Matched drawings of planar graphs

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 13, no. 3, pp. 423–445 (2009)

Matched Drawings of Planar Graphs

Emilio Di Giacomo[1] *Walter Didimo*[1] *Marc van Kreveld*[2]
Giuseppe Liotta[1] *Bettina Speckmann*[3]

[1]Dip. di Ingegneria Elettronica e dell'Informazione,
Università degli Studi di Perugia
[2]Department of Information and Computing Sciences, Utrecht University
[3]Department of Mathematics and Computer Science, TU Eindhoven

Abstract

A natural way to draw two planar graphs whose vertex sets are matched is to assign each matched pair a unique y-coordinate. In this paper we introduce the concept of such matched drawings, which is a relaxation of simultaneous geometric embeddings with mapping. We study which classes of graphs allow matched drawings and show that (i) two 3-connected planar graphs or a 3-connected planar graph and a tree may not be matched drawable, while (ii) two trees or a planar graph and a sufficiently restricted planar graph—such as an unlabeled level planar (ULP) graph or a graph of the family of "carousel graphs"—are always matched drawable.

Submitted:	Reviewed:	Revised:	Accepted:
December 2007	May 2008	August 2008	November 2008
	Final:	Published:	
	November 2008	November 2009	

Article type:	Communicated by:
Regular paper	S.-H. Hong and T. Nishizeki

Research partially supported by the MIUR Project "MAINSTREAM: Algorithms for massive information structures and data streams". Research of B.S. partially supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 639.022.707.

E-mail addresses: digiacomo@diei.unipg.it (Emilio Di Giacomo) didimo@diei.unipg.it (Walter Didimo) marc@cs.uu.nl (Marc van Kreveld) liotta@diei.unipg.it (Giuseppe Liotta) speckman@win.tue.nl (Bettina Speckmann)

1 Introduction

The visual comparison of two graphs whose vertex sets are associated in some way requires drawings of these graphs that highlight their association in a clear manner. Drawings of this type are of use for various areas of computer science, including bio-informatics, web data mining, network analysis, and software engineering. Of course each drawing individually should be as clear as possible, using, for example, few bends and crossings. But, most importantly, the positions of associated vertices in the two drawings should be "close". This makes it possible for the user to easily identify structurally identical and structurally different portions of the two graphs, or to maintain her "mental map" [19]. Structural changes between two graphs and their visualizations arise, for example, when collapsing or expanding clusters in clustered drawings [8], during the navigation of very large graphs with a topological window [7, 18], in the analysis of evolving networks [1, 16], and in dynamic graph drawing [3, 20, 21]. In all these scenarios the basic approach is to maintain the relative positions of associated vertices as much as possible to smoothly transform one graph into another. In this way changes can be captured more easily by the human eye.

Two positions are definitely "close" if they are identical. Hence a substantial research effort has recently been devoted to the problem of computing straight-line drawings of two graphs on the same set of points. More specifically, assume we are given two planar graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2|$, together with a one-to-one mapping between their vertices. A *simultaneous geometric embedding with mapping* (introduced by Brass et al. in [4]) of G_1 and G_2 is a pair of straight-line planar drawings Γ_1 and Γ_2 of G_1 and G_2, respectively, such that for any pair of matched vertices $u \in V_1$ and $v \in V_2$ the position of u in Γ_1 is the same as the position of v in Γ_2. Unfortunately, only pairs of graphs belonging to restricted subclasses of planar graphs admit a simultaneous geometric embedding with mapping. Brass et al. [4] showed how to simultaneously embed pairs of paths, pairs of cycles, and pairs of caterpillars, but they also proved that a path and a graph or two outerplanar graphs may not admit this type of drawing. Geyer, Kaufmann, and Vrt'o [17] recently proved that even a pair of trees may not have a simultaneous geometric embedding with mapping. These negative results motivated the study of relaxations of simultaneous geometric embeddings. One possibility is to introduce bends along the edges [5, 9, 10, 14], another, to allow that the same vertex occupies different locations in the two drawings [2, 4, 15], introducing ambiguity in the mapping.

In this paper we consider a different interpretation of two positions being "close". Instead of requiring that matched vertices occupy the same location, we assign each matched pair a unique y-coordinate. This enables the user to unambiguously identify pairs of matched vertices but, at the same time, leaves us more freedom to draw both graphs clearly. Specifically, let again $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two planar graphs with $|V_1| = |V_2|$. G_1 and G_2 are *matched* if there is a one-to-one mapping between V_1 and V_2. If a vertex $u \in V_1$ is matched with a vertex $v \in V_2$ then we say that u is the *partner* of v and that v is the partner of u. A *matched drawing* of G_1 and G_2 is a pair of straight-line

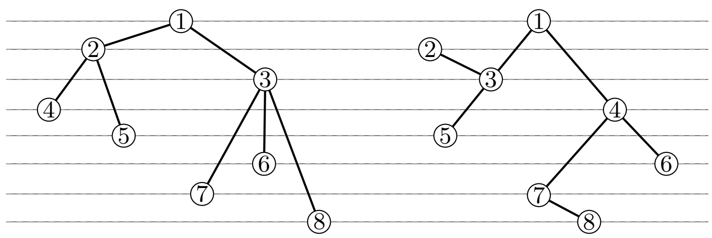

Figure 1: A matched drawing of two trees.

planar drawings Γ_1 and Γ_2 of G_1 and G_2, respectively, such that for any pair of matched vertices $u \in V_1$ and $v \in V_2$ the y-coordinate of u in Γ_1 is the same as the y-coordinate of v in Γ_2, and this y-coordinate is unique. If two matched graphs have a matched drawing, then we say that they are *matched drawable*. Matched drawings can be viewed as a relaxation of simultaneous geometric embedding with mapping. An example of a matched drawing of two trees is shown in Figure 1.

We remark that a problem similar to the matched drawability problem posed in this paper has been previously studied by Fernau et al. for the comparison of a pair of phylogenetic trees [12]. In that paper the matching is restricted to the set of leaves of the two trees, and the objective is to compute planar drawings for the two trees that minimize the number of crossings between the matching edges.

Results and Organization. We start by presenting pairs of graphs that are not matched drawable. In particular, in Section 2.1 we describe two isomorphic 3-connected planar graphs that both have 12 vertices and that are not matched drawable. We also present a 3-connected planar graph and a tree that both have 620 vertices and that are not matched drawable. This construction can be found in Section 2.2.

We continue by describing drawing algorithms for classes of graphs that are always matched drawable. In particular, in Section 3.1 we show that a planar graph and an unlabeled level planar (ULP) graph that are matched are always matched drawable. In Section 3.2 we extend these results to a planar graph and a graph of the family of "carousel graphs". Finally, in Section 3.3 we prove that two matched trees are always matched drawable.

2 Graphs that are not Matched Drawable

2.1 Two 3-connected Graphs

We start by stating a simple property of planar straight-line drawings.

Property 1 *Let G be an embedded planar graph and let Γ be a straight-line planar drawing of G. Let u be the vertex of G with the highest y-coordinate in*

Γ and let v be the vertex of G with the lowest y-coordinate in Γ. Vertices u and v belong to the external face of G.

Now assume that G_1 and G_2 are two matched graphs with the following properties: (i) G_1 contains two vertex-disjoint simple cycles $C_1 = \{u_1, \ldots, u_n\}$ and $C'_1 = \{u'_1, \ldots, u'_m\}$, (ii) G_2 contains two vertex-disjoint simple cycles $C_2 = \{v_1, \ldots, v_n\}$ and $C'_2 = \{v'_1, \ldots, v'_m\}$, and (iii) u_i is the partner of v_i $(1 \leq i \leq n)$ and u'_j is the partner of v'_j $(1 \leq j \leq m)$. If Ψ_1 is a planar embedding of G_1 such that C'_1 is inside C_1 and if Ψ_2 is a planar embedding of G_2 such that C_2 is inside C'_2, then we call Ψ_1 and Ψ_2 *interlaced embeddings* and C_1, C'_1, C_2, and C'_2 *interlaced cycles*.

