Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 17, no. 1, pp. 11–34 (2013)
DOI: 10.7155/jgaa.00282

Geometric RAC Simultaneous Drawings of Graphs

E. N. Argyriou[1] *M. A. Bekos*[2] *M. Kaufmann*[2] *A. Symvonis*[1]

[1]School of Applied Mathematics & Physical Sciences,
National Technical University of Athens, Greece
[2]Institute for Informatic,
University of Tübingen, Germany

Abstract

In this paper, we study the *geometric RAC simultaneous drawing problem*: Given two planar graphs that share a common vertex set, a geometric RAC simultaneous drawing is a straight-line drawing in which each graph is drawn planar, there are no edge overlaps, and, crossings between edges of the two graphs occur at right angles. We first prove that two planar graphs admitting a geometric simultaneous drawing may not admit a geometric RAC simultaneous drawing. We further show that a cycle and a matching always admit a geometric RAC simultaneous drawing.We also study a closely related problem according to which we are given a planar embedded graph G and the main goal is to determine a geometric drawing of G and its weak dual G^* such that: (i) G and G^* are drawn planar, (ii) each vertex of the dual is drawn inside its corresponding face of G and, (iii) the primal-dual edge crossings form right angles. We prove that it is always possible to construct such a drawing if the input graph is an outerplanar embedded graph.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
May 2012	October 2012	November 2012	December 2012	December 2012

Published:
January 2013

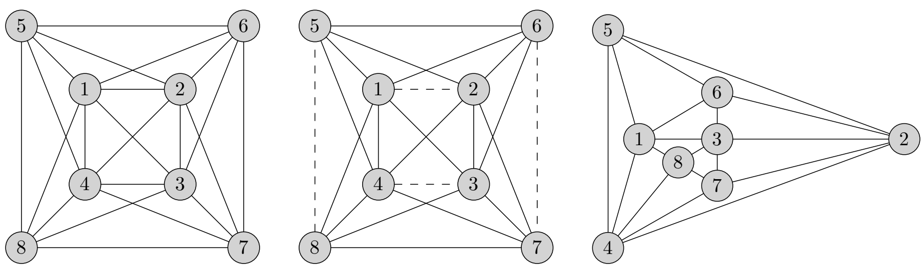
Article type:	Communicated by:
Regular paper	G. Liotta

E-mail addresses: fargyriou@math.ntua.gr (E. N. Argyriou) bekos@informatik.uni-tuebingen.gr (M. A. Bekos) mk@informatik.uni-tuebingen.gr (M. Kaufmann) symvonis@math.ntua.gr (A. Symvonis)

1 Introduction

A *geometric right angle crossing drawing* (or *geometric RAC drawing*, for short) of a graph is a straight-line drawing in which every pair of crossing edges intersects at right angle. A graph which admits a geometric RAC drawing is called *right angle crossing graph* (or *RAC graph*, for short). Motivated by cognitive experiments of Huang et al. [17], which indicate that the negative impact of an edge crossing on the human understanding of a graph drawing is eliminated in the case where the crossing angle is greater than seventy degrees, RAC graphs were recently introduced in [10] as a response to the problem of drawing graphs with optimal crossing angle resolution.

Simultaneous graph drawing deals with the problem of drawing two (or more) planar graphs on the same set of vertices on the plane, such that each graph is drawn planar[1] (i.e., only edges of different graphs are allowed to cross). The *geometric* version restricts the problem to straight-line drawings. Besides its independent theoretical interest, this problem arises in several application areas, such as software engineering, databases and social networks, where a visual analysis of evolving graphs, defined on the same set of vertices, is useful.



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

Figure 1: (a) A graph with 8 vertices and 22 edges which does not admit a RAC drawing [11]. (b) A decomposition of the graph of Figure 1a into a planar graph (solid edges; a planar drawing is given in Figure 1c) and a matching (dashed edges), which implies that a planar graph and a matching do not always admit a GRacSim drawing; their union is not RAC.

Both problems mentioned above are active research topics in the graph drawing literature and positive and negative results are known for certain variations (refer to Section 2). In this paper, we study the *geometric RAC simultaneous drawing problem* (or *GRacSim* drawing problem, for short), i.e., a combination of both problems. Formally, the GRacSim drawing problem can be stated as follows: Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two planar graphs that share a common vertex set. The main task is to place the vertices on the plane so that, when the edges are drawn as straight-lines segments, (i) each graph is drawn planar, (ii) there are no edge overlaps and (iii) crossings between edges in E_1

[1] In the graph drawing literature, the problem is known as "simultaneous graph drawing with mapping". For simplicity, we use the term "simultaneous graph drawing".

and E_2 occur at right angles. Let $G = (V, E_1 \cup E_2)$ be the graph induced by the union of G_1 and G_2. Observe that G should be a RAC graph, which implies that $|E_1 \cup E_2| \leq 4|V| - 10$ [10]. We refer to this relationship as the *RAC-size constraint*.

If two graphs do not admit a geometric simultaneous drawing they, obviously, do not admit a GRacSim drawing. For instance, since it is known that there exists a planar graph and a matching that do not admit a geometric simultaneous drawing [7], as a consequence, the same graph and matching do not admit a GRacSim drawing either. Figure 1 depicts an alternative and simpler technique to prove such negative results for GRacSim drawings, which is based on the fact that not all graphs that obey the RAC-size constraint are actually RAC graphs. On the other hand, as we will shortly see, two planar graphs admitting a geometric simultaneous drawing may not admit a GRacSim drawing.

The GRacSim drawing problem is of interest since it combines two current research topics in graph drawing. Our motivation to study this problem rests on the work of Didimo et al. [10] who proved that the crossing graph of a geometric RAC drawing is bipartite[2]. Thus, the edges of a geometric RAC drawing of a graph $G = (V, E)$ can be partitioned into two sets E_1 and E_2, such that no two edges of the same set cross. So, the problem we study is, in a sense, equivalent to the problem of finding a geometric RAC drawing of an input graph (if one exists), given its crossing graph.

A closely related problem to the GRacSim drawing problem is the following: *Given a planar embedded graph G, determine a geometric drawing of G and its weak dual G^* (i.e., without the face-vertex corresponding to the external face) such that: (i) G and G^* are drawn planar, (ii) each vertex of the dual is drawn inside its corresponding face of G and, (iii) the primal-dual edge crossings form right angles.* We refer to this problem as the *geometric Graph-Dual RAC simultaneous drawing problem* (or *GDual-GRacSim* for short). Note that, the GDual-GRacSim drawing problem is not a new problem. Back in 1963, W.T. Tutte asked whether "Can we construct simultaneous straight representations ... of G and G^* in which the ... corresponding edges are represented by perpendicular segments ?" [18, p.p.767].

The remainder of this paper is structured as follows: In Section 2, we review relevant previous research. In Section 3, we demonstrate that two planar graphs admitting a geometric simultaneous drawing may not admit a GRacSim drawing. In Section 4, we prove that a cycle and a matching always admit a GRacSim drawing, which can be constructed in linear time. In Section 5, we study the GDual-GRacSim drawing problem. We show that given a planar embedded graph, a GDual-GRacSim drawing of the planar graph and its weak dual does not always exist. If the input graph is an outerplanar embedded graph, we present an algorithm that constructs a GDual-GRacSim drawing of the outerplanar graph and its weak dual. We conclude in Section 6 with open problems.

[2]This can be interpreted as follows: "If two edges of a geometric RAC drawing cross a third one, then these two edges must be parallel."

