Simultaneous Drawing of Planar Graphs with Right-Angle Crossings and Few Bends*

Michael A. Bekos[1], Thomas C. van Dijk[2], Philipp Kindermann[2], and Alexander Wolff[2]

[1] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany.
bekos@informatik.uni-tuebingen.de
[2] Lehrstuhl für Informatik I, Universität Würzburg, Germany.
http://www1.informatik.uni-wuerzburg.de/en/staff

Abstract. Given two planar graphs that are defined on the same set of vertices, a *RAC simultaneous drawing* is a drawing of the two graphs where each graph is drawn planar, no two edges overlap, and edges of one graph can cross edges of the other graph only at right angles. In the geometric version of the problem, vertices are drawn as points and edges as straight-line segments. It is known, however, that even pairs of very simple classes of planar graphs (such as wheels and matchings) do not always admit a geometric RAC simultaneous drawing.
In order to enlarge the class of graphs that admit RAC simultaneous drawings, we allow edges to have bends. We prove that any pair of planar graphs admits a RAC simultaneous drawing with at most six bends per edge. For more restricted classes of planar graphs (e.g., matchings, paths, cycles, outerplanar graphs, and subhamiltonian graphs), we significantly reduce the required number of bends per edge. All our drawings use quadratic area.

1 Introduction

A simultaneous embedding of two planar graphs embeds each graph in a planar way—using the same vertex positions for both embeddings. Edges of one graph are allowed to intersect edges of the other graph. There are two versions of the problem: In the first version, called *Simultaneous Embedding with Fixed Edges* (SEFE), edges that occur in both graphs must be embedded in the same way in both graphs (and hence, cannot be crossed by any other edge). In the second version, called *Simultaneous Embedding*, these edges can be drawn differently for each of the graphs. Both versions of the problem have a geometric variant where edges must be drawn using straight-line segments.

Simultaneous embedding problems have been extensively investigated over the last few years, starting with the work of Brass et al. [6] on simultaneous straight-line drawing problems. Bläsius et al. [5] recently surveyed the area. For example, it is possible to decide in linear time whether a pair of graphs admits a SEFE or not, if the common

* This research was supported by the ESF EuroGIGA project GraDR (DFG grant Wo 758/5-1). The work of M.A. Bekos is implemented within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

M.S. Rahman and E. Tomita (Eds.): WALCOM 2015, LNCS 8973, pp. 222–233, 2015.

graph is biconnected [1]. When actually drawing these simultaneous embeddings, a natural choice is to use straight-line segments. Only very few graphs can be drawn in this way, however, and some existing results require exponential area. For instance, there exist a tree and a path which cannot be drawn simultaneously with straight-line segments [2], and the algorithm for simultaneously drawing a tree and a matching [8] does not provide a polynomial area bound. For the case of edges with bends, that is, polygonal edges, Erten and Kobourov [10] showed that three bends per edge and quadratic area suffice for any pair of planar graphs (without fixed edges), and that one bend per edge suffices for pairs of trees. Kammer [11] reduced the number of bends to two for the general planar case. In these results, however, the *crossing angles* can be very small.

We suggest a new approach that overcomes the aforementioned problems. We insist that crossings occur at right angles, thereby "taming" them. We do this while drawing all vertices and all bends on a grid of size $O(n) \times O(n)$ for any n-vertex graph, and we can still draw any pair of planar graphs simultaneously. We do not consider the problem of fixed edges. In a way, our results give a measure for the geometric complexity of simultaneous embeddability for various pairs of graph classes, some of which can be combined more easily (that is, with fewer bends) than others.

Brightwell and Scheinermann [7] proved that the problem of simultaneously drawing a (primal) embedded graph and its dual always admits a solution if the input graph is a triconnected planar graph. Erten and Kobourov [9] presented an $O(n)$-time algorithm that computes simultaneous drawings of a triconnected planar graph and its dual on an integer grid of size $O(n^2)$, where n is the total number of vertices in the graph and its dual. However, these drawings can have non-right angle crossings.

In this paper, we study the *RAC simultaneous (*RACSIM*) drawing problem*. Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two planar graphs on the same vertex set. We say that G_1 and G_2 admit a RACSIM drawing if we can place the vertices on the plane such that (i) each edge is drawn as a polyline, (ii) each graph is drawn planar, (iii) there are no edge overlaps, and (iv) crossings between edges in E_1 and E_2 occur at right angles.

Argyriou et al. [3] introduced and studied the geometric version of RACSIM drawing. In particular, they proved that any pair of a cycle and a matching admits a geometric RACSIM drawing on a grid of quadratic size, while there exists a pair of a wheel and a cycle that does not admit a geometric RACSIM drawing. The problem that we study was left as an open problem.

Our contribution. Our main result is that any pair of planar graphs admits a RAC simultaneous drawing with at most six bends per edge. We can compute such drawings in linear time. For pairs of subhamiltonian graphs and pairs of outerplanar graphs, we need four bends and three bends per edge, respectively; see Section 2. Then, we turn our attention to pairs of more restricted graph classes where we can guarantee one bend per edge or two bends per edge; see Sections 3 and 4, respectively. Table 1 lists our results. The main approach of all our algorithms is to find linear orders on the vertices of the two graphs and then to compute coordinates for the vertices based on these orders. All crossings in our drawings appear between horizontal and vertical edge segments. We call the non-rectilinear edge segments *slanted* segments. All our drawings fit on a grid of quadratic size. Due to lack of space, some proofs are only sketched. The corresponding detailed proofs are given in the full version [4].