Lemma 1 *Let G_1 and G_2 be two matched graphs with interlaced embeddings Ψ_1 and Ψ_2. There is no matched drawing Γ_1 and Γ_2 of G_1 and G_2 such that Γ_1 preserves Ψ_1 and Γ_2 preserves Ψ_2.*

Proof: Denote by C_1, C'_1, C_2, and C'_2 the interlaced cycles of Ψ_1 and Ψ_2. Suppose by contradiction that Γ_1 and Γ_2 exist. Denote by $\overline{\Gamma_1}$ the subdrawing of Γ_1 restricted to the subgraph $C_1 \cup C'_1$ and by $\overline{\Gamma_2}$ the subdrawing of Γ_2 restricted to the subgraph $C_2 \cup C'_2$.

Since in Ψ_1 cycle C'_1 is inside cycle C_1, by Property 1 the top-most and the bottom-most vertices of $\overline{\Gamma_1}$ belong to C_1; denote these two vertices by u_t and u_b. Since $\overline{\Gamma_1}$ is planar and since the drawing of C'_1 is completely inside the drawing of C_1, every vertex u'_j of C'_1 has a y-coordinate that is greater than the y-coordinate of u_b and smaller than the y-coordinate of u_t. Since Γ_1 and Γ_2 are matched drawings, every vertex v'_j of C'_2 in $\overline{\Gamma_2}$ has a y-coordinate that is greater than the y-coordinate of v_b (i.e., the partner of u_b) and smaller than the y-coordinate of v_t (i.e., the partner of u_t). However, since in Ψ_2 cycle C_2 is inside cycle C'_2, by Property 1 the top-most and the bottom-most vertices of $\overline{\Gamma_2}$ belong to C'_2, a contradiction. □

Theorem 1 *There exist two 3-connected planar graphs that are not matched drawable.*

Proof: Consider the two 3-connected planar graphs G_1 and G_2 in Figure 2. The partner of a vertex of G_1 is any vertex in G_2 that has the same label. To prove that G_1 and G_2 are not matched drawable, we show that all planar embeddings of G_1 and G_2 are interlaced embeddings.

Since G_1 and G_2 are 3-connected graphs, all their planar embeddings differ only in the choice of the external face. In G_1 and G_2 we can have five possible types of external face, depending on the labels of the vertices of such a face. Namely, an external face of G_1 can have vertices with labels in one of these sets: $\{a\}$, $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, $\{d\}$, while an external face of G_2 can have vertices with labels in one of these sets: $\{c\}$, $\{c, d\}$, $\{d, a\}$, $\{a, b\}$, $\{b\}$. For any label $\ell \in \{a, b, c, d\}$, let $C_{1,\ell}$ and $C_{2,\ell}$ denote the three-cycles formed by the vertices with label ℓ in G_1 and in G_2, respectively. For any pair of external faces in G_1 and G_2 there are two distinct labels $\ell, \ell' \in \{a, b, c, d\}$ such that $C_{1,\ell'}$ is inside

$C_{1,\ell}$ in G_1 and $C_{2,\ell}$ is inside $C_{2,\ell'}$ in G_2. Table 1(a) shows the inclusion relations between the three-cycles of G_1 for each type of external face, where we use the notation $\ell \succ \ell'$ to denote that cycle $C_{1,\ell'}$ is inside $C_{1,\ell}$. Table 1(b) shows the inclusions between the three-cycles of G_2.

For each pair of external faces of G_1 and G_2, Table 2 shows two labels ℓ, ℓ' such that $C_{1,\ell}, C_{1,\ell'}, C_{2,\ell}, C_{2,\ell'}$ are interlaced cycles. More precisely, in Table 2 the rows are the labels of the external face of G_1, the columns are the labels of the external face of G_2, and in each cell two labels ℓ, ℓ' are shown such that $\ell \succ \ell'$ in G_1 and $\ell' \succ \ell$ in G_2. For example, if the external face of G_1 is the three-cycle $C_{1,a}$ and if the external face of G_2 is the three-cycle $C_{2,b}$, we have that in G_1 cycle $C_{1,d}$ is inside $C_{1,c}$, while in G_2 cycle $C_{2,c}$ is inside $C_{2,d}$. Hence, any pair of planar embeddings of G_1 and G_2 is a pair of interlaced embeddings. □

2.2 A 3-connected Graph and a Tree

The two graphs described in Theorem 1 are both 3-connected. Hence the question arises if two planar graphs, at least one of which is not 3-connected, are always matched drawable. This is unfortunately not the case: in the following we present a planar graph and a tree that are not matched drawable.

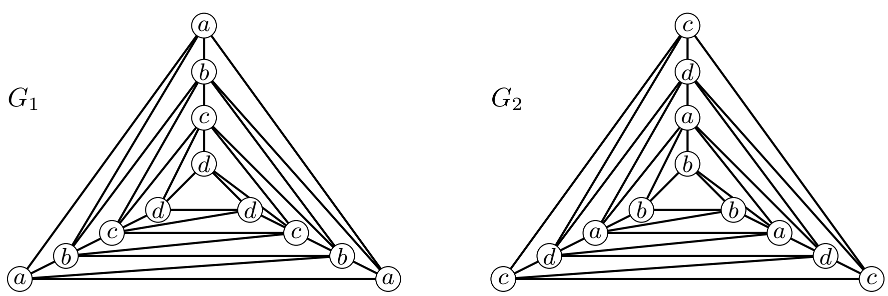

Figure 2: Two 3-connected planar graphs that are not matched drawable. The partner of a vertex of G_1 is any vertex in G_2 that has the same label.

Ext. face labels	*Incl. of cycles*
$\{a\}$	$a \succ b \succ c \succ d$
$\{a, b\}$	$b \succ c \succ d$
$\{b, c\}$	$b \succ a$; $c \succ d$
$\{c, d\}$	$c \succ b \succ a$
$\{d\}$	$d \succ c \succ b \succ a$

(a)

Ext. face labels	*Incl. of cycles*
$\{c\}$	$c \succ d \succ a \succ b$
$\{c, d\}$	$d \succ a \succ b$
$\{d, a\}$	$d \succ c$; $a \succ b$
$\{a, b\}$	$a \succ d \succ c$
$\{b\}$	$b \succ a \succ d \succ c$

(b)

Table 1: Inclusions between the three-cycles of G_1 (table (a)) and of G_2 (table (b)).

	$\{c\}$	$\{c,d\}$	$\{d,a\}$	$\{a,b\}$	$\{b\}$
$\{a\}$	a,c	a,c	c,d	c,d	c,d
$\{a,b\}$	b,d	b,d	c,d	c,d	c,d
$\{b,c\}$	b,a	b,a	b,a	c,d	c,d
$\{c,d\}$	b,a	b,a	b,a	c,a	c,a
$\{d\}$	b,a	b,a	b,a	d,a	d,a

Table 2: Interlaced cycles for each pair of external faces. The rows are the labels in the external face of G_1; the columns are the labels in the external face of G_2. In each cell two labels ℓ, ℓ' are shown such that $\ell \succ \ell'$ in G_1 and $\ell' \succ \ell$ in G_2.

Given a vertex v of a graph G and a drawing Γ of G, we denote by $x(v)$ and $y(v)$ the x- and y-coordinate of v in Γ. Let $T^* = (V^*, E^*)$ be the tree depicted in Figure 3. Estrella-Balderrama et al. [11] proved the following lemma:

Lemma 2 (Estrella-Balderrama et al. [11]) *Let T^* be the tree depicted in Figure 3. A straight-line planar drawing Γ of T^* such that $y(v_0) < y(v_7) < y(v_3) < y(v_2) < y(v_4) < y(v_1) < y(v_5) < y(v_6)$ in Γ does not exist.*

Let T^* be rooted at vertex v_0, and for each vertex v_i, denote by $d(v_i)$ the graph-theoretic distance of v_i from the root ($i = 0, 1, \dots, 7$). We construct a tree T by using T^* as a model. See Figure 4 for an illustration. T has $3^{d(v_i)}$ copies of each vertex v_i ($i = 0, 1, \dots, 7$). The $3^{d(v_i)}$ copies of v_i are denoted as $v_{i,0}, v_{i,1}, \dots, v_{i,3^{d(v_i)}-1}$. Vertex $v_{h,k}$ is a child of vertex $v_{i,j}$ in T if and only if v_h is a child of v_i in T^* and $j = \lfloor k/3 \rfloor$ ($0 \le i, h \le 7$), ($0 \le j \le 3^{d(v_i)} - 1$), ($0 \le k \le 3^{d(v_h)} - 1$). So T has one copy of v_0 whose children are the three copies $v_{1,0}$, $v_{1,1}$, and $v_{1,2}$ of v_1. The children of each copy of v_1 are three of the nine copies of v_2, and so on. Three vertices of T with the same parent are called a *triplet* of T. The total number of vertices of T is 310.