2 Related Work

Didimo et al. [10] were the first to study the geometric RAC drawing problem and proved that any graph with $n \geq 3$ vertices that admits a geometric RAC drawing has at most $4n - 10$ edges. Arikushi et al. [4] presented bounds on the number of edges of polyline RAC drawings with at most one or two bends per edge. Angelini et al. [1] presented acyclic planar digraphs that do not admit upward geometric RAC drawings and proved that the corresponding decision problem is $\mathcal{NP}$-hard. Argyriou et al. [3] proved that it is $\mathcal{NP}$-hard to decide whether a given graph admits a geometric RAC drawing (i.e., the upwardness requirement is relaxed). Di Giacomo et al. [8] presented tradeoffs on the maximum number of bends per edge, the required area and the crossing angle resolution. Didimo et al. [9] characterized classes of complete bipartite graphs that admit geometric RAC drawings. Van Kreveld [19] showed that the quality of a planar drawing of a planar graph (measured in terms of area required, edge-length and angular resolution) can be improved if one allows right angle crossings. Eades and Liotta [11] proved that a *maximally dense RAC graph* (i.e., $|E| = 4|V| - 10$) is also 1-planar, i.e., it admits a drawing in which every edge is crossed at most once.

Regarding the geometric simultaneous graph drawing problem, Brass et al. [5] presented algorithms for drawing simultaneously (a) two paths, (b) two cycles and, (c) two caterpillars. They also proved that there exist three paths that do not admit a geometric simultaneous drawing. Estrella-Balderrama et al. [14] proved that the problem of determining whether two planar graphs admit a geometric simultaneous drawing is $\mathcal{NP}$-hard. Erten and Kobourov [13] showed that a planar graph and a path cannot always be drawn simultaneously. Frati, Kaufmann and Kobourov [15] proved this negative result, even for the case where the planar graph and the path do not share any edges. Geyer, Kaufmann and Vrt'o [16], showed that a geometric simultaneous drawing of two trees does not always exist. Angelini et al. [2] proved the same result for a path and a tree. Cabello et al. [7] showed that a geometric simultaneous drawing of a matching and (a) a wheel, (b) an outerpath or, (c) a tree always exists, while there exist a planar graph and a matching that cannot be drawn simultaneously. For a quick overview of known results on this research area refer to Table 1 of [15].

Brightwell and Scheinermann [6] proved that the GDual-GRacSim drawing problem always admits a solution if the input graph is a triconnected planar graph. To the best of our knowledge, this is the only result which incorporates the requirement that the primal-dual edge crossings form right angles. Erten and Kobourov [12], presented an $O(n)$ time algorithm that results into a simultaneous drawing but not a RAC drawing of a triconnected planar graph and its dual on an $O(n^2)$ integer grid, where n is the total number of vertices in the graph and its dual.

Before we proceed with the description of our results, we introduce some necessary notation. Let $G = (V, E)$ be a simple, undirected graph drawn on the plane. We denote by $\Gamma(G)$ the drawing of G. By $x(v)$ and $y(v)$, we denote the x- and y-coordinate of $v \in V$ in $\Gamma(G)$. We refer to the vertex (edge) set of G

as $V(G)$ ($E(G)$). Given two graphs G and G', we denote by $G \cup G'$ the graph induced by the union of G and G'.

3 A Wheel and a Matching: A Negative Result

In this section, we show that there exists a pair of planar graphs that admits a geometric simultaneous drawing, their union meets the RAC size constraint and they do not admit a GRacSim drawing (i.e, the class of graphs that admit GRacSim drawings is a subset of the class of graphs for which a simultaneous drawing is possible). We achieve this by showing that there exists a wheel and a matching which do not admit a GRacSim drawing[3]. Cabello et al. [7] have shown that a geometric simultaneous drawing of a wheel and a matching always exists. Before we proceed with the detailed description of our proof, we first present a known property of RAC graphs, which has been independently observed by Didimo, Eades and Liotta [10] and Angelini et al. [1].

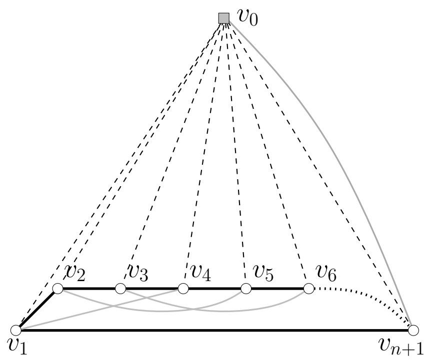
Rac-Property 1 *Let (u, v) and (u, v') be a pair of non-overlapping edges incident to the same vertex. We say that (u, v) and (u, v') form a* fan anchored *at u. In a straight-line RAC drawing no edge can cross a fan.*

Theorem 1 *There exists a wheel and a matching which do not admit a* GRacSim *drawing.*

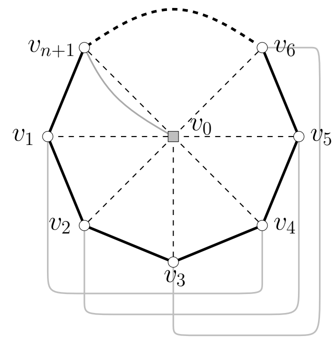
Proof: We denote the wheel by $\mathcal{W}$ and the matching by $\mathcal{M}$. Let the common vertex set of $\mathcal{W}$ and $\mathcal{M}$ be $V = \{v_0, v_1, \ldots, v_{n+1}\}$, where $n \geq 6$. We further assume that $n = 6k$, for some $k \in \mathbb{Z}$. If v_0 is the center of $\mathcal{W}$ and $v_1 \to \ldots \to v_{n+1} \to v_1$ is the rim of $\mathcal{W}$, then $E(\mathcal{W}) = \{(v_i, v_{i+1});\ i = 1, \ldots, n\} \cup \{(v_{n+1}, v_1)\} \cup \{(v_0, v_i);\ i = 1, \ldots, n+1\}$. Matching $\mathcal{M}$ contributes $n/2+1$ edges; one edge of $\mathcal{M}$ connects v_0 with v_{n+1}. The $6k$ vertices on the rim (excluding vertex $6k+1$) are split into k groups, with group i, $1 \leq i \leq k$, consisting of vertices $v_{6i}, \ldots, v_{6i+5}$. Then, in each group i, vertex v_{6i+j} is matched with v_{6i+j+3}, $j = 0, 1, 2$. More formally, $E(\mathcal{M}) = \{(v_{6i-j-3}, v_{6i-j});\ i = 1, \ldots, n/6,\ j = 0, 1, 2\} \cup \{(v_0, v_{n+1})\}$.

In any planar drawing of $\mathcal{W}$, the outerface is bounded either by a 3-cycle formed by v_0 and two consecutive vertices of the rim of $\mathcal{W}$ (see Figure 2a where we assume without loss of generality that (v_1, v_{n+1}) is an edge of the boundary) or by the rim of $\mathcal{W}$ itself (see Figure 2b). Now observe that in both cases each edge of $\mathcal{M}$ (except for edge (v_0, v_{n+1})) connects two vertices that belong to two triangles of $\mathcal{W}$ incident at v_0 and are not consecutive around v_0, in the planar drawing of $\mathcal{W}$. Hence, by RAC-Property 1 it follows that the edges of $\mathcal{M}$ cannot cross the spokes of $\mathcal{W}$. This implies that the edges of $\mathcal{M}$ and the edges of $\mathcal{W}$ cannot cross with each other, except for edge (v_0, v_{n+1}) that belongs to both $\mathcal{M}$ and $\mathcal{W}$. However, under this assumption $\mathcal{M}$ cannot be drawn planar. □

[3] A wheel and a matching on a vertex set of size n contribute $5n/2 - 2$ edges, which meets the RAC size constraint.