Table 1. A short summary of our results

Graph classes			Number of bends	Ref.
planar	$+$	planar	$6+6$	Thm. 1
subhamiltonian	$+$	subhamiltonian	$4+4$	Cor. 1
outerplanar	$+$	outerplanar	$3+3$	Thm. 2
cycle	$+$	cycle	$1+1$	Thm. 3
caterpillar	$+$	cycle	$1+1$	Thm. 4
four matchings			$1+1+1+1$	Thm. 5
tree	$+$	matching	$1+0$	Thm. 6
wheel	$+$	matching	$2+0$	Thm. 7
outerpath	$+$	matching	$2+1$	Thm. 8

2 RacSim Drawings of General Graphs

In this section, we study general planar graphs and show how to efficiently construct RacSim drawings with few bends per edge in $O(n) \times O(n)$ area. We prove that two planar graphs on a common set of n vertices admit a RacSim drawing with six bends per edge (Theorem 1). For pairs of subhamiltonian graphs, we lower the number of bends per edge to 4 (Corollary 1) and for pairs of outerplanar graphs to 3 (Theorem 2). Recall that the class of subhamiltonian graphs is equivalent to the class of 2-page book-embeddable graphs, and the class of outerplanar graphs is equivalent to the class of 1-page book-embeddable graphs.

Central to our approach is an algorithm by Kaufmann and Wiese [12] that embeds any planar graph such that vertices are mapped to points on a horizontal line (called *spine*) and each edge crosses the spine at most once; see Fig. 1(a). We introduce a *dummy vertex* at each spine crossing. This yields a linear order of the (original and dummy) vertices with the property that every edge is either above or below the spine. For our problem, in order to determine the locations of the (original and dummy) vertices of the two given graphs, we basically use the linear order induced by one graph for the x-coordinates and the order induced by the other graph for the y-coordinates. (We reserve additional rows and columns for routing the edges of the first and second graph, respectively.) Let R be the bounding box of the vertex positions. Then, for the first graph, we route, in a planar fashion, the above/below-edges using short slanted segments and long vertical segments inside R as well as horizontal segments above/below R; see Fig. 1(c). We treat the second graph analogously, but turn the drawing by $90°$. As the following theorem assures, the resulting simultaneous drawing has only right-angle crossings; see Fig. 1(d). Note that the algorithm of Kaufmann and Wiese has been used for simultaneous drawing problems before [10].

Theorem 1. *Two planar graphs on a common set of n vertices admit a* RacSim *drawing on an integer grid of size $(14n-26) \times (14n-26)$ with six bends per edge. The drawing can be computed in $O(n)$ time.*

Proof. Let $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ be the given planar graphs. For $m = 1, 2$, let ξ_m be an embedding of $\mathcal{G}_m$ according to the algorithm of Kaufmann and Wiese,

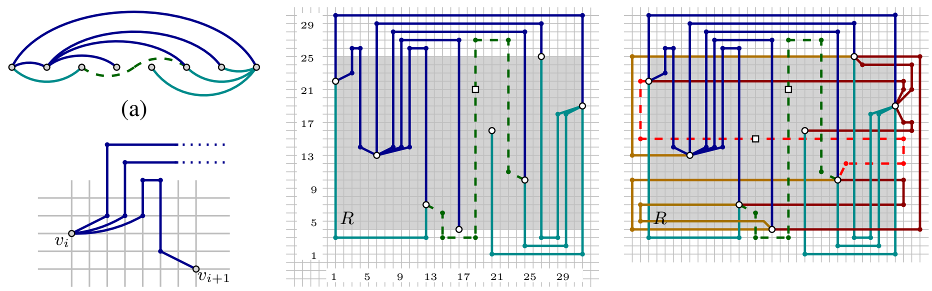


Fig. 1. (a) A drawing of a planar graph by Kaufmann and Wiese [12], (b) reserving additional columns between two vertices, (c) the graph in (a) drawn by our algorithm with at most six bends per edge, and (d) the RACSIM drawing of two planar graphs by our algorithm. The edges that cross the spine are drawn dashed; the dummy vertices on these edges are drawn as squares.

and denote by A_m and B_m the edges that are drawn completely above and below the spine in ξ_m, respectively. Further, let $\mathcal{G}'_m = (V'_m, E'_m) = (V \cup V_m, A_m \cup B_m)$ be the resulting graph, where V_m is the set of dummy vertices of $\mathcal{G}_m$.

We now show how to determine the x-coordinates of the vertices in V'_1; the y-coordinates of the vertrices in V'_2 are determined analogously. The missing y-coordinates of the dummy vertices in V_1 and the missing x-coordinates of the dummy vertices in V_2 are arbitrary (as long as they are inside R).

Let n'_1 be the number of vertices in V'_1 and let $v_1, \ldots, v_{n'_1}$ be the linear order of the vertices in V'_1 along the spine in ξ_1. We place v_1 in the first column. Between any two consecutive vertices v_i and v_{i+1}, we reserve several columns for the bends of the edges incident to v_i and v_{i+1}, in the following order; see Fig. 1(b):

(i) a column for the first bend on all edges leaving v_i in A_2;
(ii) a column for each edge $(v_i, v_j) \in E'_1$ with $j > i$;
(iii) a column for each edge $(v_k, v_{i+1}) \in E'_1$ with $k \leq i$;
(iv) a column for the last bend on all edges entering v_{i+1} in B_2.