The tree T is matched with a *nested-triangles graph*, which is defined as follows. A single vertex v is a nested-triangles graph denoted by G_0. A trian-

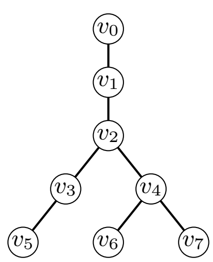

Figure 3: A tree that does not have a straight-line planar drawing with $y(v_0) < y(v_7) < y(v_3) < y(v_2) < y(v_4) < y(v_1) < y(v_5) < y(v_6)$ [11].

vertex	*copies*	*triplets*	*levels*
v_0	1	0	0
v_7	81	27	1...27
v_3	27	9	28...36
v_2	9	3	37...39
v_4	27	9	40...48
v_1	3	1	49
v_5	81	27	50...76
v_6	81	27	77...103

Table 3: Matching between the vertices of T and the vertices of G_{103}.

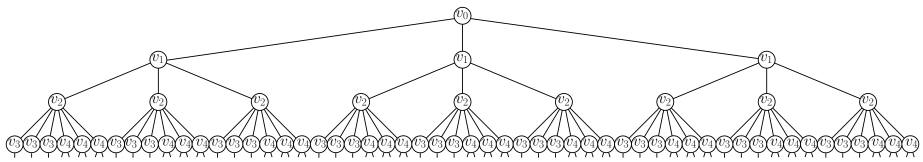

Figure 4: Tree T of the construction. Nodes of the last level (i.e. copies of nodes v_5, v_6, and v_7) are not shown.

gulated planar embedded graph G_k ($k > 0$) is a nested-triangles graph if the external face of G_k has exactly three vertices and the graph G_{k-1}, obtained by removing the vertices on the external face, is still a nested-triangles graph. A levelling of the vertices is naturally defined for the vertices of G_k: level i of G_k contains the vertices that are on the external face of G_i ($i = 0, 1, \dots, k$). Note that G_k has $3k + 1$ vertices and $k + 1$ levels. Thus, G_{103} has 310 vertices and 104 levels.

T and G_{103} are matched in the following way. Vertex v_0 is mapped to the (only) vertex of level 0. Each triplet of T is mapped to three vertices of G_{103} such that the level of these three vertices is the same in G_{103}. Also, all triplets formed by vertices that are copies of the same vertex of T^* are mapped to consecutive levels of G_{103}. The exact mapping is described in Table 3. Each row of the table refers to a different vertex of T^* and shows the number of copies of that vertex in T, the number of triplets in T, and the levels of G_{103} to which these triplets are mapped (a triplet for each level).

We now prove that, with the mapping described by Table 3, T and G_{103} are not matched drawable if we insist that the drawing of G_{103} preserves the embedding of G_{103}. We start with a useful property.

Property 2 *Let $\Gamma_{G_{103}}$ be any planar straight-line drawing of G_{103} that preserves the embedding of G_{103}. For each level i ($0 \leq i \leq 103$) there exists a vertex of level i that has y-coordinate greater than the y-coordinates of all the vertices having level less than i.*

Lemma 3 *A matched drawing Γ_T and $\Gamma_{G_{103}}$ of the tree T and the graph G_{103} such that $\Gamma_{G_{103}}$ preserves the embedding of G_{103} does not exist.*

Proof: Let $\Gamma_{G_{103}}$ be any planar straight-line drawing of G_{103} that preserves the embedding of G_{103}. By exploiting Property 2, we can show that $\Gamma_{G_{103}}$ induces an ordering λ of the vertices of T along the y-direction such that there exists a subtree T' of T isomorphic to T^* for which the ordering λ restricted to the vertices of T' is the ordering given in Lemma 2. This implies that T' (and hence T) does not have a planar straight-line drawing that respects the ordering induced by $\Gamma_{G_{103}}$.

Denote by V_i the set of vertices of T that are copies of a vertex $v_i \in T^*$ ($i = 0, 1, \dots, 7$). We define subtree T' as follows. T' consists of eight vertices $\overline{v}_0, \overline{v}_1, \dots, \overline{v}_8$, where $\overline{v}_i \in V_i$. Of course, $\overline{v}_0 = v_0$. Vertex $\overline{v}_i$ is a vertex $v_{i,j}$ of V_i

such that: (i) the parent of $v_{i,j}$ is in T', in particular, it is $\overline{v}_{\lfloor j/3 \rfloor}$; and (ii) $v_{i,j}$ is the vertex of its level for which Property 2 holds. Notice that the isomorphism between T' and T^* is guaranteed by the fact that there is one vertex for each set V_i and that a vertex is selected only if its parent is also selected.

We write $V_i < V_j$ if all levels containing vertices of V_i are inside levels containing vertices of V_j in the embedding of G_{103}. Based on the mapping given in Table 3 we have that $V_0 < V_7 < V_3 < V_2 < V_4 < V_1 < V_5 < V_6$. This along with the fact that for each selected vertex Property 2 holds, implies that $y(\overline{v}_0) < y(\overline{v}_7) < y(\overline{v}_3) < y(\overline{v}_2) < y(\overline{v}_4) < y(\overline{v}_1) < y(\overline{v}_5) < y(\overline{v}_6)$. But by Lemma 2, T' does not admit a planar straight-line drawing such that the ordering of the vertices along the y-direction is the one given above. □

According to Lemma 3, T and G_{103} are not matched drawable in the case that one wants a drawing of G_{103} that preserves the embedding of G_{103}. In the following theorem we show that T and G_{103} can be used to construct a new tree and a new 3-connected planar graph that are not matched drawable even if we allow the embedding to be changed.

Theorem 2 *There exist a tree and a 3-connected planar graph that are not matched drawable.*

Proof: Let $\overline{T}$ be a tree that consists of two copies of T whose roots are adjacent. Let G be a graph obtained by taking two distinct copies of G_{103} and connecting the vertices of their external faces in such a way that the obtained graph is a triangulated planar graph. Denote as T' and T'' the two copies of T that form $\overline{T}$ and as G'_{103} and G''_{103} the two copies of G_{103} that form G. Also, define a mapping between $\overline{T}$ and G such that the vertices of T' are mapped to the vertices of G'_{103} according to the mapping defined by Table 3, and the vertices of T'' are mapped to the vertices of G''_{103} according to the mapping defined by Table 3. Since G is triangulated, it has a unique planar embedding except for the choice of the external face. Whatever face of G is chosen as the external one, the resulting embedding of G is such that either the embedding of G'_{103} or the embedding of G''_{103} is preserved. Thus either T' and G'_{103}, or T'' and G''_{103} are in the condition of Lemma 3 and therefore are not matched drawable. □

3 Matched Drawable Graphs

In this section we describe drawing algorithms for classes of graphs that are always matched drawable. In particular, in Section 3.1 we show that a planar graph and an unlabeled level planar (ULP) graph that are matched are always matched drawable. In Section 3.2 we extend these results to a planar graph and a graph of the family of "carousel graphs". Finally, in Section 3.3 we prove that two matched trees are always matched drawable.

These results show that matched drawings do indeed allow larger classes of graphs to be drawn than simultaneous geometric embeddings with mapping (a path and a planar graph may not admit a simultaneous geometric embedding with mapping [4] and the same negative result also holds for pairs of trees [17]).

3.1 Planar Graphs and ULP Graphs

ULP graphs were defined by Estrella-Balderrama, Fowler, and Kobourov [11]. Let G be a planar graph with n vertices. A *y-assignment* of the vertices of G is a one-to-one mapping $\lambda : V \rightarrow \mathbb{N}$. A *drawing of G compatible with* λ is a planar straight-line drawing of G such that $y(v) = \lambda(v)$ for each vertex $v \in V$. A planar graph G is *unlabeled level planar* (ULP) if for any given y-assignment λ of its vertices, G admits a drawing compatible with λ.