(a) The outerface of $\mathcal{W}$ is bounded by a 3-cycle



(b) The outerface of $\mathcal{W}$ is bounded by its rim

Figure 2: In both figures, the center of $\mathcal{W}$ is marked by a box, the spokes of $\mathcal{W}$ are drawn as dashed line-segments, the rim of $\mathcal{W}$ is drawn in bold, while matching $\mathcal{M}$ is drawn in gray. The examples correspond to the case where $k = 1$, i.e., wheel of rim-size seven. Wheels of rim-size $6k + 1$, $k > 1$ can be obtained by subdividing the dotted drawn edges and appropriately inserting wheel and matching edges.

4 A Cycle and a Matching: A Positive Result

In this section, we first prove that a path and a matching always admit a GRacSim drawing and then we show that a cycle and a matching always admit a GRacSim drawing as well. Note that, the union of a path and a matching is not necessarily a planar graph. Cabello et al. [7] provide an example of a path and a matching, which form a subdivision of $K_{3,3}$. We denote the path by $\mathcal{P}$ and the matching by $\mathcal{M}$. Let $v_1 \to v_2 \to \ldots \to v_n$ be the edges of $\mathcal{P}$ (see Figure 3). In order to keep the description of our algorithm simple, we will initially assume that $\mathcal{P}$ and $\mathcal{M}$ do not share edges, i.e., $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$. Since $\mathcal{P}$ and $\mathcal{M}$ are defined on the same vertex set, n should be even and $|E(\mathcal{M})| = n/2$ (i.e., $\mathcal{M}$ is a perfect matching). Later on this section, we will describe how to cope with the cases where $E(\mathcal{P}) \cap E(\mathcal{M}) \neq \emptyset$ or $\mathcal{M}$ is not a perfect matching.

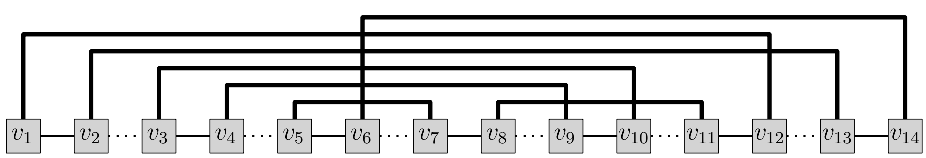


Figure 3: An example of a path $\mathcal{P}$ and a matching $\mathcal{M}$. The path appears at the bottom of the figure. The edges of $\mathcal{M}$ are drawn bold, with two bends each. The edges of path $\mathcal{P}$ form two matchings, i.e., $\mathcal{P}_{odd}$ and $\mathcal{P} - \mathcal{P}_{odd}$. The edges of $\mathcal{P}_{odd}$ are drawn solid, while the edges of $\mathcal{P} - \mathcal{P}_{odd}$ dotted.

The basic idea of our algorithm is to identify in the graph induced by the union of $\mathcal{P}$ and $\mathcal{M}$ a set of cycles $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, $k \leq n/4$, such that: (i) $|E(\mathcal{C}_1)| + |E(\mathcal{C}_2)| + \ldots + |E(\mathcal{C}_k)| = n$, (ii) $\mathcal{M} \subseteq \mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_k$, and, (iii) the edges of cycle $\mathcal{C}_i$, $i = 1, 2, \ldots, k$ alternate between edges of $\mathcal{P}$ and $\mathcal{M}$. Note that properties (i) and (ii) imply that the cycle collection will contain half of $\mathcal{P}$'s edges (in particular $\lceil |E(P)|/2 \rceil$ edges) and all of $\mathcal{M}$'s edges. In our drawing, these edges will not cross with each other. The remaining edges of $\mathcal{P}$ will introduce only right angle crossings with the edges of $\mathcal{M}$.

Let $\mathcal{P}_{odd}$ be a subgraph of $\mathcal{P}$ which contains each second edge of $\mathcal{P}$, starting from its first edge, i.e., $E(\mathcal{P}_{odd}) = \{(v_i, v_{i+1});\ 1 \leq i < n,\ i \text{ is odd}\}$. In Figure 3, the edges of $\mathcal{P}_{odd}$ are drawn solid. Clearly, $\mathcal{P}_{odd}$ is a matching. Since we have assumed that n is even, $\mathcal{P}_{odd}$ contains exactly $n/2$ edges. Hence, $|E(\mathcal{P}_{odd})| = |E(\mathcal{M})|$. In addition, $\mathcal{P}_{odd}$ covers all vertices of $\mathcal{P}$, and, $E(\mathcal{P}_{odd}) \cap E(\mathcal{M}) = \emptyset$. The later equation trivially follows from our initial hypothesis, which states that $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$. We conclude that $\mathcal{P}_{odd} \cup \mathcal{M}$ is a 2-regular graph. Thus, each connected component of $\mathcal{P}_{odd} \cup \mathcal{M}$ corresponds to a cycle of even length, which alternates between edges of $\mathcal{P}_{odd}$ and $\mathcal{M}$. This is the cycle collection mentioned above (see Figure 4).

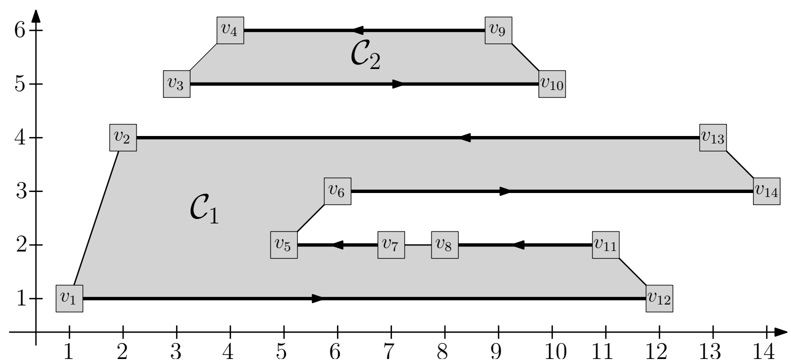


Figure 4: $\mathcal{P}_{odd} \cup \mathcal{M}$ (of Figure 3) consists of cycles $\mathcal{C}_1$ and $\mathcal{C}_2$. The edges of $\mathcal{P}_{odd}$ are drawn solid, while the edges of $\mathcal{M}$ are drawn bold.

Initially, we fix the x-coordinate of each vertex[4] of $\mathcal{P}$ by setting $x(v_i) = i$, $1 \leq i \leq n$. This ensures that $\mathcal{P}$ is x-monotone and hence planar. Later on, we will slightly change the x-coordinate of some vertices of $\mathcal{P}$ without affecting $\mathcal{P}$'s monotonicity. The y-coordinate of each vertex of $\mathcal{P}$ is established by considering the cycles of $\mathcal{P}_{odd} \cup \mathcal{M}$.

We draw each of these cycles in turn. More precisely, assume that zero or more cycles have been completely drawn and let $\mathcal{C}$ be the cycle in the cycle collection which contains the leftmost vertex, say v_i, of $\mathcal{P}$ that has not been drawn yet (initially, v_i is identified by v_1). Then, vertex v_i should be an odd-

[4]Note that, the algorithm can be adjusted so that the x and y coordinates of each vertex are computed at the same time (without affecting neither the correctness of the algorithm nor its running time). We have chosen to compute them separately in order to simplify the presentation.