Note that, for (ii) and (iii), we can save some columns because an edge in A_1 and an edge in B_1 can use the same column for their bend. Further, we may save the additional column of (i) and (iv) if no such edges exist. Now, we draw $\mathcal{G}'_1$ and $\mathcal{G}'_2$ with at most four bends per edge such that all edge segments of $\mathcal{G}'_1$ in R are either vertical or of y-length exactly 1, and all edge segments of $\mathcal{G}'_2$ in R are either horizontal or of x-length exactly 1; see Fig. 1(d).

First, we draw the edges $(v_i, v_j) \in A_1$ with $i < j$ in a nested order: When we place the edge (v_i, v_j), then there is no edge $(v_k, v_l) \in A_1$ with $k \leq i$ and $l \geq j$ that has not already been drawn. Recall that the first column to the right and the first column to the left of every vertex is reserved for the edges in E_1; hence, we assume that they are already used. We draw (v_i, v_j) with at most four bends as follows. We start with a slanted segment that has its endpoint in the row above v_i and in the first unused column that does not lie to the left of v_i. We follow with a vertical segment to the top that leaves R. We add a horizontal segment above R. In the last unused column that does not lie to the right of v_j, we add a vertical segment that ends one row above v_j. We close the edge with a slanted segment that has its endpoint in v_j. We draw the edges in B_1 symmetrically with the horizontal segment below R.

Note that this algorithm always uses the top and the bottom port of a vertex v, if there is at least one edge incident to v in A_1 and B_1, respectively. There is exactly one edge incident to each dummy vertex t in A_1 and B_1, respectively. Thus, the edges incident to t only use the top and the bottom port. We create a drawing of $\mathcal{G}_1$ and $\mathcal{G}_2$ with at most 6 bends per edge by removing the dummy vertices from the drawing.

We now show that combining the drawings of $\mathcal{G}_1$ and $\mathcal{G}_2$ yields a RACSIM drawing. By construction, all segments of E_1 inside R are either vertical segments or slanted segments of y-length 1, and all segments of E_2 inside R are either horizontal segments or slanted segments of x-length 1. Thus, the slanted segments cannot overlap. Furthermore, all crossings inside R occur between a horizontal and a vertical segment, and thus form right angles. Also, there are no segments in E_1 that lie to the left or to the right of R, and there are no segments in E_2 that lie above or below R. Hence, there are no crossings outside R, and the drawing is a RACSIM drawing.

We now count the columns used by the drawing. For the leftmost and the rightmost vertex, we reserve one additional column for its incident edges in E_2; for the remaining vertices, we reserve two such columns. For each edge in E_1, we need up to three columns: one for each endpoint of the slanted segment at each vertex and one for the vertical segment that crosses the spine, if it exists. Note that at least one edge per vertex does not need a slanted segment. For each edge in E_2, we need at most one column for the vertical segment to the side of R. Since there are at most $3n-6$ edges, we need at most $3n-2+3\cdot(3n-6)-n+3n-6=14n-26$ columns. By symmetry, we need the same number of rows.

Since the algorithm of Kaufmann and Wiese runs in $O(n)$ time, our algorithm also runs in $O(n)$ total time. □

We can improve the results of Theorem 1 for subhamiltonian graphs. Recall that a subhamiltonian graph has a 2-page book embedding, in which no edges cross the spine. Since such edges are the only ones that need six bends, we can reduce the number of bends per edge to four. Further, the number of columns and rows are reduced by one per edge. This yields the following corollary.

Corollary 1. *Two subhamiltonian graphs on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $(11n-32)\times(11n-32)$ with four bends per edge.*

Theorem 2. *Two outerplanar graphs on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $(7n-10)\times(7n-10)$ with three bends per edge.*

Proof. Let $\mathcal{O}_1=(V,E_1)$ and $\mathcal{O}_2=(V,E_2)$ be the given outerplanar graphs. First, we create a 1-page book embedding for $\mathcal{O}_1$ and $\mathcal{O}_2$. This gives us the order of the x-coordinates and y-coordinates, respectively. It follows by Corollary 1 that, by using the algorithm described in the proof of Theorem 1, we obtain a RACSIM drawing with at most four bends per edge. We will now show how to adjust the algorithm to reduce the number of bends by one.

It follows by Nash-Williams' formula [13] that every outerplanar graph has arboricity 2, that is, it can be decomposed into two forests. We embed both graphs on two pages with one forest per page. Let A_1 and B_1 be the two forests $\mathcal{O}_1$ is decomposed into. We will draw the edges of A_1 above the spine and the edges B_1 below the spine. By rooting the trees in A_1 in arbitrary vertices, we can direct each edge such that every vertex has exactly one incoming edge. Recall that, in the drawing produced in Theorem 1, one edge per vertex can use the top port. We adjust the algorithm such that every directed edge (v, w) enters the vertex w from the top port. To do so, we draw the edge as follows. We start with a slanted segment of y-length exactly 1. We follow with a vertical segment to the top. We proceed with a horizontal segment that ends directly above w and finish the edge with a vertical segment that enters w from the top port. We use the same approach for the edges in B_1, using the bottom port. We treat the second outerplanar graph $\mathcal{O}_2$ analogously, but turn the drawing by $90°$.