Theorem 3 *A planar graph and an ULP graph are always matched drawable.*

Proof: Let G_1 be a planar graph and let G_2 be an ULP graph. Compute a planar straight-line drawing of G_1 such that each vertex has a different y-coordinate, for example with a slight variant of the algorithm of de Fraysseix, Pach, and Pollack [6]. The drawing of G_1 together with the mapping between G_1 and G_2 defines a y-assignment λ for G_2. Since G_2 is ULP it admits a drawing compatible with λ. It follows that G_1 and G_2 are matched drawable. □

ULP trees are characterized in [11]. A complete characterization of ULP graphs is given in [13]. A planar graph is ULP if and only if it is either a *generalized caterpillar*, or a *radius*-2 *star*, or a *generalized degree-3 spider.* These graphs are defined as follows (see also [13]). A graph is a *caterpillar* if deleting all vertices of degree one produces a path, which is called the *spine* of the caterpillar. A *generalized caterpillar* is a graph that contains cycles of length at most 4 in which every spanning tree is a caterpillar such that no three cut vertices are pairwise adjacent and no pair of adjacent cut vertices belong to the same 4-cycle. A *radius*-2 *star* is a $K_{1,k}$, $k > 2$, in which every edge is subdivided at most once. The only vertex of degree k is called the *center* of the star. A *degree*-3 *spider* is an arbitrary subdivision of $K_{1,3}$. A *generalized degree-3 spider* is a graph with maximum degree 3 in which every spanning tree is either a path or a degree-3 spider.

Corollary 4 *Let G_1 and G_2 be two matched graphs such that G_1 is a planar graph and G_2 is either a generalized caterpillar, or a radius-2 star, or a generalized degree-3 spider. Then G_1 and G_2 are matched drawable.*

3.2 Planar Graphs and Carousel Graphs

In this section we extend the result of Theorem 3 by describing a family of graphs that also includes non-ULP graphs and whose members have a matched drawing with any planar graph. Let G be a planar graph, let v be a vertex of G, and let Γ be a planar straight-line drawing of G. Γ is *v-stretchable* if: (i) there is a vertical ray from v going to $+\infty$ that does not intersect any edge of Γ, and (ii) for any given $\Delta > 0$, there exists a value $\Delta' \geq \Delta$ such that the drawing obtained by translating each vertex u with $x(u) \geq x(v)$ to point $(x(u) + \Delta', y(u))$ is still planar. Graph G is ULP *v-stretchable* if for every given y-assignment λ of its vertices, G admits a v-stretchable drawing compatible with λ. For example, the graph G shown in Figure 5(a) is ULP [13]. Furthermore, it is easy to see that

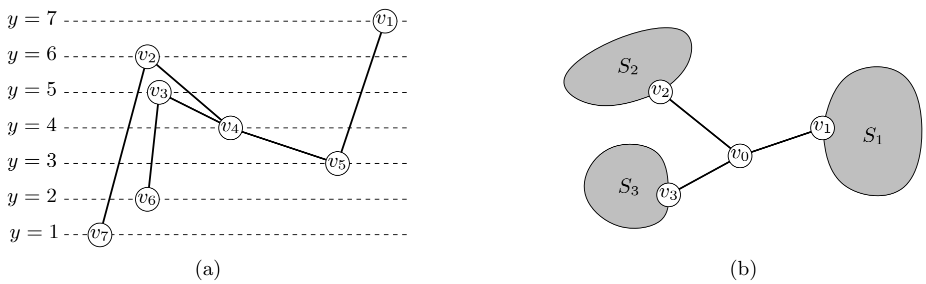

Figure 5: (a) A ULP graph which is ULP v_4-stretchable but not ULP v_3-stretchable. (b) A carousel graph.

for any possible y-assignment, G admits a v_4-stretchable drawing, and therefore G is ULP v_4-stretchable. On the other hand, for the y-assignment shown in Figure 5(a), G does not admit a drawing that is v_3-stretchable. Namely, in order to make v_3 visible from vertically above, the path from v_6 to v_4 must cross the path from v_7 to v_4 or the path from v_1 to v_4, or the paths from v_1 to v_4 and v_7 to v_4 must cross. Thus, G is not ULP v_3-stretchable.

A *carousel graph* is a connected planar graph G consisting of a vertex v_0, called the *pivot* of G, and of a set of disjoint subgraphs $S_1, \ldots, S_k$ $(k > 1)$ such that each S_i has a single vertex v_i adjacent to v_0 $(i = 1, \ldots, k)$ and S_i is ULP v_i-stretchable. Each subgraph S_i is called a *seat* of G. Vertex v_i is called the *hook* of S_i. Figure 5(b) illustrates the definition of a carousel graph.

Theorem 5 *Any planar graph and any carousel graph that are matched are matched drawable.*

Proof: Let G_1 be a planar graph and let G_2 be a carousel graph. Let v_0 be the pivot of G_2 and let u be the partner of v_0 in G_1. Compute a planar straight-line drawing Γ_1 of G_1 such that all vertices have different y-coordinates and u has the largest y-coordinate. Drawing Γ_1 together with the mapping between G_1 and G_2 defines a y-assignment λ for the vertices in a drawing Γ_2 of G_2. Clearly $\lambda(w) < \lambda(v_0) = y_M$ for all vertices $w \neq v_0$ of G_2.

In the following we describe an incremental method to compute a drawing Γ_2 compatible with λ. Let $S_1, \ldots, S_k$ $(k > 1)$ be the seats of G_2 and let v_i be the hook of S_i $(1 \leq i \leq k)$. Let λ_i be the y-assignment of the vertices of S_i induced by λ. As a preliminary step we compute a drawing Σ_i for each S_i that is compatible with λ_i and that is v_i-stretchable. Such a drawing exists because S_i is ULP v_i-stretchable. We further assume that the distance between any two different x-coordinates is at least 1 unit.

We incrementally construct Γ_2 in $k+1$ steps. Denote by $\Gamma_{2,i}$ the partial drawing of G_2 obtained at the end of step i $(0 \leq i \leq k)$. $\Gamma_{2,0}$ just consists of vertex v_0 placed at position $(0, y_M)$. Drawing $\Gamma_{2,i}$ is constructed from $\Gamma_{2,i-1}$ by adding Σ_i at a suitable x-location and possibly translating some of its vertices further in x-direction (see Figure 6). Hence the resulting drawing $\Gamma_{2,i}$ respects

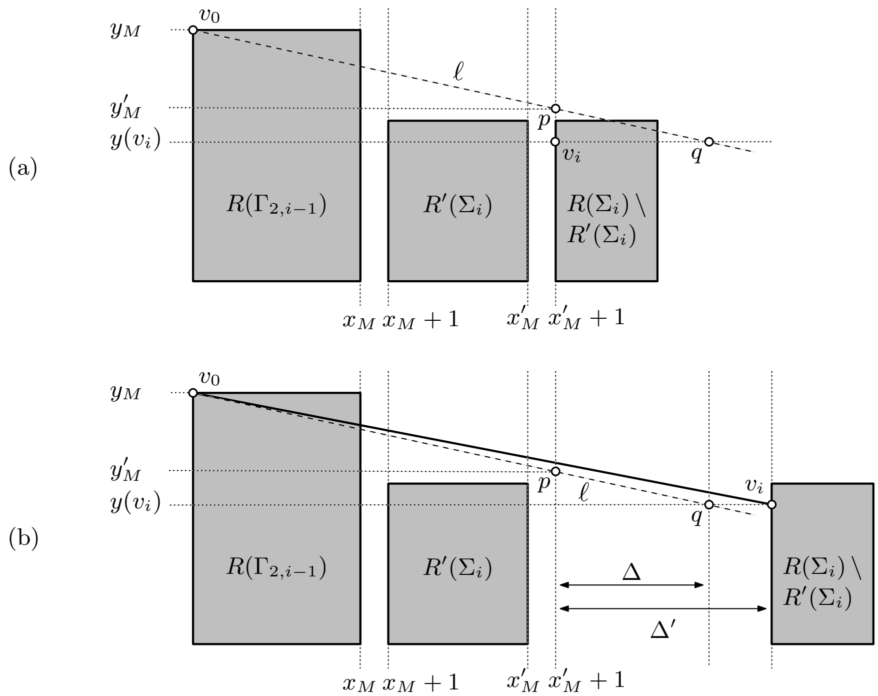

Figure 6: Adding Σ_i to $\Gamma_{2,i-1}$.

λ. The remainder of the proof focuses on the incremental step that adds Σ_i to $\Gamma_{2,i-1}$.