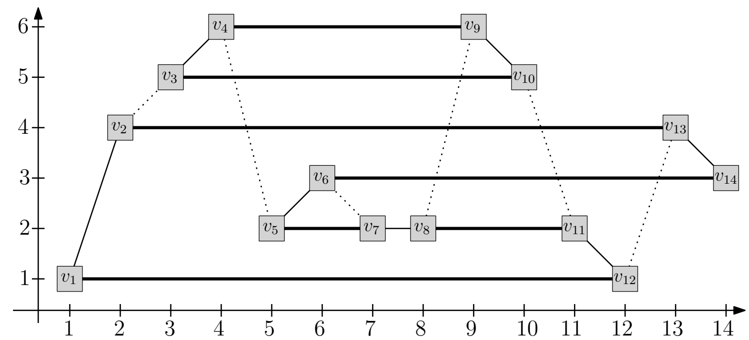
indexed vertex and thus (v_i, v_{i+1}) belongs to $\mathcal{C}$. Orient cycle $\mathcal{C}$ so that vertex v_i is the first vertex of cycle $\mathcal{C}$ and v_{i+1} is the last (see Figure 4). Based on this orientation, we will draw the edges of $\mathcal{C}$ in a snake-like fashion, starting from vertex v_i and reaching vertex v_{i+1} last. The first edge to be drawn is incident to vertex v_i and belongs to $\mathcal{M}$. We draw it as a horizontal line-segment at the bottommost available layer in the produced drawing (initially, $L_1 : y = 1$). Since cycle $\mathcal{C}$ alternates between edges of $\mathcal{P}_{odd}$ and $\mathcal{M}$, the next edge to be drawn belongs to $\mathcal{P}_{odd}$ followed by an edge of $\mathcal{M}$. If we can draw both of them in the current layer without introducing edge overlaps, we do so. Otherwise, we employ an additional layer, i.e., the edge of $\mathcal{P}_{odd}$ is drawn oblique starting from the current layer towards the next one and the edge of $\mathcal{M}$ is drawn horizontally at the new layer. We continue in the same manner, until edge (v_i, v_{i+1}) is reached in the traversal of cycle $\mathcal{C}$. This edge connects two consecutive vertices of $\mathcal{P}$ that are the leftmost in the drawing of $\mathcal{C}$. Therefore, edge (v_i, v_{i+1}) can be added in the drawing of $\mathcal{C}$ without introducing any crossings. Thus, cycle $\mathcal{C}$ is drawn planar.

So far, we have drawn all edges of $\mathcal{M}$ and half of the edges of $\mathcal{P}$ (i.e., $\mathcal{P}_{odd}$) and we have obtained a planar drawing in which all edges of $\mathcal{M}$ are drawn as horizontal, non-overlapping line segments. In the worst case, this drawing occupies $n/2$ layers.

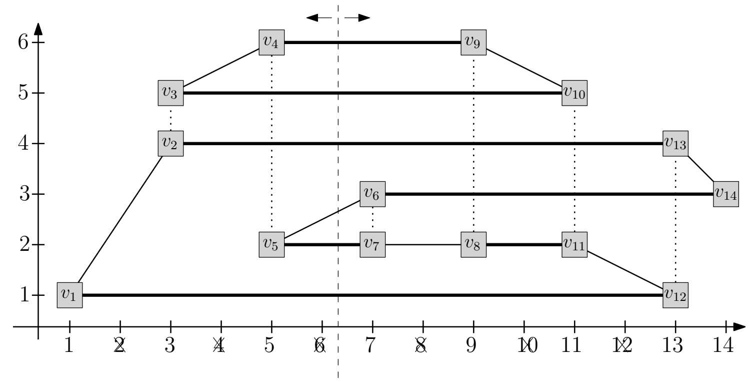
We proceed to incorporate the remaining edges of $\mathcal{P}$, i.e, the ones that belong to $\mathcal{P} - \mathcal{P}_{odd}$, into the drawing (refer to the dotted drawn edges of Figure 5a). Since $x(v_i) = i$, $i = 1, 2, \ldots, n$, the edges of $\mathcal{P}$ do not cross with each other and therefore $\mathcal{P}$ is drawn planar. In contrast, an edge of $\mathcal{P} - \mathcal{P}_{odd}$ may cross multiple edges of $\mathcal{M}$, and, these crossings do not form right angles (see Figure 5a). In order to fix these crossings, we move each even-indexed vertex of $\mathcal{P}$ one unit to the right (keeping its y-coordinate unchanged), except for the last vertex of $\mathcal{P}$. Then, the endpoints of the edges of $\mathcal{P} - \mathcal{P}_{odd}$ have exactly the same x-coordinate and cross at right angles the edges of $\mathcal{M}$ which are drawn as horizontal line-segments. The path remains x-monotone (but not strictly anymore) and hence planar. In addition, it is not possible to introduce vertex overlaps, since in the produced drawing each edge of $\mathcal{M}$ has at least two units length (recall that $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$). Since the vertices of the drawing do not occupy even x-coordinates, the width of the drawing can be reduced from n to $n/2 + 1$ (see Figure 5b). We can further reduce the width of the produced drawing by merging consecutive columns that do not interfere in y-direction into a common column (see Figure 5c). However, this post-processing does not result into a drawing of asymptotically smaller area. This completes the description of our algorithm for the case where $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$.

Theorem 2 *A path $\mathcal{P}$ and a perfect matching $\mathcal{M}$ on the same vertex set and such that $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$ always admit a* GRacSim *drawing on an $(n/2+1) \times n/2$ integer grid, where n is the size of the vertex set. Moreover, the drawing can be computed in linear time.*

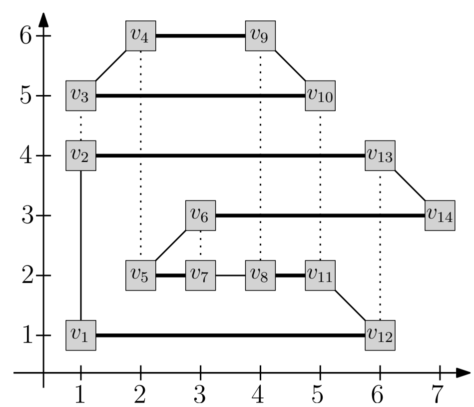
Proof: Finding the cycles of $\mathcal{P}_{odd} \cup \mathcal{M}$ can be done in $O(n)$ time, where n is the number of vertices of $\mathcal{P}$; we identify the leftmost vertex of each cycle and



(a) A drawing obtained by incorporating the edges of $\mathcal{P} - \mathcal{P}_{odd}$ into the drawing of Figure 4.



(b) A drawing obtained by moving the even-indexed vertices of $\mathcal{P}$ in the drawing of Figure 5a one unit to the right.



(c) A compact GRacSim drawing.

Figure 5: In the drawings the edges of $\mathcal{P}_{odd}$ are drawn solid, while the edges of $\mathcal{P} - \mathcal{P}_{odd}$ dotted. The edges of $\mathcal{M}$ are drawn bold.