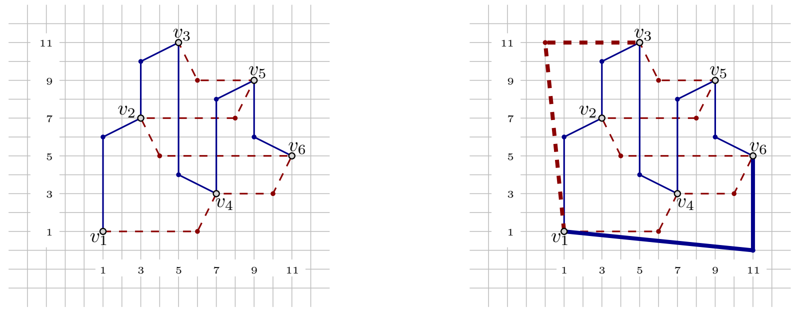
Since every port of a vertex is only used once, the drawing has no overlaps. We now analyze the number of columns used. For every vertex except for the leftmost and rightmost, we again reserve two additional columns for the edges in E_2; for the remaining two vertices, we reserve one additional column. However, the edges in E_1 now only need one column for the bend of the single slanted segment. For every edge in E_2, we need up to one column for the vertical segment to the side of R. Since there are at most $2n - 4$ edges, our drawing needs $3n - 2 + 2n - 4 + 2n - 4 = 7n - 10$ columns. Analogously, we can show that the algorithm needs $7n - 10$ rows. □

3 RACSIM Drawings with One Bend Per Edge

In this section, we study simple classes of planar graphs and show how to efficiently construct RACSIM drawings with one bend per edge in quadratic area. In particular, we prove that two cycles or four matchings on a common set of n vertices admit a RACSIM drawing on an integer grid of size $2n \times 2n$; see Theorems 3 and 5, respectively. If the input to our problem is a caterpillar and a cycle, then we can compute a RACSIM drawing with one bend per edge on an integer grid of size $(2n-1) \times 2n$; see Theorem 4. For a tree and a cycle, we can construct a RACSIM drawing with one bend per tree edge and no bends in the matching edges on an integer grid of size $n \times (n-1)$; see Theorem 6.

Lemma 1. *Two paths on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $2n \times 2n$ with one bend per edge. The drawing can be computed in $O(n)$ time.*

Proof. Let $\mathcal{P}_1 = (V, E_1)$ and $\mathcal{P}_2 = (V, E_2)$ be the two input paths. Following standard practices from the literature (see, e.g., Brass et al. [6]), we draw $\mathcal{P}_1$ x-monotone and $\mathcal{P}_2$ y-monotone. This ensures that the drawing of both paths will be planar. We will now describe how to compute the exact coordinates of the vertices and how to draw the edges of $\mathcal{P}_1$ and $\mathcal{P}_2$, such that all crossings are at right angles and, more importantly, no edge segments overlap.



(a) two paths: $\mathcal{P}_1$ (solid) and $\mathcal{P}_2$ (dashed)

(b) two cycles: $\mathcal{C}_1$ (solid) and $\mathcal{C}_2$(dashed)

Fig. 2. RACSIM drawings with one bend per edge

For $m = 1, 2$ and any vertex $v \in V$, let $\pi_m(v)$ be the position of v in $\mathcal{P}_m$. Then, v is drawn at the point $(2\pi_1(v) - 1, 2\pi_2(v) - 1)$; see Fig. 2a. It remains to determine, for each edge $e = (v, v')$, where it bends. First, assume that $e \in E_1$ and e is directed from its left to its right endpoint. Then, we place the bend at $v' - (2, \operatorname{sgn}(y(v') - y(v)))$. Second, assume that $e \in E_2$ and e is directed from its bottom to its top endpoint. Then, we place the bend at $v' - (\operatorname{sgn}(x(v') - x(v)), 2)$.

Clearly, the area required by the drawing is $(2n - 1) \times (2n - 1)$. The edges of $\mathcal{P}_1$ leave the left endpoint vertically and enter the right endpoint with a slanted segment of x-length 1 and y-length 2. Similarly, the edges of $\mathcal{P}_2$ leave the the bottom endpoint horizontally and enter the top endpoint with a slanted segment of x-length 2 and y-length 1. Hence, the slanted segments cannot be involved in crossings or overlaps. Since $\mathcal{P}_1$ and $\mathcal{P}_2$ are x- and y-monotone, respectively, it follows that all crossings must involve a vertical edge segment of $\mathcal{P}_1$ and a horizontal edge segment of $\mathcal{P}_2$, which clearly yields right angles at the crossing points. $\square$

We say that an edge uses the bottom/left/right/top *port* of a vertex if it enters the vertex from the bottom/left/right/top.