For a drawing Γ we denote by $R(\Gamma)$ the bounding box of Γ. Let (x_M, y_M) be the coordinates of the top-right corner of $R(\Gamma_{2,i-1})$. Place the drawing Σ_i such that the left side of $R(\Sigma_i)$ is contained in the vertical line $x = x_M + 1$. Let $R'(\Sigma_i)$ be the (possibly empty) sub-rectangle of $R(\Sigma_i)$ delimited by the x-coordinates $x_M + 1$ and $x'_M = x(v_i) - 1$. Furthermore, let y'_M denote the maximum y-coordinate of any vertex of $\Gamma_{2,i-1} \cup \Sigma_i$ distinct from v_0 and let $p = (x'_M + 1, y'_M)$; if $\Gamma_{2,i-1} \cup \Sigma_i$ does not have any vertex distinct from v_0 we let $p = (1, y_M - 1)$. The line ℓ through v_0 and p crosses neither $\Gamma_{2,i-1}$ nor the portion of Σ_i contained in $R'(\Sigma_i)$ (see Figure 6(a)). Let q denote the intersection of ℓ with the horizontal line at $y(v_i)$ and let $\Delta = x(q) - x(v_i)$. Since Σ_i is v_i-stretchable, there exists a value $\Delta' \geq \Delta$ such that we can translate the portion of Σ_i contained in $R(\Sigma_i) \setminus R'(\Sigma_i)$ to the right by Δ' without creating any crossing (see Figure 6(b)). It can easily be verified that the edge between v_0 and v_i does not have any crossings in the resulting drawing. □

Lemma 4 *Let G be a simple cycle and let v be any vertex of G. G is ULP v-stretchable.*

Proof: Let λ be any y-assignment of the vertices of G and let u be the vertex of G that has the smallest y-coordinate. Let $u = v_0, v_1, \ldots, v_{n-1}$ be the vertices of G in the order they are encountered when walking clockwise along G. Place each vertex v_i at point $(i, \lambda(v_i))$. Clearly none of the edges (v_i, v_{i+1}) ($i = 0, 1, \ldots, n-2$) cross each other. To avoid crossings between edge (v_0, v_{n-1}) and the other edges we translate v_{n-1} to the right until the segment connecting v_0 to v_{n-1} does not cross any other segment. By this construction it follows that such a drawing is v-stretchable for every vertex v of G. □

Corollary 6 *Let G_1 and G_2 be two matched graphs such that G_1 is a planar graph and G_2 is a cycle. Then G_1 and G_2 are matched drawable.*

The drawing techniques in [11] imply the following two lemmata.

Lemma 5 *Let G be a caterpillar and let v be a vertex of its spine. G is ULP v-stretchable.*

Lemma 6 *Let G be a radius-2 star and let v be the center of G. G is ULP v-stretchable.*

Corollary 7 *Let G_1 and G_2 be two matched graphs such that G_1 is a planar graph and G_2 is a connected graph consisting of a vertex*

v_0 and a set of disjoint subgraphs $S_1, S_2, \ldots S_k$, each S_i having a single vertex v_i connected to v_0. If each S_i is either a caterpillar with v_i on its spine, or a radius-2 star with v_i as its center, or a cycle, then G_1 and G_2 are matched drawable.

The family of carousel graphs described by Corollary 7 contains graphs that are not ULP. For example, the graph depicted in Figure 3 is a carousel graph with pivot v_2, and the three seats are caterpillars.

Remark: The proof of Theorem 5 is constructive, it can be used to to compute a matched drawing for the graphs G_1 and G_2. However, it may result in a matched drawing that has more than exponential size. For example, let G_1 be a path $u_0, \ldots, u_{n-1}$ and assume that its drawing Γ_1 assigns the y-coordinate $n - i$ to vertex u_i. Let G_2 be a star graph with v_0 as its center; G_2 obviously is a carousel graph where each of the $n - 1$ seats is a single vertex (see Figure 7). Let the matching be such that u_i and v_i are partners, and let the seats of G_2 be such that $S_i = v_i$ for $1 \leq i \leq n - 1$. Suppose that $v_0 = (0, n)$, that we have constructed $\Gamma_{2,i-1}$, and that $v_{i-1} = (x_{i-1}, n-i+1)$ for some integer value x_{i-1}. Then our construction will place v_i at $(x_i, n - i)$ with $x_i > i \cdot x_{i-1}$, because $p = (x_{i-1} + 1, n - 1)$ and $q = (i \cdot (x_{i-1} + 1), n - i)$. It follows that the largest x-coordinate x_{n-1} is more than exponential in n. We do not known whether a polynomial-size matched drawing of a planar graph and a carousel graph which are matched always exists.

3.3 Two Trees

Let T_1 and T_2 be two matched trees with n vertices each. We describe an algorithm to compute a matched drawing of T_1 and T_2 and prove its correctness.

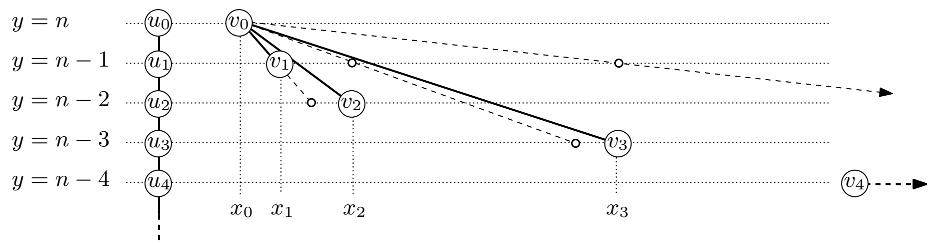

Figure 7: A matched drawing produced by our method that has more than exponential size.

The algorithm has two phases. In the first phase each vertex of a tree T_j ($j = 1, 2$) is assigned a distinct integer number from 1 to n, so that two matched vertices receive the same number; we denote by $\mathrm{ord}(v)$ the number assigned to a vertex v. Numbers are assigned to vertices in increasing order in n steps. In the second phase vertices are added to the drawing according to the order defined by the numbers assigned in the first phase.

To describe the two phases we need some definitions. A *chunk of rank i* is any tree of the forest obtained from T_1 or T_2 by removing all vertices v that are already processed and have $\mathrm{ord}(v) \leq i$. Notice that in Phase 1, a chunk of rank i is a tree of vertices that have not yet received a number at the end of Step i; in Phase 2, a chunk of rank i is a tree of vertices not yet drawn at the end of Step i. A chunk C of rank i can be adjacent only to vertices v such that $\mathrm{ord}(v)$ is defined and $\mathrm{ord}(v) \leq i$; we call these vertices the *anchor vertices of C*. At Step i ($1 \leq i \leq n$) the *pertinent tree of Step i* is T_1 if i is odd and T_2 if i is even; the other tree is the *non-pertinent tree of Step i*.

3.3.1 Description of Phase 1

Phase 1 consists of n steps. Number i is assigned to a vertex v of the pertinent tree of Step i; the same number is assigned to the partner of v. We maintain the following invariant throughout Phase 1:

Invariant 1 *For each integer $i \in [1, n]$:*

- *In the pertinent tree of Step i, every chunk of rank i has at most two anchor vertices;*
- *In the non-pertinent tree of Step i, there is at most one chunk of rank i with three anchor vertices, and every other chunk of rank i has at most two anchor vertices.*

At Step 1 the algorithm arbitrarily selects a vertex v of T_1 and sets $\mathrm{ord}(v) = 1$. Assume now that Invariant 1 holds at the end of Step $i-1$ ($i \geq 2$). Let T_j be the pertinent tree of Step i. Two cases are possible:

Case 1: In T_j, every chunk of rank $(i-1)$ has at most two anchor vertices. Let C be an arbitrary chunk of rank $(i-1)$ in T_j. The algorithm selects any vertex v of C, for example one that is adjacent to an anchor vertex of C, and sets $\mathrm{ord}(v) = i$ (see Figure 8).

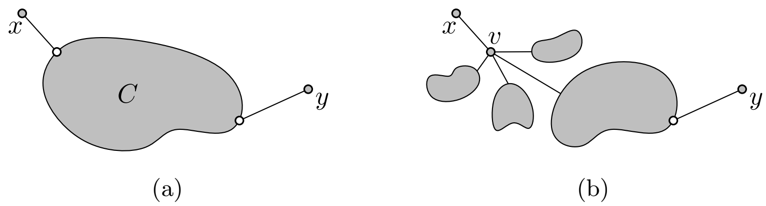

Figure 8: Illustration of Case 1: (a) Chunk C has two anchor vertices x and y. (b) Transformation of C after v is selected. In this figure v is chosen as one of the two vertices adjacent to the anchor vertices of C.