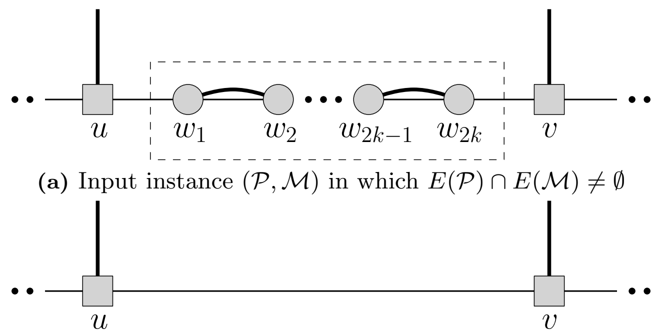


Figure 6: In the drawings the edges of $\mathcal{P}$ are drawn plain, while the edges of $\mathcal{M}$ are drawn bold. Vertices of $V_{dis}(\mathcal{P})$ ($V_{com}(\mathcal{P})$, resp.) are drawn as squares (circles, resp.)

then we traverse it. Having computed the cycle collection of $\mathcal{P}_{odd} \cup \mathcal{M}$, the coordinates of the vertices are computed in $O(n)$ total time by a traversal of the cycle. □

For the case where $\mathcal{P}$ and $\mathcal{M}$ share edges (i.e., $E(\mathcal{P}) \cap E(\mathcal{M}) \neq \emptyset$), our intention is to construct a drawing on an $n \times n/2$ integer grid by extending the algorithm that supports Theorem 2. To achieve this, we utilize the fact that the resulting drawings for the case where $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$ are *stretchable* in the following sense: If we draw any vertical non-grid line that crosses part of the drawing (refer to the dashed drawn line of Figure 5b), then we can shift to the left (right, resp.) the drawing that is to the left (right, resp.) of this line without affecting either the planarity of $\mathcal{P}$ and $\mathcal{M}$ or the angles in which $\mathcal{P}$ and $\mathcal{M}$ cross (since crossings always appear at grid points; horizontal stretching). Similarly, one can vertically stretch the drawing by employing a horizontal non-grid line.

We initially assume that the first and the last edge of $\mathcal{P}$ do not appear in $\mathcal{M}$[5], i.e., edges that are in both $\mathcal{P}$ and $\mathcal{M}$ are interior edges of $\mathcal{P}$. Let $V_{com}(\mathcal{P})$ ($V_{dis}(\mathcal{P})$, resp.) be the set of vertices of $\mathcal{P}$ which are (are not, resp.) incident to an edge that belongs to both $\mathcal{P}$ and $\mathcal{M}$ (see Figure 6). More formally, $V_{com}(\mathcal{P}) = \{u \in V(\mathcal{P}) : \exists u \in V(\mathcal{P})\ s.t.\ (u,v) \in E(\mathcal{P}) \cap E(\mathcal{M})\}$ and $V_{dis}(\mathcal{P}) = V(\mathcal{P}) - V_{com}(\mathcal{P})$. Since we have assumed that neither the first nor the last edge of $\mathcal{P}$ appear in $\mathcal{M}$, $V_{com}(\mathcal{P}) \subset V(\mathcal{P})$. Similarly, we define $V_{com}(\mathcal{M})$ and $V_{dis}(\mathcal{M})$. Obviously, $V_{com}(\mathcal{P}) = V_{com}(\mathcal{M})$. Since $\mathcal{P}$ and $\mathcal{M}$ are defined on the same vertex set, it follows that $V_{dis}(\mathcal{P}) = V_{dis}(\mathcal{M})$.

If there exist edges that belong to both $\mathcal{P}$ and $\mathcal{M}$, we momentarily remove them from both $\mathcal{P}$ and $\mathcal{M}$ as follows: If $u \to w_1 \to \ldots \to w_{2k} \to v$ is a subpath of vertices of $\mathcal{P}$ such that $u, v \in V_{dis}(\mathcal{P})$ and $w_i \in V_{com}(\mathcal{P})$, $i = 1, 2, \ldots, 2k$ (see Figure 6a), we replace it by a single edge (u, v) of $\mathcal{P}$ (see Figure 6b). This

[5] Later on this section, we will describe how to cope with the degenerated cases where either the first or the last edge of $\mathcal{P}$ appear in $\mathcal{M}$.

will result into a new path $\mathcal{P}'$ of n' vertices and a new matching $\mathcal{M}'$ with the following properties:

i) $V(\mathcal{P}') = V_{dis}(\mathcal{P})$

ii) $V(\mathcal{M}') = V_{dis}(\mathcal{M})$

iii) $E(\mathcal{P}') \cap E(\mathcal{M}') = \emptyset$

iv) n' is even

v) $|E(\mathcal{M}')| = n'/2$

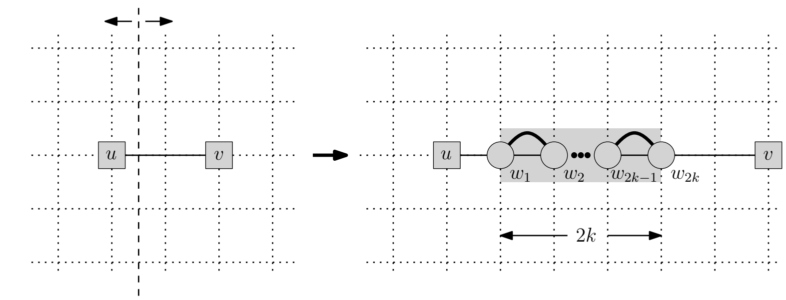
Hence, $\mathcal{P}'$ and $\mathcal{M}'$ can be drawn simultaneously due to Theorem 2. The width (height, resp.) of the produced drawing equals to $|V_{dis}(\mathcal{P})|/2{+}1$ ($|V_{dis}(\mathcal{M})|/2$, resp.) In order to incorporate the removed vertices and edges in the produced drawing, we utilize the fact that the resulting drawing is stretchable.

Let $u \to w_1 \to \ldots \to w_{2k} \to v$ be a subpath of degree-2 vertices of $\mathcal{P}$ which was contracted into a single edge $(u, v) \in E(\mathcal{P}')$, when transforming instance $(\mathcal{P}, \mathcal{M})$ to $(\mathcal{P}', \mathcal{M}')$. We distinguish the following cases:

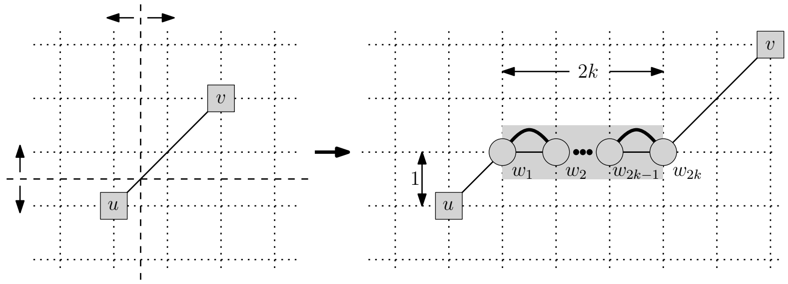
- *Edge (u, v) is drawn as a horizontal line segment in the drawing of* $(\mathcal{P}', \mathcal{M}')$: This case is illustrated in Figure 7a, in which we have assumed that in the drawing of $\mathcal{P}'$ and $\mathcal{M}'$ it holds that $x(u) < x(v)$. In this case, we apply a horizontal stretching in order to allocate $2k$ units of length directly next to u. Then, vertices $w_1,\ w_2, \ldots, w_{2k}$ are drawn at consecutive x-coordinates along the line $y = y(u)$ starting from $x = x(u) + 1$ (i.e., $x(w_i) = x(u) + i + 1$, $i = 1, 2, \ldots, 2k$).

- *Edge (u, v) is drawn as a sloped (neither vertical nor horizontal) line segment in the drawing of* $(\mathcal{P}', \mathcal{M}')$: This case is illustrated in Figure 7b, in which we have assumed that in the drawing of $\mathcal{P}'$ and $\mathcal{M}'$ it holds that $x(u) < x(v)$ and $y(u) < y(v)$. In this case, we first apply a vertical stretching in order to allocate one unit of length directly above u followed by a horizontal stretching, in order to allocate $2k$ units of length directly next to u. In the resulting drawing, vertices $w_1,\ w_2, \ldots, w_{2k}$ are drawn at consecutive x-coordinates along the line $y = y(u) + 1$ starting from $x = x(u) + 1$ (i.e., $x(w_i) = x(u) + i + 1$, $i = 1, 2, \ldots, 2k$).