Theorem 3. *Two cycles on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $2n \times 2n$ with at most one bend per edge. The drawing can be computed in $O(n)$ time.*

Proof. Let $\mathcal{C}_1 = (V, E_1)$ and $\mathcal{C}_2 = (V, E_2)$ be the two input cycles, and let $v \in V$ be an arbitrary vertex. We temporarily delete one edge $(v, w_1) \in E_1$ from $\mathcal{C}_1$ and $(v, w_2) \in E_2$ from $\mathcal{C}_2$ (refer to the bold-drawn edges of Figure 2b). This way, we obtain two paths $\mathcal{P}_1 = \langle v, \ldots, w_1 \rangle$ and $\mathcal{P}_2 = \langle v, \ldots, w_2 \rangle$. We employ the algorithm desribed in Lemma 1 to construct a RACSIM drawing of $\mathcal{P}_1$ and $\mathcal{P}_2$ on an integer grid of size $(2n - 1) \times (2n - 1)$. Since v is the first vertex in both paths, it is placed at the bottom-left corner of the bounding box containing the drawing. Since w_1 and w_2 are the last vertices in $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, w_1 is placed on the right side, and w_2 on the top side of the bounding box. By construction, the bottom port of w_1 and the left port of w_2 are both unoccupied. Hence, the edges (v, w_1) and (v, w_2) that form $\mathcal{C}_1$ and $\mathcal{C}_2$ can

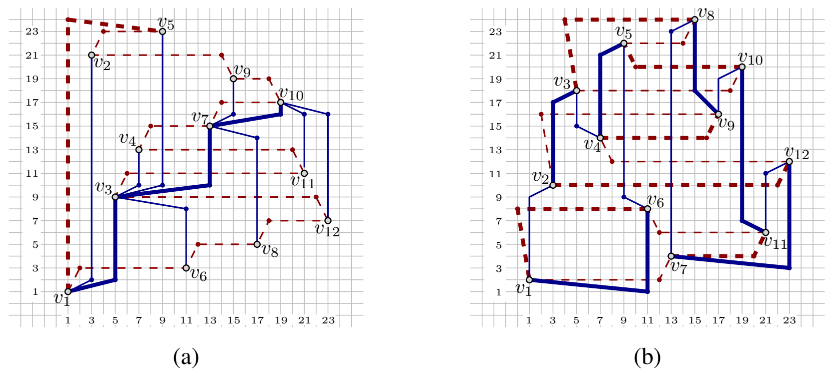


Fig. 3. RACSIM drawings with one bend per edge of: (a) a caterpillar $\mathcal{A}$ (solid; its spine is drawn bold) and a cycle $\mathcal{C}$ (dashed); (b) four matchings $\mathcal{M}_1$ (solid-plain), $\mathcal{M}_2$ (solid-bold), $\mathcal{M}_3$ (dashed-plain) and $\mathcal{M}_4$ (dashed-bold)

be drawn with a single bend at points $(2n-1, 0)$ and $(0, 2n-1)$, respectively; see Figure 2b. Clearly, none of them is involved in crossings, while the total area of the drawing gets larger by a single unit in each dimension. □

Theorem 4. *A caterpillar and a cycle on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $(2n-1) \times 2n$ with one bend per edge. The drawing can be computed in $O(n)$ time.*

Proof. We denote by $\mathcal{A} = (V, E_{\mathcal{A}})$ and $\mathcal{C} = (V, E_{\mathcal{C}})$ the caterpillar and the cycle, respectively. Let $v_1, v_2, \ldots, v_n$ be the vertex set of $\mathcal{A}$ ordered as follows (see Fig. 3a). Starting from an endpoint of the spine of $\mathcal{A}$, we traverse the spine such that we first visit all legs incident to a spine vertex before moving on to the next spine vertex. This order defines the x-order of the vertices in the output drawing.

As in the proof of Theorem 3, we temporarily delete an edge of $\mathcal{C}$ incident to v_1 (see the bold dashed edge in Fig. 3a) and obtain a path $\mathcal{P} = (V, E_{\mathcal{P}})$. For any vertex $v \in V$, let $\pi(v)$ be the position of v in $\mathcal{P}$. The map π determines the y-order of the vertices in our drawing. For $i = 1, 2, \ldots, n$, we draw v_i at point $(2i-1, 2\pi(v_i)-1)$. It remains to determine, for each edge $e = (v, v')$, where it bends. First, assume that $e \in E_{\mathcal{P}}$ and e is directed from its bottom to its top endpoint. Then, we place the bend at $v + (\operatorname{sgn}(x(v') - x(v)), 2)$. Second, assume that $e \in E_{\mathcal{A}}$ and e is directed from its left to its right endpoint. Then, we place the bend at $(x(v'), y(v) + \operatorname{sgn}(y(v') - y(v)))$.

The approach described above ensures that $\mathcal{P}$ is drawn y-monotone, hence planar. The spine of $\mathcal{A}$ is drawn x-monotone. The legs of a spine vertex of $\mathcal{A}$ are drawn to the right of their parent spine vertex and to the left of the next vertex along the spine. Hence, $\mathcal{A}$ is drawn planar as well. The slanted segments of $\mathcal{A}$ are of y-length 1, while the slanted segments of $\mathcal{P}$ are of x-length 1. Thus, they cannot be involved in crossings, which implies that all crossings form right angles.

It remains to draw the edge e in $E_{\mathcal{C}} \setminus E_{\mathcal{P}}$. Recall that e is incident to v_1, which lies at the bottom-left corner of the bounding box containing our drawing. Let v_j be the other endpoint of e. Since $\pi(v_j) = n$, vertex v_j lies at the top side of the bounding box. The top port of v_1 is not used, so we draw the first segment of e vertically, bending at $(1, 2n)$; see the bold dashed edge in Fig. 3a.

