertex as in Fig. 11(q) or v–s–x, v–u–y, and v–w–t with v as the root vertex as in Fig. 11(r), which prevents $G \in \mathcal{G}_\mathcal{C}$. Hence, we only have to consider adding some other vertex z to the non-isomorphic vertices s, u, v or x. of G_κ. Adding incident edges s–z or s–x also creates a copy of L, namely v–s–z, v–u–x, and v–w–t with v as the root vertex as in Fig. 11(s) or u–x–z, u–v–s, and v–w–t with u as the root vertex as in Fig. 11(t). Adding incident u–z or v–z allows for u, v and w to remain as three pairwise adjacent cut vertices violating condition (2). This completes the proof since we have shown that the definition of a GC is equivalent to $G \in \mathcal{G}_\mathcal{C}$ where G does not contain a forbidden graph. □

6 Linear Time Recognition

We use our characterization to obtain a linear time recognition algorithm.

Corollary 15 *The class of graphs $\mathcal{P}$ that have a simultaneous geometric embedding with a strictly monotone path can be recognized in linear time. Given an graph G with a labeling, one can decide in $O(n)$ time if it is always possible to planarly realize G.*

Proof. Let $G(V, E)$ be a graph with n vertices and m edges. As detailed in [5], an R-2S is recognized by verifying all degree-2 vertices are adjacent to the root vertex and a leaf, which forms the basis of a linear time recognition algorithm. A E3-S has maximum degree 3 where every spanning tree is a degree-3 spider. We can find a spanning tree T of G in $O(m)$ time. However, if $m > 3n - 6$, then G cannot be planar, hence cannot be level planar for any level assignment, and thus is not in $\mathcal{P}$. As such, we only consider graphs with no more than $3n - 6$ edges. Hence, it only takes $O(n)$ time to find T, or we reject G for having too many edges. From the possibilities given by Definition 3, a E3-S has at most four vertices of degree 3 giving at most three more edges than T. We determine if G is 2-connected by seeing if removing any of the (at most four) degree-3 vertices disconnects the graph, which takes $O(n)$ time. If not, then it must match the form of a 1-C E 3-S of 2(i) of Definition 3, in which there can be at most two extra edges that is easily verified. Otherwise, it must match the form of a 2-C E 3-S of 2 (ii) of Definition 3, in which case the extra edges can be verified to match one of the three sub-possibilities shown in Fig. 4(ii). If G is a GC, then T must a caterpillar, which can be determined in linear time by removing all degree-1 vertices. Using T, we apply the procedure outlined in the drawing algorithm for a GC in Lemma 6 to decompose the GC via T into its four sets of edges, K_4, K_3^*, C_4^+ and kite edges w.r.t T spending constant time per edge. If this cannot be done, or some edge, such as K_4, exists in the middle of the GC, then we reject the G as not being in $\mathcal{P}$. Thus, we can determine in $O(n)$ time if G is in $\mathcal{P}$ by being a R-2 S, GC, or E 3-S, or reject G accordingly. □

7 Previous and Future Work

Level planar graphs are historically studied in the context of directed graphs, which restricts the types of levelings that can be assigned. Additionally, they are generally considered in the context of a particular leveling such as ones given by hierarchical relationships with an emphasis on minimizing the number of levels required to maintain planarity. In contrast, our application of level planarity has been in terms of the underlying undirected graph with one vertex per level with no consideration given to minimizing levels.

Many of the problems regarding level planarity have been addressed, including the ability to recognize a level planar graph and produce an embedding in linear time [8, 9]. However, all of these results are for a particular leveling and do not generalize to the context of considering the level planarity of all the level graphs induced by all possible $n!$ labelings of a given undirected graph. Running either of these linear time algorithms for each possible level graph leads to an exponential running time. Using our approach we achieve this in linear time

We gave a characterization of ULP graphs akin to Kuratowksi's characterizations of planar graphs [10]; we provided a forbidden set of graphs $\mathcal{F}$ in Fig. 11 that play the same role with respect to ULP graphs that K_5 and $K_{3,3}$ play with respect to planar graphs. Just as Kuratowksi's theorem states that a graph is planar if and only if it does not contain a subgraph that is a subdivision of K_5 or $K_{3,3}$, we show a graph is ULP if and only if it does not contain a subgraph homeomorphic to a forbidden graph of $\mathcal{F}$.

The analogue of Kuratowksi's theorem for level planar graphs are minimum level non-planar patterns [7]. These are based on the characterization of hierarchies by Di Battista and Nardelli [3]. Unlike our characterization, these patterns are not solely based upon the underlying graph, but also upon the given leveling. The same graph with two different levelings that is level non-planar for each may very well match two distinct patterns since the reasons that a crossing is forced in each can be entirely different.

Estrella *et al*. [5] presented linear time recognition algorithms for the class of ULP trees. We provided a similar linear time recognition algorithm for ULP graphs. What is missing from the recognition algorithm and left for future work is a certificate of unlabeled level non-planarity, i.e., the vertices that correspond to the offending forbidden graph it exists.

References

1. U. Brandes, C. Erten, J. Fowler, F. Frati, M. Geyer, C. Gutwenger, S. Hong, M. Kaufmann, S. Kobourov, G. Liotta, P. Mutzel, and A. Symvonis. Colored simultaneous geometric embeddings. In *13th Computing and Combinatorics Conference (COCOON)*, 2007. To appear in 2007.
2. P. Brass, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous graph embedding. *Computational Geometry: Theory and Applications*, 36(2):117–130, 2007.
3. G. Di Battista and E. Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(6):1035–1046, 1988.
4. P. Eades, Q. Feng, X. Lin, and H. Nagamochi. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. *Algorithmica*, 44(1):1–32, 2006.
5. A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Characterization of unlabeled level planar trees. In Kaufman and Wagner, editors, *14th Symposium on Graph Drawing (GD)*, volume 4372 of *Lecture Notes in Computer Science*, pages 367–369, 2006.
6. M. Geyer, M. Kaufmann, and I. Vrto. Two trees which are self-intersecting when drawn simultaneously. In *13th Symposium on Graph Drawing*, pages 201–210, 2005.
7. P. Healy, A. Kuusik, and S. Leipert. A characterization of level planar graphs. *Discrete Math.*, 280(1-3):51–63, 2004.
8. M. Jünger and S. Leipert. Level planar embedding in linear time. *J. Graph Algorithms Appl.*, 6(1):67–113, 2002.
9. M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In *6th Symposium on Graph Drawing (GD)*, pages 224–237, 1998.
10. C. Kuratowski. Sur les problèmes des courbes gauches en Topologie. *Fundamenta Mathematicae*, 15:271–283, 1930.