Case 2: In T_j, there exists a chunk C of rank $(i-1)$ with three anchor vertices. Let x, y, and z be the anchor vertices of C, and let π_1, π_2, and π_3 the three paths of T_j from x to y, from x to z, and from y to z, respectively. The algorithm selects the unique vertex v shared by π_1, π_2, and π_3, and sets $\mathrm{ord}(v) = i$ (see Figure 9).

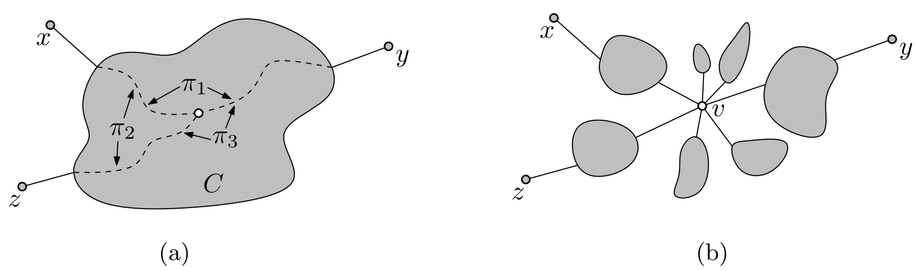

Figure 9: Illustration of Case 2: (a) Chunk C has three anchor vertices x, y, and z. Vertex v is the unique vertex shared by π_1, π_2, and π_3. (b) Transformation of C after v is selected.

Lemma 7 *Invariant 1 holds throughout Phase* 1 *of the algorithm.*

Proof: We prove the lemma by induction. The Invariant holds at Step 1 because all chunks of rank 1 (of both T_1 and T_2) are adjacent to the only vertex v with $\mathrm{ord}(v) = 1$. Assume by induction that Invariant 1 holds for $i-1$ ($i \geq 2$). Let T_j be the pertinent tree of Step i and let T_{3-j} be the non-pertinent tree of Step i. Let v be the vertex of T_j selected at Step i.

Assume first that v was selected according to Case 1. Let C be the chunk of rank $i-1$ that contains v. In this case, since C is a tree and since it has at most two anchor vertices, C is split into at most one chunk with two anchor vertices (one of which is v and the other one is an anchor vertex of C) and a certain number of chunks with v as the only anchor vertex (see Figure 8). Assume now that v was selected according to Case 2. Let C be the chunk of rank $i-1$ that contains v. Since C is a tree and since it has three anchor vertices, C is split into at most three chunks with two anchor vertices (one of which is v and the other one is an anchor vertex of C) and a certain number of chunks with v as the only anchor vertex (see Figure 9). Therefore Invariant 1 holds for T_j at Step i.

Let C' be the chunk of rank $i-1$ in T_{3-j} that contains the partner v' of v. By induction C' has at most two anchor vertices. Since C' is a tree, it is split into at most one chunk with three anchor vertices (one of which is v' and the other two are the anchor vertices of C') and a certain number of chunks with v' as the only anchor vertex (see Figure 10). Or, C' is split into at most two chunks with two anchor vertices and a certain number of chunks with v' as the only anchor vertex. This implies that Invariant 1 also holds for T_{3-j} at Step i. □

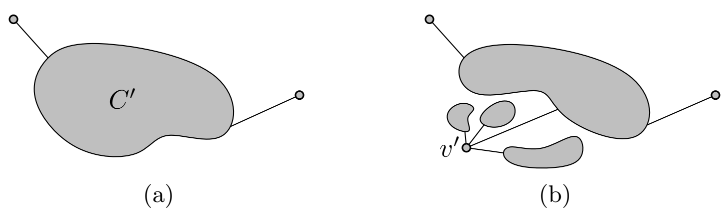

Figure 10: Creation of a chunk with three anchor vertices.

3.3.2 Description of Phase 2

Phase 2 also consists of n steps. At Step i the algorithm draws the two matched vertices numbered i in Phase 1. The y-coordinates are assigned as follows. Let v and v' be the two matched vertices with $\mathrm{ord}(v) = \mathrm{ord}(v') = i$; the algorithm sets $y(v) = y(v') = n - \frac{i-1}{2}$ if i is odd, and $y(v) = y(v') = \frac{i}{2}$, if i is even. In other words, vertices are assigned consecutively to y-coordinates $n, 1, n-1, 2, \ldots$. Thus, at the end of Step i there is no vertex drawn yet in the plane strip between the horizontal lines $y = n - \frac{i-1}{2}$ and $y = \frac{i-1}{2}$ if i is odd, and between the horizontal lines $y = n - \frac{i-2}{2}$ and $y = \frac{i}{2}$ if i is even. This strip is called the *strip of rank* i and it is assumed to be an open set (see Figure 11). The half-plane below the strip of rank i is called the *bottom side* of the drawing, while the half-plane above the strip of rank i is called the *top side* of the drawing. In order to assign the x-coordinates to the vertices, at Step i each chunk C of rank i is associated with a convex polygon P; C will be drawn inside P. We say that a polygon P *spans* the strip of rank i if each horizontal line $y = j$ with $j \in \mathbb{N}$ in the strip of rank i has non-empty intersection with the interior of P. An edge is drawn when both of its end-vertices are drawn. More precisely, let $e = (u, v)$ be

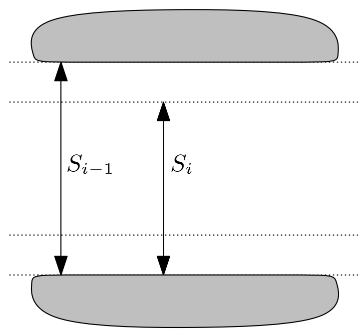

Figure 11: S_{i-1} is the strip of rank $i-1$ and S_i is the strip of rank i when i is assumed to be odd. The top side and bottom side of the drawing at Step $i-1$ are the grey parts above and below the strip.

an edge and let $\mathrm{ord}(u) = j$ and $\mathrm{ord}(v) = i$ with $j < i$. When vertex v is drawn at Step i, edge e is also drawn because u was drawn before, and we say that e is an *edge drawn at Step i*. We maintain the following invariant throughout Phase 2:

Invariant 2 *For each integer $i \in [1, n]$ and for each chunk C of rank i in any of the two trees, there exists a convex polygon P associated with C such that:*

- *The anchor vertices of C are corners of P;*
- *P spans the strip of rank i;*
- *The intersection between P and any edge e drawn at some Step j with $j \leq i$ is either empty or it consists of an end-vertex of e;*
- *The intersection between P and the polygon associated with any other chunk of rank i is either empty or it consists of a common corner;*

In what follows we describe how the algorithm assigns x-coordinates to the vertices of T_1. The x-coordinates of the vertices of T_2 are assigned analogously. At Step 1 vertex v with $\mathrm{ord}(v) = 1$ is given an arbitrary x-coordinate. Assume now that Invariant 2 holds at the end of Step $i-1$ ($i \geq 2$). Let v be the vertex with $\mathrm{ord}(v) = i$, let C be the chunk of rank $i-1$ that contains v, and let P be the polygon associated with C. We analyze the cases when i is odd and the cases when i is even, and their subcases.

Case 1: i is odd. Recall that by Invariant 1, when i is odd C can have three anchor vertices. If C has three anchor vertices, however, they cannot all be on the top side of the drawing. Namely, according to Phase 1, when a chunk with three anchor vertices is created, the next vertex that receives a number is chosen in such a way that the chunk has no longer three anchor vertices. This implies that if a chunk of rank $i-1$ has three anchor vertices, one of them is the vertex u with $\mathrm{ord}(u) = i-1$. Since $i-1$ is even, vertex u has been drawn at Step $i-1$ in the bottom side of the drawing. Therefore at least one anchor vertex is in the bottom side of the drawing. Let $C_1, C_2, \ldots, C_k$ be the chunks of rank i obtained by splitting C. Recall that, by Invariant 1 these chunks have at most two anchor points. The position of v and the polygons $P_1, P_2, \ldots, P_k$ associated with $C_1, C_2, \ldots, C_k$ are computed according to the cases below.

In **Cases 1.1**, **1.2**, and **1.3**, at most three chunks among $C_1, C_2, \ldots, C_k$ have two anchor vertices: one of them is v and the other one is an anchor vertex of C. All the other chunks have v as their only anchor vertex. In **Case 1.4** there are at most two chunks among $C_1, C_2, \ldots, C_k$ with two anchor vertices: one of them is v and the other one is an anchor vertex of C. All the other chunks have v as their only anchor vertex.