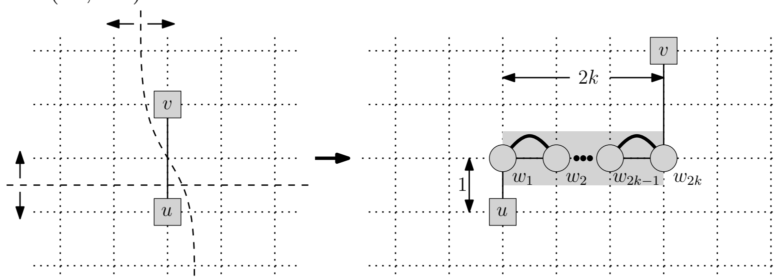
- *Edge (u, v) is drawn as a vertical line segment in the drawing of* $(\mathcal{P}', \mathcal{M}')$: This case is illustrated in Figure 7a and is quite similar to the previous one. Again, without loss of generality, we have assume that in the drawing of $\mathcal{P}'$ and $\mathcal{M}'$ it holds that $y(u) < y(v)$. In this case, we first apply a vertical stretching in order to allocate one unit of length directly above u. Then, we apply a transformation similar to a horizontal stretching, in order to allocate $2k$ units of length between u and v (see Figure 7c). More formally, in order to achieve the transformation illustrated in Figure 7c, we use a vertical live segment that coincides with edge (u, v), and, we shift the top



(a) Edge (u,v) is drawn as a horizontal line segment in the drawing of $(\mathcal{P}',\mathcal{M}')$



(b) Edge (u,v) is drawn as a sloped line segment in the drawing of $(\mathcal{P}',\mathcal{M}')$



(c) Edge (u,v) is drawn as a vertical line segment in the drawing of $(\mathcal{P}',\mathcal{M}')$

Figure 7: Different cases that occur when reinserting a contracted subpath $u \rightarrow w_1 \rightarrow \ldots \rightarrow w_{2k} \rightarrow v$ of degree-2 vertices of $\mathcal{P}$. In the drawings the edges of the path are drawn plain, while the edges of the matching are drawn bold.

endpoint of edge (u,v), while keeping its bottom endpoint in place. In the resulting drawing, vertices $w_1,\ w_2,\ldots,w_{2k}$ are drawn at consecutive x-coordinates along the line $y = y(u)+1$ starting from $x = x(u)$ (i.e., $x(w_i) = x(u)+i$, $i = 1,2,\ldots,2k$).

From the above, it follows that when we reinsert the $2k$ vertices of a contracted subpath of degree-2 vertices of $\mathcal{P}$ the width of the drawing gets larger by $2k$ units of length. So in total, the width of the drawing is at most n. On the other hand, all matching edges are drawn as horizontal line segments. In worst case, no two matching edges share the same horizontal grid line. So in total, the height of the drawing is at most $n/2$.

Recall that in order to cope with the case where $\mathcal{P}$ and $\mathcal{M}$ share edges, we had initially assumed that neither the first nor the last edge of $\mathcal{P}$ appear in $\mathcal{M}$. The reason for this assumption is that the transformations that we have described so far (see Figure 7) require subpaths that start from a vertex of $V_{dis}(\mathcal{P})$ and (through an even number of vertices of $V_{com}(\mathcal{P})$) end to another vertex of $V_{dis}(\mathcal{P})$ again. If, for example, the first vertex of $\mathcal{P}$ belongs to $V_{com}(\mathcal{P})$ (i.e., the first edge of $\mathcal{P}$ appears in $\mathcal{M}$), then there might exist a whole subpath of vertices of $V_{com}(\mathcal{P})$ at the beginning of $\mathcal{P}$[6]. However, such a subpath is not supported by the transformations that we have described so far, since it does not start from a vertex of $V_{dis}(\mathcal{P})$.

If there exists a subpath of vertices of $V_{com}(\mathcal{P})$ at the beginning or at the end of $\mathcal{P}$, we momentarily remove it from $\mathcal{P} \cup \mathcal{M}$ and draw it in a snake-like fashion as illustrated in Figure 8. The remaining part of $\mathcal{P} \cup \mathcal{M}$ is either the empty graph or a graph with the property that neither the first nor the last edge of the path appear in the matching, which can be drawn with the developed algorithm. In the former case, the resulting drawing is a snake-like drawing of $\mathcal{P}$ and $\mathcal{M}$. In the latter case, we simply plug the snake-like drawings of the removed parts to the first and last vertices of the drawing of the remaining part of $\mathcal{P} \cup \mathcal{M}$, which are drawn leftmost and rightmost, respectively. The height of the resulting drawing gets larger by at most two units of length, while its width is at most the number of vertices of the path. This ensures that the total area of the drawing is not affected. The following theorem summarizes our result.

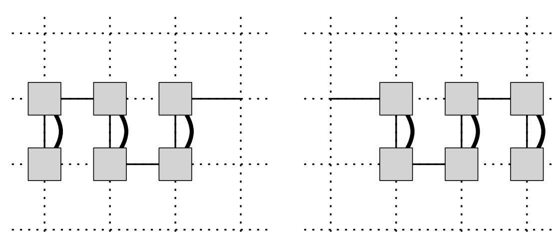


Figure 8: Snake-like drawings in the case where the matching pairs each second tuple of vertices of the path.

Theorem 3 *A path $\mathcal{P}$ and a perfect matching $\mathcal{M}$ on the same vertex set always admit a* GRacSim *drawing on an $n \times n/2$ integer grid, where n is the size of the vertex set. Moreover, the drawing can be computed in linear time.*

We extend the algorithm that produces a GRacSim drawing of a path and a matching to also cover the case of a cycle $\mathcal{C}$ and a matching $\mathcal{M}$. Obviously, if

[6] In worst case, all vertices of $\mathcal{P}$ belong to $V_{com}(\mathcal{P})$, i.e., $\mathcal{M}$ matches each second pair of vertices of $\mathcal{P}$.

we remove an edge from the input cycle (preferably one that belongs to $E(\mathcal{C}) - E(\mathcal{M})$), the remaining graph is a path $\mathcal{P}$ (see Figure 9). Then, we apply the developed algorithm and obtain a GRacSim drawing of $\mathcal{P}$ and $\mathcal{M}$.

We describe in detail the case where $E(\mathcal{P}) \cap E(\mathcal{M}) = \emptyset$. The case where $E(\mathcal{P}) \cap E(\mathcal{M}) \neq \emptyset$ is treated similarly. By Theorem 2, the drawing fits in a $(n/2+1) \times n/2$ integer grid. Additionally, the first vertex of $\mathcal{P}$ is drawn at the bottommost layer (hence its incident edge in $\mathcal{M}$ is not crossed), and the last vertex of $\mathcal{P}$ is drawn rightmost. With these two properties, we can add the removed edge between the first and the last vertex of $\mathcal{P}$ without introducing new crossings. To achieve this, we move the first vertex of $\mathcal{P}$ at most $n/2+2$ units downwards (keeping its x-coordinate unchanged) and the last vertex of $\mathcal{P}$ at most $n/2+1$ units rightwards (keeping its y-coordinate unchanged). Then, the insertion in the drawing of the edge that closes the cycle does not introduce any crossings, as desired. The following theorem summarizes our result.