Clearly, the total area required by the drawing is $(2n-1) \times 2n$. $\square$

Theorem 5. *Four matchings on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $2n \times 2n$ with at most one bend per edge. The drawing can be computed in $O(n)$ time.*

Proof. Let $\mathcal{M}_1 = (V, E_1)$, $\mathcal{M}_2 = (V, E_2)$, $\mathcal{M}_3 = (V, E_3)$ and $\mathcal{M}_4 = (V, E_4)$ be the input matchings. W.l.o.g, we assume that all matchings are perfect; otherwise, we augment them to perfect matchings. Let $\mathcal{M}_{1,2} = (V, E_1 \cup E_2)$ and $\mathcal{M}_{3,4} = (V, E_3 \cup E_4)$. Since $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined on the same vertex set, $\mathcal{M}_{1,2}$ is a 2-regular graph. Thus, each connected component of $\mathcal{M}_{1,2}$ corresponds to a cycle of even length which alternates between edges of $\mathcal{M}_1$ and $\mathcal{M}_2$; see Fig. 3b. The same holds for $\mathcal{M}_{3,4}$. We will determine the x-coordinates of the vertices from $\mathcal{M}_{1,2}$, and the y-coordinates from $\mathcal{M}_{3,4}$.

We start with choosing an arbitrary vertex $v \in V$. Let $\mathcal{C}$ be the cycle of $\mathcal{M}_{1,2}$ containing vertex v. We determine the x-coordinates of the vertices of $\mathcal{C}$ by traversing $\mathcal{C}$ in some direction, starting from vertex v. For each vertex u in $\mathcal{C}$, let $\pi_1(u)$ be the discovery time of u according to this traversal, with $\pi_1(v) = 0$. Then, we set $x(u) = 2\pi_1(u) + 1$. Next, we determine the y-coordinates of the vertices of all cycles $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of $\mathcal{M}_{3,4}$ that have at least one vertex with a determined x-coordinate, ordered as follows. For $i = 1, \ldots, k$, let a_i be the *anchor* of $\mathcal{C}_i$, that is, the vertex with the smallest determined x-coordinate of all vertices in $\mathcal{C}_i$. Then, $x(a_1) < \ldots < x(a_k)$. In what follows, we start with the first cycle $\mathcal{C}_1$ of the computed order and determine the y-coordinates of its vertices. To do so, we traverse $\mathcal{C}_1$ in some direction, starting from its anchor vertex a_1. For each vertex u in $\mathcal{C}_1$, let $\pi_2(u)$ be the discovery time of u according to this traversal, with $\pi_2(a_1) = 0$. Then, we set $y(u) = 2\pi_2(u) + 1$. We proceed analogously with the remaining cycles $\mathcal{C}_i$, $i = 2, \ldots, k$, setting $\pi_2(a_i) = \max_{u \in \mathcal{C}_{i-1}} \pi_2(u) + 1$.

Now, there are no vertices with only one determined x-coordinate. However, there might exist vertices with only one determined y-coordinate. If this is the case, we repeat the aforementioned procedure to determine the x-coordinates of the vertices of all cycles of $\mathcal{M}_{1,2} \setminus \mathcal{C}$ that have at least one vertex with a determined y-coordinate, but without determined x-coordinates. If there are no vertices with only one determined coordinate left, either all coordinates are determined, or we restart this procedure with another arbitrary vertex that has no determined coordinates. Thus, our algorithm guarantees that the x- and y-coordinate of all vertices are eventually determined.

Note that, for each cycle in $\mathcal{M}_{1,2}$, there is exactly one edge $e = (v, v')$, called *closing edge*, with $\pi_1(v') > \pi_1(v) + 1$. Analogously, for each cycle in $\mathcal{M}_{3,4}$, there is exactly one closing edge $e = (u, u')$ with $\pi_2(u') > \pi_2(u) + 1$.

It remains to determine, for each edge $e = (v, v')$, where it bends. First, assume that $e \in E_1 \cup E_2$ and e is directed from its left to its right endpoint. If e is not a closing edge, we place the bend at $v' - (2, \operatorname{sgn}(y(v') - y(v))$. Otherwise, we place the bend at $(x(v'), y(v) - 1)$. Second, assume that $e \in E_3 \cup E_4$ and e is directed from its bottom to its top endpoint. If e is not a closing edge, we place the bend at $v' - (\operatorname{sgn}(x(v') - x(v)), 2)$. Otherwise, we place the bend at $(x(v) - 1, y(v'))$; see Fig. 3b.

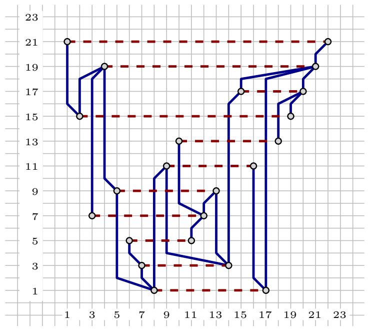


Fig. 4. A RACSIM drawing of a tree (solid) and a matching (dashed)

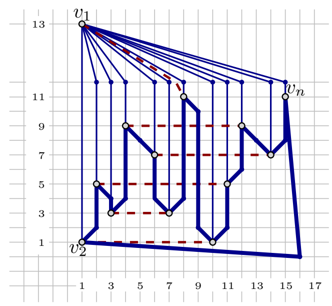


Fig. 5. A RACSIM drawing of a wheel (solid; its rim is drawn bold) and a matching (dashed)

Our choice of coordinates guarantees that the x-coordinates of the cycles of $\mathcal{M}_{1,2}$ and the y-coordinates of the cycles of $\mathcal{M}_{3,4}$ form disjoint intervals. Thus, the area below a cycle of $\mathcal{M}_{1,2}$ and the area to the left of a cycle of $\mathcal{M}_{3,4}$ are free from vertices. Hence, the slanted segments of the closing edges cannot have a crossing that violates the RAC restriction. Clearly, the total area required by the drawings is $2n \times 2n$. $\square$