Case 1.1: C has three anchor vertices in the bottom side of the drawing. In this case vertex v is assigned an arbitrary x-coordinate such that the point representing v is in the interior of P. The polygons $P_1, P_2, \ldots, P_k$ are computed as shown in Figure 12. More precisely, denote as u_1, u_2, and u_3 the anchor vertices of C. Let C_1, C_2, and C_3 be the chunks having two anchor vertices. Assume that the anchor vertices of C_i are v and u_i ($1 \leq i \leq 3$). Since i is odd, the strip of rank i is defined by the two horizontal lines $y = n - \frac{i-1}{2}$ and $y = \frac{i-1}{2}$. Let ℓ be the horizontal line $y = \frac{i-1}{2} + 1$, which is contained in the strip of rank i. Let s_i be the segment connecting v to u_i ($1 \leq i \leq 3$), and let p_i be the intersection point between s_i and ℓ. Let p_0 and p_{k+1} be the intersection points between the border of P and the horizontal line ℓ. Assume, without loss of generality, that p_0, p_1, p_2, p_3, and p_{k+1} appear in this left-to-right order along ℓ. Let $p_4, p_5, \ldots, p_k$ be $k-3$ points on ℓ that fall, in this left-to-right order, between p_3 and p_{k+1}. For each point p_i ($1 \leq i \leq k$), choose two new points p_i^- and p_i^+ such that the left-to-right order along ℓ is $p_0, p_1^-, p_1, p_1^+, p_2^-, p_2, \ldots, p_{k-1}^+, p_k^-, p_k, p_k^+, p_{k+1}$. Polygon P_i associated with C_i ($1 \leq i \leq 3$) is the polygon whose corners are v, p_i^-, p_i^+, and u_i. Let q_i be the intersection point between the straight line through v and p_i and the border of P ($4 \leq i \leq k$). Polygon P_i associated with C_i ($4 \leq i \leq k$) is the polygon whose corners are v, p_i^-, p_i^+, and q_i.

Case 1.2: C has three anchor vertices, and two of them are in the top side of the drawing. Let Δ be the triangle whose corners are the anchor vertices of C. Notice that Δ is contained in P and spans the strip of rank i.

Vertex v is assigned an arbitrary x-coordinate such that the point representing v is in the interior of Δ. The polygons $P_1, P_2, \ldots, P_k$ are computed with an approach similar to that of **Case 1.1**. We omit the details and refer to Figure 13(a).

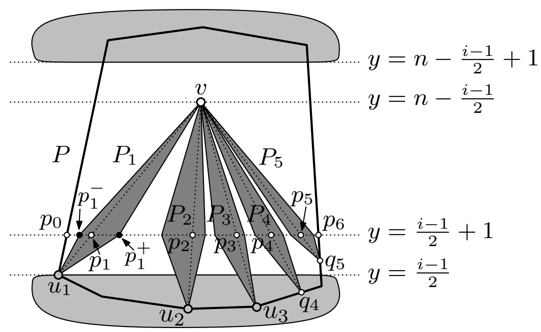

Figure 12: Illustration for **Case 1.1**.

Case 1.3: C **has three anchor vertices, and two of them are in the bottom side of the drawing.**

The x-coordinate of v is computed as in **Case 1.2**. The polygons $P_1, P_2, \ldots, P_k$ are computed as shown in Figure 13(b).

Case 1.4: C **has less than three anchor vertices.**

This case can be reduced to one of **Cases 1.2**, and **1.3** by selecting one or two corners of P as dummy anchor vertices. See Figure 13(c) for an example with two anchor vertices.

Case 2: i **is even.** By Invariant 1, when i is even C cannot have three anchor vertices. However, it may happen that at most one of the chunks of rank i obtained by splitting C has three anchor vertices. Let $C_1, C_2, \ldots, C_k$ be the chunks of rank i obtained by splitting C. The position of v and the polygons $P_1, P_2, \ldots, P_k$ associated with $C_1, C_2, \ldots, C_k$ are computed according to the following cases:

Case 2.1: No chunk of rank i **has three anchor vertices.** This case can be handled symmetrically to **Case 1.4**.

Case 2.2: A chunk of rank i **has three anchor vertices.** In this case C necessarily has two anchor vertices. Depending on the position of the two anchor vertices of C, we distinguish between three different cases. In all cases we consider a triangle Δ analogous to the one described in **Case 1.2**, i.e. (i) Δ is contained in P; (ii) all anchor vertices of P are corners of Δ; (iii) Δ spans the strip of rank i.

Case 2.2.1: The two anchor vertices of C **are in the bottom side of the drawing.** Vertex v is assigned an arbitrary x-coordinate such that the point representing v is on the border of Δ. The polygons $P_1, P_2, \ldots, P_k$ are computed as shown in Figure 13(d).

Case 2.2.2: The two anchor vertices of C **are in the top side of the drawing.** Vertex v is assigned an arbitrary x-coordinate such that the point representing v is in the interior of Δ. The polygons $P_1, P_2, \ldots, P_k$ are computed as shown in Figure 13(e).

Case 2.2.3: The two anchor vertices of C are in different sides of the drawing. Vertex v is assigned an arbitrary x-coordinate such that the point representing v is in the interior of Δ. The polygons $P_1, P_2, \ldots, P_k$ are computed as shown in Figure 13(f).

In all cases above, let u be an anchor vertex of C. If u and v are not adjacent, then there exists a chunk C_j of rank i ($0 \leq j \leq k$), and Figures 12 and 13 show how to compute a polygon P_j associated with it. If u and v are adjacent, then chunk C_j does not exist, polygon P_j is not defined and edge (u, v) is drawn as a straight-line segment. It follows that the intersection between the segment representing (u, v) and the polygons associated with the chunks of rank i (or edges connecting v to other anchor vertices) consists of the single vertex v. Hence, Invariant 2 is maintained.

Theorem 8 *Any two trees are matched drawable.*

Proof: Let T_1 and T_2 be two matched trees. We prove that the algorithm described above correctly computes a matched drawing of T_1 and T_2. By Lemma 7, Phase 1 computes an order of the vertices that satisfies Invariant 1. Phase 2 uses this order to draw the vertices.

First of all, notice that in each of the cases considered in the description of Phase 2, a point to represent v exists. Namely, in all cases v has a y-coordinate that is assigned depending only on the value of i: it is either $y = n - \frac{i-1}{2}$, or $y = \frac{i}{2}$. So in each case v must be drawn on a point of a horizontal line ℓ that is either $y = n - \frac{i-1}{2}$, or $y = \frac{i}{2}$. In **Case 1.1** the algorithm chooses a point of ℓ that is inside P. Since P spans the strip of rank i, the intersection between the interior of P and ℓ is not empty. In all other cases the algorithm chooses a point that is either in the interior of triangle Δ, or on its border. Since the number of anchor points of C is at most three, and since if there are three anchor vertices then they are on different sides (because otherwise we are in **Case 1.1**), a triangle Δ exists with three corners a, b, and c such that: (i) a, b, and c are corners of P; (ii) all anchor vertices of C are in the set $\{a, b, c\}$; (iii) a, b, and c are not all on the same side of the drawing. By construction, Δ is contained in P and all anchor vertices of C are corners of Δ. Also, Δ spans the strip of rank i because it has at least one corner in the bottom side of the drawing and at least one corner in the top side of the drawing. Since Δ spans the strip of rank i, at least one point of ℓ inside P exists that can be used to represent v.

Invariant 2 holds throughout Phase 2 by construction. It remains to prove that the drawings computed by the algorithm form a matched drawing of T_1 and T_2. Since two matched vertices have the same y-coordinate, we only need to show that the drawings of T_1 and T_2 are planar. We prove this for T_1; an analogous proof holds for T_2.