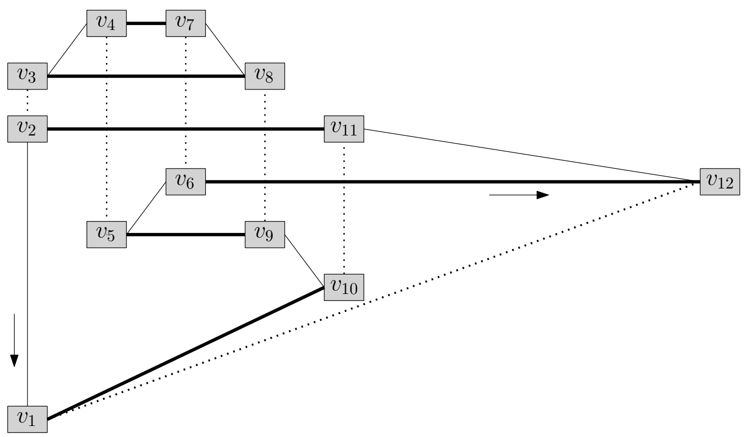


Figure 9: A GRacSim drawing of a cycle and a matching.

Theorem 4 *A cycle $\mathcal{C}$ and a perfect matching $\mathcal{M}$ on the same vertex set and such that $E(\mathcal{C}) \cap E(\mathcal{M}) = \emptyset$ always admit a* GRacSim *drawing on an $(n+2) \times (n+2)$ integer grid, where n is the size of the vertex set. Moreover, the drawing can be computed in linear time.*

As already stated, the case where $E(\mathcal{P}) \cap E(\mathcal{M}) \neq \emptyset$ is treated similarly. Since by Theorem 3 we have to reinsert the deleted edge in a drawing of size $n \times n/2$, the resulting drawing will be of size $3n/2 \times 3n/2$. Hence, we can state the following theorem.

Theorem 5 *A cycle $\mathcal{C}$ and a perfect matching $\mathcal{M}$ on the same vertex set always admit a* GRacSim *drawing on an $3n/2 \times 3n/2$ integer grid, where n is the size of the vertex set. Moreover, the drawing can be computed in linear time.*

4.1 Algorithm Extensions

According to the formal definition of the GRacSim drawing problem, the input graphs share the same vertex set. This immediately implies that $\mathcal{M}$ is a perfect matching and n is even. However, our algorithm can be adjusted to support the case where the input graphs are not necessarily on the same vertex set.

Consider the case where the input consists of a path $\mathcal{P}$ and a matching $\mathcal{M}$, which have to be drawn simultaneously and assume without loss of generality that the union of $\mathcal{P}$ and $\mathcal{M}$ is a connected graph[7]. Let V be the union of $V(\mathcal{P})$ and $V(\mathcal{M})$ and denote by n the size of V. Without loss of generality, we assume that n is even[8]. In a first step, we augment $\mathcal{P}$ so that it spans all vertices of V. Let $\mathcal{P}_{aug}$ be the resulting path. Since n is even, we can augment $\mathcal{M}$ so that it spans all vertices of V, as well. Denote by $\mathcal{M}_{aug}$ the augmented matching. Obviously, $\mathcal{P}_{aug}$ and $\mathcal{M}_{aug}$ are defined on the same vertex set, which is of even size, and $\mathcal{M}_{aug}$ is a perfect matching. Hence, $\mathcal{P}_{aug}$ and $\mathcal{M}_{aug}$ can be drawn simultaneously by the algorithm supporting Theorem 3. In order to obtain a GRacSim of $\mathcal{P}$ and $\mathcal{M}$, it is enough to remove from the GRacSim drawing of $\mathcal{P}_{aug}$ and $\mathcal{M}_{aug}$ the extra edges that were used in order to augment $\mathcal{P}$ and $\mathcal{M}$ to $\mathcal{P}_{aug}$ and $\mathcal{M}_{aug}$, respectively. The following theorem summarizes our result.

Theorem 6 *A path $\mathcal{P}$ and a matching $\mathcal{M}$ always admit a* GRacSim *drawing on an $n \times n/2$ integer grid, where n is the size of the union of $V(\mathcal{P})$ and $V(\mathcal{M})$. Moreover, the drawing can be computed in linear time.*

In the case where the input consists of a cycle $\mathcal{C}$ and a matching $\mathcal{M}$, we follow a similar but slightly different approach, since $\mathcal{C}$ cannot be further augmented. Again, we assume without loss of generality that the union of $\mathcal{C}$ and $\mathcal{M}$ is a connected graph. We proceed as follows. We monetarily remove the vertices that belong exclusively to $\mathcal{M}$ (i.e., their incident edges belong exclusively to $\mathcal{M}$) and their incident edges in $\mathcal{M}$. Then, we proceed to augment $\mathcal{M}$, so that it spans all vertices of $\mathcal{C}$ (and obtain a new matching $\mathcal{M}_{aug}$, as previously). Observe that this is feasible, only when $|V(\mathcal{C})|$ is even. If $|V(\mathcal{C})|$ is odd, then unavoidably there exists a vertex of $\mathcal{C}$ which is not covered by $\mathcal{M}_{aug}$. Say without loss of generality that this is the case. In order to conform with the assumptions of the algorithm supporting Theorem 5, we momentarily remove this particular vertex from $\mathcal{C}$, by connecting its two incident vertices in $\mathcal{C}$ by an edge. The resulting cycle, say $\mathcal{C}_{dec}$, and $\mathcal{M}_{aug}$ can be drawn simultaneously by means of the algorithm supporting Theorem 5, since they are defined on the same vertex set and $\mathcal{M}_{aug}$ is a perfect matching.

In order to obtain a GRacSim of $\mathcal{C}$ and $\mathcal{M}$, we first remove from the GRacSim drawing of $\mathcal{C}_{dec}$ and $\mathcal{M}_{aug}$ the extra edges that were used in order to augment $\mathcal{M}$ to $\mathcal{M}_{aug}$. Then, we proceed to incorporate into the resulting drawing the vertex of $\mathcal{C}$ that was previously removed, when transforming the odd-length cycle $\mathcal{C}$ to the even-length cycle $\mathcal{C}_{dec}$. To achieve this, a horizontal stretching

[7] If this is not the case, our algorithm treats each connected component separately.

[8] If n is odd, we add an isolated vertex in V. Hence, its size become even.

similar to the ones described in the proof of Theorem 3 is enough. However, since both the width and the height of the drawing might become larger by one unit of length, special attention should be paid at the bottommost drawn edge of the cycle (i.e., the one which is drawn neither as a horizontal nor as a vertical line segment; see Figure 9). More precisely, potential crossings posed by this particular edge can be resolved by shifting the bottommost (rightmost, resp.) vertex of the drawing one unit downwards (rightwards, resp.).

Now it remains to explain how to incorporate into the resulting drawing the vertices that belong exclusively to $\mathcal{M}$ and their corresponding edges of the matching (which we had removed at the begging of this procedure). Since we have assumed that the union of $\mathcal{C}$ and $\mathcal{M}$ is a connected graph, each of these vertices should be incident to a vertex of $\mathcal{C}$ (through an edge of $\mathcal{M}$). Let $v \in V(\mathcal{C})$ be such a vertex of $\mathcal{C}$. Obviously, in the drawing constructed so far v is not incident to a matching edge, which implies that either its left or right "port" is free. Hence, we can incorporate its incident matching edge into the drawing constructed so far, as a horizontal line segment of unit length incident to v's free port, through a horizontal stretching directly next to v's free port. Again, potential crossings posed by the presence of the bottommost drawn edge of the cycle can be resolved by shifting the bottommost (rightmost, resp.) vertex of the drawing one unit downwards (rightwards, resp.), each time a new vertex among those belonging exclusive to $\mathcal{M}$ is incorporated. Note that this does not affect the total area occupied by the drawing, which is still $3n/2 \times 3n/2$, where n is the size of the union of $V(\mathcal{C})$ and $V(\mathcal{M})$. The following theorem summarizes our result.