Theorem 6. *A tree and a matching on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $n \times (n-1)$ with one bend per tree-edge, and no bends in the edges of the matching. The drawing can be computed in $O(n)$ time.*

Sketch of Proof. We inductively place each matching edge in one row. In every step, we decide whether to add the next matching edge to the stack at the top or at the bottom. We determine the x-coordinates of the matching with the help of a specific post-order visit; see Fig. 4. A detailed proof is given in the full version [4]. $\square$

4 RACSIM Drawings with Two Bends Per Edge

In this section, we study more complex classes of planar graphs, and show how to efficiently construct RACSIM drawings with two bends per edge in quadratic area. In particular, we prove that a wheel and a matching on a common set of n vertices admit a RACSIM drawing on an integer grid of size $(1.5n-1) \times (n+2)$ with two bends per edge and no bends, respectively; see Theorem 7. If the input to our problem is an outerpath—that is, an outerplanar graph whose weak dual is a path—and a matching, then a RACSIM drawing with two bends per edge and no bends, respectively, is also possible on an integer grid of size $(3n-2) \times (3n-2)$; see Theorem 8.

Theorem 7. *A wheel and a matching on a common set of n vertices admit a* RACSIM *drawing on an integer grid of size $(1.5n-1) \times (n+2)$ with two bends per edge and no bends, respectively. The drawing can be computed in $O(n)$ time.*

Proof. We denote the wheel by $\mathcal{W} = (V, E_{\mathcal{W}})$ and the matching by $\mathcal{M} = (V, E_{\mathcal{M}})$. A wheel can be decomposed into a cycle, called *rim*, a *center* vertex, and a set of edges that connect the center to the rim, called *spikes*. Let $V = \{v_1, v_2, \ldots, v_n\}$, such that v_1 is the center of $\mathcal{W}$ and $\mathcal{C} = \langle v_2, v_3, \ldots, v_n, v_2 \rangle$ is the rim of $\mathcal{W}$. Thus, $E_{\mathcal{W}} = \{(v_i, v_{i+1}) \mid i = 1, \ldots, n-1\} \cup \{(v_n, v_2)\} \cup \{(v_1, v_i) \mid i = 2, \ldots, n\}$. Let $\mathcal{M}' = (V, E_{\mathcal{M}'})$ be the matching $\mathcal{M}$ without the edge incident to v_1.

We first compute the x-coordinates of the vertices, such that $\mathcal{C} - \{(v_n, v_2)\}$ is x-monotone (if drawn with straight-line edges). More precisely, for $i = 2, \ldots, n$ we set $x(v_i) = 2i - 3$. The y-coordinates of the vertices are computed based on the matching $\mathcal{M}'$, as follows. Let $E_{\mathcal{M}'} = \{e_1, \ldots, e_k\}$ be the matching edges with v_2 incident to e_1. For $i = 1, \ldots, k$, we assign the y-coordinate $2i - 1$ to the endpoints of e_i. Next, we assign the y-coordinate $2k + 1$ to the vertices incident to the rim without a matching edge in $\mathcal{M}'$. Finally, the center v_1 of $\mathcal{W}$ is located at point $(1, 2k + 3)$.

It remains to determine, for each edge $e \in E_{\mathcal{W}}$, where it bends, as $\mathcal{M}'$ is drawn bendless. First, let $e = (v_1, v_i)$, $i = 3, \ldots, n$ be a spike. Then, we place the bend at $(x(v_j), 2k + 2)$. Since both v_1 and v_2 are located in column 1, we can save the bend of the spike (v_1, v_2). Second, let $e = (v_i, v_{i+1})$, $i = 2, \ldots, n-1$ be an edge of the rim $\mathcal{C}$. If $y(v_{i+1}) > y(v_i)$, we place the bend at $(x(v_{i+1}), y(v_i) + 1)$. If $y(v_{i-1}) > y(v_i) > y(v_{i+1})$, we place the bend at $(x(v_{i+1}), y(v_i) - 1)$. If $y(v_i) > y(v_{i-1}), y(v_{i+1})$, the bottom port at v_i is already used. Thus, we draw the edge with two bends at $(x(v_{i+1}), y(v_i) - 1)$ and $(x(v_{i+1}), y(v_{i+1}) + 1)$. Finally, let $e = (v_n, v_2)$ be the remaining edge of the rim. Then, we place the bend at $(2n - 2, 0)$. See Fig. 5 for an illustration.

Our approach ensures that $\mathcal{C} - \{(v_n, v_2)\}$ is drawn x-monotone, hence planar. The last edge (v_n, v_2) of $\mathcal{C}$ outside of the bounding box containing the vertices; thus, it is crossing-free. Further, the spikes are not involved in crossings with the rim, as they are outside of the bounding box containing the rim edges. Hence, $\mathcal{W}$ is drawn planar. On the other hand, all edges of $\mathcal{M}'$ are drawn as horizontal, non-overlapping line segments. Thus, $\mathcal{M}'$ is drawn planar as well. The slanted segments of $\mathcal{W} - (v_n, v_2)$ are of y-length 1. So, they cannot be crossed by the edges of $\mathcal{M}'$. As the edge (v_n, v_2) is not involved in crossings, it follows that all crossings between $\mathcal{W}$ and $\mathcal{M}'$ form right angles.