Consider two edges e_1 and e_2 in the drawing of T_1. Assume that e_1 is an edge drawn at Step j, and that e_2 is an edge drawn at Step i, with $j \leq i$. If

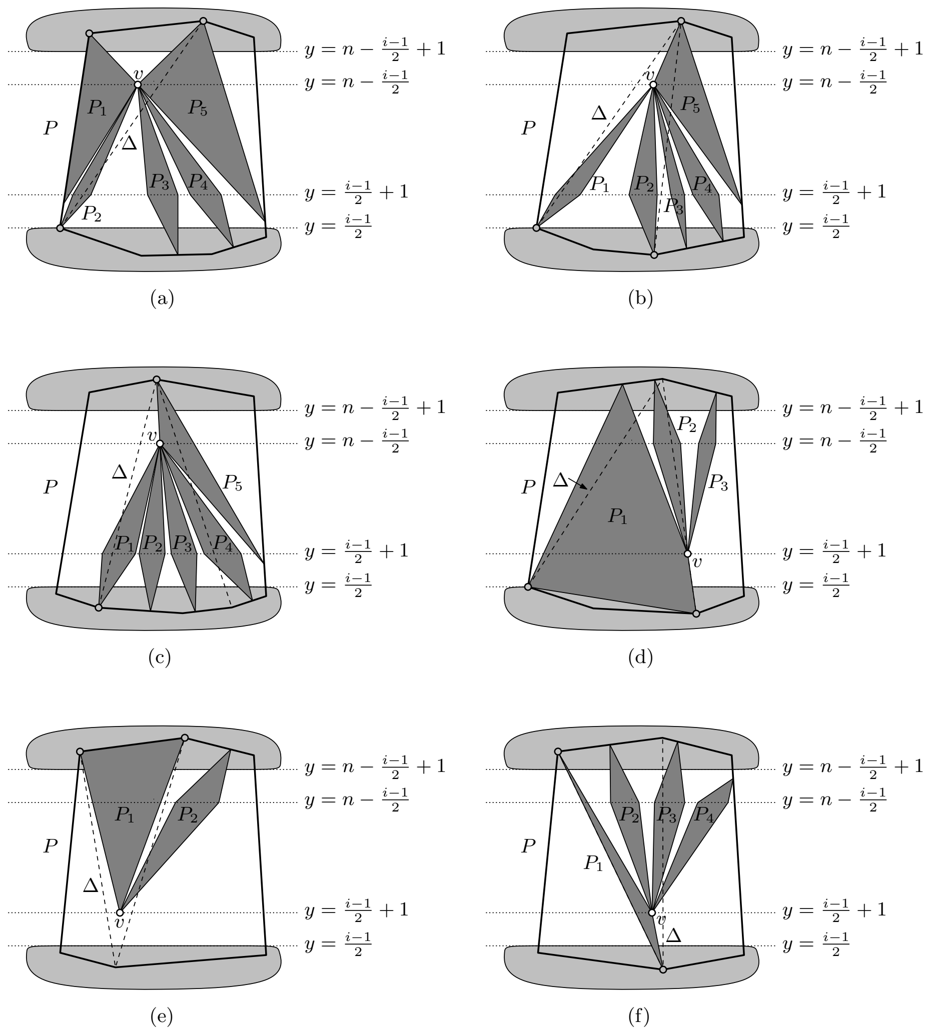

Figure 13: (a) **Case 1.2**; (b) **Case 1.3**; (c) **Case 1.4**; (d) **Case 2.2.1**; (e) **Case 2.2.2**; (f) **Case 2.2.3**.

$j = i$ then e_1 and e_2 share an endvertex (the one drawn at Step i) and they cannot cross. If $j < i$, edge e_1 is drawn before edge e_2. Let v be the endvertex of e_2 that is drawn at Step i, let C be the chunk of rank $i-1$ that contains v, and let P be the polygon associated with C. Edge e_2 is drawn inside P, since e_2 connects v to an anchor vertex of C, which is a corner of P. By Invariant 2, the intersection between P and e_1 is either empty or it consists of an endvertex of e_1. Thus e_1 and e_2 either have no intersection or they share a common endvertex. □

4 Conclusions and Open Problems

In this paper we introduced the concept of matched drawings, which are a natural way to draw two planar graphs whose vertex sets are matched. Since this is the first study of these drawings, many interesting and challenging open problems remain. First of all, in the light of Theorems 2 and 5, we would like to characterize the subclass of planar graphs that admit a matched drawing with any planar graph. Secondly, the drawing techniques of Theorems 5 and 8 may give rise to drawings where the area is at least exponential in the size of the graphs. It would be interesting to determine for what classes of graphs polynomial-size matched drawings exist. On a related note, some of our drawing techniques rely on a planar straight-line drawing of a planar graph where each vertex has a different y-coordinate. How big a grid is necessary to guarantee such a drawing with integer coordinates? Another question concerns the counterexample for a planar graph and a tree described in Section 2.2, which consists of 620 vertices. It would be nice to construct a counterexample with a smaller number of vertices. And finally, given any two matched graphs, what is the algorithmic complexity of testing whether they are matched drawable?

References

[1] U. Brandes and T. Erlebach, editors. *Network Analysis*, volume 3418 of *LNCS*. Springer, 2005.

[2] U. Brandes, C. Erten, J. Fowler, F. Frati, M. Geyer, C. Gutwenger, S.-H. Hong, M. Kaufmann, S. Kobourov, G. Liotta, P. Mutzel, and A. Symvonis. Colored simultaneous geometric embeddings. In *Proc. 13th Annual International Computing and Combinatorics Conference*, volume 4598 of *LNCS*, pages 254–263, 2007.

[3] J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, number 2025 in LNCS, pages 228–246. Springer, 2001.

[4] 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 planar graph embeddings. *Computational Geometry: Theory and Applications*, 36(2):117–130, 2007.

[5] J. Cappos, A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Simultaneous graph embedding with bends and circular arcs. In *Proc. 14th International Symposium on Graph Drawing (GD 2006)*, volume 4372 of *LNCS*, pages 95–107, 2007.

[6] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[7] C. Demetrescu, G. Di Battista, I. Finocchi, G. Liotta, M. Patrignani, and M. Pizzonia. Infinite trees and the future. In J. Kratochvíl, editor, *Proc. 7th International Symposium on Graph Drawing (GD'99)*, volume 1731 of *LNCS*, pages 379–391. Springer, 1999.

[8] E. Di Giacomo, W. Didimo, L. Grilli, and G. Liotta. Graph visualization techniques for web clustering engines. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):294–304, 2007.

[9] E. Di Giacomo and G. Liotta. Simultaneous embedding of outerplanar graphs, paths, and cycles. *International Journal of Computational Geometry and Applications*, 17(2):139–160, 2007.

[10] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *Journal of Graph Algorithms and Applications*, 9(3):347–364, 2005.

[11] A. Estrella-Balderrama, J. J. Fowler, and S. G. Kobourov. Characterization of unlabeled level planar trees. In *Proc. 14th International Symposium on Graph Drawing (GD 2006)*, volume 4372 of *LNCS*, pages 367–379. Springer, 2007.

[12] H. Fernau, M. Kaufmann, and M. Poths. Comparing trees via crossing minimization. In *Proc. 25th Conference on Foundations of Software Technology and Theoretical Computer Science*, number 3821 in LNCS, pages 457–469, 2005.

[13] J. J. Fowler and S. G. Kobourov. Characterization of unlabeled level planar graphs. In *Proc. 15th International Symposium on Graph Drawing (GD 2007)*, volume 4875 of *LNCS*, pages 37–49. Springer, 2008.

[14] F. Frati. Embedding graphs simultaneously with fixed edges. In *Proc. 14th International Symposium on Graph Drawing (GD 2006)*, volume 4372 of *LNCS*, 2007.

[15] F. Frati, M. Kaufmann, and S. G. Kobourov. Constrained simultaneous and near-simultaneous embeddings. In *Proc. 15th International Symposium on Graph Drawing (GD 2007)*, volume 4875 of *LNCS*, pages 268–279. Springer, 2008.

[16] C. Friedrich and P. Eades. Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6(3):353–370, 2002.

[17] M. Geyer, M. Kaufmann, and I. Vrto. Two trees which are self-intersecting when drawn simultaneously. In *Proc. 13th International Symposium on Graph Drawing (GD 2005)*, volume 3843 of *LNCS*, pages 201–210, 2006.

[18] M. L. Huang, P. Eades, and R. F. Cohen. WebOFDAV-navigating and visualising the web online with animated context swapping. *Computer Networks and ISDN Systems*, 30(1–7):638–642, 1998.

[19] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, 1995.

[20] S. North. Incremental layout in DynaDAG. In *Proc. 3rd International Symposium on Graph Drawing (GD'95)*, volume 1027 of *LNCS*, pages 409–418. Springer, 1996.

[21] A. Papakostas and I. G. Tollis. Interactive orthogonal graph drawing. *IEEE Transactions on Computers*, 47(11):1297–1309, 1998.