Finally, we have to insert the matching edge (v_1, v_i) in $E_{\mathcal{M}} \setminus E'_{\mathcal{M}}$. Since v_i is not incident to a matching edge in $\mathcal{M}'$, it is placed above all matching edges. Then, $(v_1, v_i) \in \mathcal{W}$ does not cross a matching edge, so we can use this edge as a double edge.

We will now prove the area bound of the drawing algorithm. To that end, we remove all columns that contain neither a vertex, nor a bend. First, we count the rows used. Since we remove the matching edge incident to v_1, the matching $\mathcal{M}'$ has $k \leq n/2 - 1$ matching edges. We place the bottommost vertex in row 1 and the topmost vertex, that is, vertex v_1, in row $2k+3$. We add one extra bend in row 0 for the edge (v_n, v_2). Thus, our drawing uses $2k + 3 + 1 \leq n + 2$ rows. Next, we count the columns used. The vertices $v_2, \ldots, v_n$ are each placed in their own column. Every spike has exactly one bend in the column of a vertex. An edge (v_i, v_{i+1}) of rim $\mathcal{W}$ has exactly one bend in a vertex column, except for the case that $y(v_i) > y(v_{i-1}), y(v_{i+1})$, in which it needs an extra bend between v_i and v_{i+1}, $i = 1, \ldots, n-1$. Clearly, there can be at most $n/2 - 1$ vertices satisfying this condition. Since the edge (v_n, v_2) uses an extra column to the right of v_n, our drawing uses $(n - 1) + (n/2 - 1) + 1 = 1.5n - 1$ columns. $\square$

Theorem 8. *An outerpath and a matching on a common set of n vertices admit a* RAC-SIM *drawing on an integer grid of size $(3n-2) \times (3n-2)$ with two bends per edge and one bend, respectively. The drawing can be computed in $O(n)$ time.*

Sketch of Proof. We augment the outerpath to a maximal outerpath. Removing its outer cycle, the result is a caterpillar, which determines the x-coordinates of the vertices as outlined in Thm. 4. Then, the y-coordinates are computed similar to Thm. 7 such that the matching is planar. A detailed proof is given in the full version [4]. $\square$

5 Conclusions and Open Problems

The results presented in this paper raise several questions that remain open.

1. Is it possible to reduce the number of bends per edge for the classes of graphs that we presented in this paper? What additional non-trivial classes of graphs admit a RACSIM drawing with better-than-general number of bends?
2. As a variant of the problem, it may be possible to reduce the required number of bends per edge by relaxing the strict constraint that intersections must be right-angled and instead ask for drawings that have close-to-optimal crossing resolution.
3. What is the computational complexity of the general problem: Given two or more planar graphs on the same set of vertices and an integer k, is there a RACSIM drawing in which each graph is drawn with at most k bends per edge, and the crossings are at right angles?

References

1. Angelini, P., Battista, G.D., Frati, F., Patrignani, M., Rutter, I.: Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. J. Discrete Algorithms 14, 150–172 (2012)
2. Angelini, P., Geyer, M., Kaufmann, M., Neuwirth, D.: On a tree and a path with no geometric simultaneous embedding. J. Graph Algorithms Appl. 16(1), 37–83 (2012)
3. Argyriou, E.N., Bekos, M.A., Kaufmann, M., Symvonis, A.: Geometric RAC simultaneous drawings of graphs. J. Graph Algorithms Appl. 17(1), 11–34 (2013)
4. Bekos, M.A., van Dijk, T.C., Kindermann, P., Wolff, A.: Simultaneous drawing of planar graphs with right-angle crossings and few bends. Arxiv report (2014), available at http://arxiv.org/abs/1408.3325
5. Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: Tamassia, R. (ed.) Handbook of Graph Drawing and Visualization, chap. 11, pp. 349–381. CRC Press (2013)
6. Brass, P., Cenek, E., Duncan, C.A., Efrat, A., Erten, C., Ismailescu, D.P., Kobourov, S.G., Lubiw, A., Mitchell, J.S.: On simultaneous planar graph embeddings. Comput. Geom. Theory Appl. 36(2), 117–130 (2007)
7. Brightwell, G., Scheinerman, E.R.: Representations of planar graphs. SIAM J. Discrete Math. 6(2), 214–229 (1993)
8. Cabello, S., van Kreveld, M., Liotta, G., Meijer, H., Speckmann, B., Verbeek, K.: Geometric simultaneous embeddings of a graph and a matching. J. Graph Algorithms Appl. 15(1), 79–96 (2011)

9. Erten, C., Kobourov, S.G.: Simultaneous embedding of a planar graph and its dual on the grid. Theory Comput. Syst. 38(3), 313–327 (2005)
10. Erten, C., Kobourov, S.G.: Simultaneous embedding of planar graphs with few bends. J. Graph Algorithms Appl. 9(3), 347–364 (2005)
11. Kammer, F.: Simultaneous embedding with two bends per edge in polynomial area. In: Arge, L., Freivalds, R. (eds.) SWAT 2006, LNCS, vol. 4059, pp. 255–267. Springer, Heidelberg (2006)
12. Kaufmann, M., Wiese, R.: Embedding vertices at points: Few bends suffice for planar graphs. J. Graph Algorithms Appl. 6(1), 115–129 (2002)
13. Nash-Williams, C.St.J.A.: Decomposition of finite graphs into forests. J. London Math. Soc. 39, 12 (1964)