Theorem 7 *A cycle $\mathcal{C}$ and a matching $\mathcal{M}$ always admit a* GRacSim *drawing on an $3n/2 \times 3n/2$ integer grid, where n is the size of the union of $V(\mathcal{C})$ and $V(\mathcal{M})$. Moreover, the drawing can be computed in linear time.*

From the above, it follows that we can identify a class of RAC graphs.

Corollary 1 *Let G be a simple connected graph that can be decomposed into a matching and either a path or a cycle. Then, G is a RAC graph.*

5 A Planar Graph and its Dual: An Interesting Variation

In this section, we examine the GDual-GRacSim drawing problem. This problem can be considered as a variation of the GRacSim drawing problem, where the first graph (i.e., the planar graph) determines the second one (i.e., the dual) and places restrictions on its layout. Recall that according to the GDual-GRacSim drawing problem, we are given a planar embedded graph G and the main task is to determine a geometric drawing of G and its weak dual G^* such that: (i) G and G^* are drawn planar, (ii) each vertex of the dual is drawn inside its corresponding face of G and, (iii) the primal-dual edge crossings form right angles.

As already stated in Section 2, Brightwell and Scheinermann [6] proved that the GDual-GRacSim problem always admits a solution if the input graph is a triconnected planar graph. For the general case of planar graphs, we demonstrate by an example that it is not always possible to compute such a drawing, and thus, we concentrate our study in the case of outerplanar graphs.

Initially, we consider the case where the planar drawing $\Gamma(G)$ of graph G is specified as part of the input and it is required that it remains unchanged in the output. We demonstrate by an example that it is not always feasible to incorporate G^* into drawing $\Gamma(G)$ and obtain a GDual-GRacSim drawing of G and G^*. The example is illustrated in Figure 10. In the following, we prove that if the input graph is a planar embedded graph, then the GDual-GRacSim drawing problem does not always admit a solution, as well.

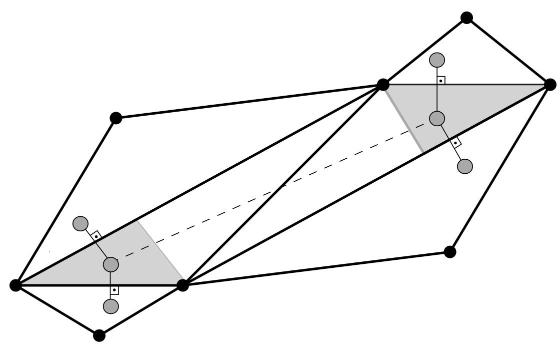


Figure 10: The input planar drawing of the primal graph G is sketched with black colored vertices and bold edges and should remain unchanged in the output. The vertices of the dual G^* are colored gray. Then, the dual's dashed drawn edge will inevitably introduce a non right angle crossing.

Theorem 8 *There exists a planar graph G with the following property: For non of the planar embeddings of G a* GDual-GRacSim *drawing of G and its weak dual G^* is possible.*

Proof: The graph G used to establish the theorem is depicted in Figure 11, where the vertices drawn as boxes belong to the dual graph G^*. Observe that if we replace the two subgraphs drawn with dashed edges by two edges, the resulting graph is a triconnected planar graph, which has unique planar drawing up to the choice of the outerface, translations, rotations and stretchings. This implies that in any planar drawing of G, either $u_1w_1v_1x_1$ or $u_2w_2v_2x_2$ is an internal faces. Without loss of generality, we consider the case where $u_1w_1v_1x_1$ is an internal face. Now, observe that the dual graph should have two vertices within each of the gray-colored faces of Figure 11 (refer to the vertices which are drawn as boxes). Each of these two vertices is incident to two vertices of the dual that lie within the triangular faces of the dashed drawn subgraph of G, incident to the two gray-colored faces. Observe that in any RAC drawing of

G and G^* both quadrilaterals $u_1w_1v_1u_1'$ and $u_1x_1v_1v_1'$ must be convex, which is impossible.

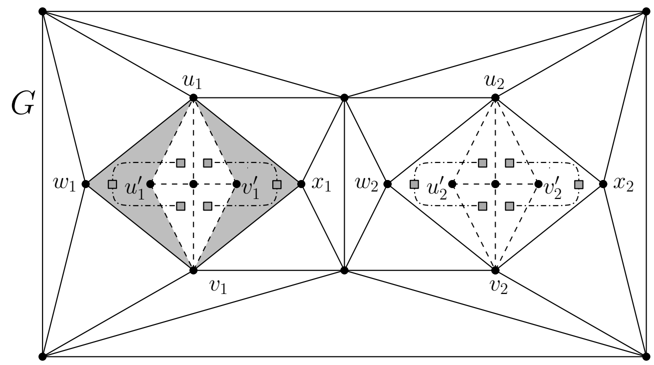


Figure 11: An example of a planar graph G, for which the GDual-GRacSim does not admit a solution. The problematic faces are drawn in gray.

□

Corollary 2 *There exists an infinite class of planar graphs $\mathcal{F}$ with the following property: For any graph $G \in \mathcal{F}$ and any planar embedding $\mathcal{E}(G)$ of G, a* GDual-GRacSim *drawing of G and its weak dual G^* is not possible.*

We now proceed to study the GDual-GRacSim drawing problem in the case where the input graph is outerplanar. Notice that an outerplanar graph is not necessarily triconnected (i.e., by deleting any pair of vertices on the same interior face and non-consecutive on the outerface, the graph is disconnected). On the other hand, one can augment an outerplanar graph to a triconnected planar (but not necessarily outerplanar) graph, by introducing an additional vertex incident to all vertices of the outerplanar graph. This easily implies that an outerplanar embedded graph and its dual always admit a GDual-GRacSim drawing, due Brightwell and Scheinermann [6]. However, their constructive approach simply proves that such a drawing exists and cannot be utilized to construct the corresponding drawing, since the computation involves sequences that do not necessarily converge after a finite number steps.

Theorem 9 *Given an outerplane embedding of an outerplanar graph G, it is always possible to determine a* GDual-GRacSim *drawing of G and its weak dual G^*.*

Proof: The proof is given by a recursive geometric construction which computes a GDual-GRacSim drawing of G and its dual G^*. Consider an arbitrary edge (u, v) of the outerplanar graph that does not belong to its external face and let f and g be the faces to its left and the right side, respectively, as we move along (u, v) from vertex u to vertex v. Then, (f, g) is an edge of the dual graph G^*. Since the dual of an outerplanar graph is a tree, the removal of edge (f, g)

results in two trees T_f and T_g that can be considered to be rooted at vertices f and g of G^*, respectively. For the recursive step of our drawing algorithm, we assume that we have already produced a GDual-GRacSim drawing for T_f and its corresponding subgraph of G that satisfies the following invariant properties:

I-P1: *Edge (u,v) is drawn on the external face of the GDual-GSimRAC drawing of T_f.* Let u and v be drawn at points p_u and p_v, respectively. Denote by $\ell_{u,v}$ the line defined by p_u and p_v.

I-P2: Let the face-vertex f be drawn at point p_f. *The perpendicular from point p_f to line $\ell_{u,v}$ intersects the line segment $p_u p_v$.* Let p be the point of intersection.

I-P3: *There exists two parallel semi-lines ℓ_u and ℓ_v passing from p_u and p_v, respectively, that define a semi-strip to the right of segment $p_u p_v$ that does not intersect the drawing constructed so far.* Denote this empty semi-strip by $R_{u,v}$.

We proceed to describe how to recursively produce a drawing for tree T_g and its corresponding subgraph of G so that the overall drawing is a GDual-GRacSim drawing for G and its dual. Refer to Figure 12. Let p_g be a point in semi-strip $R_{u,v}$ that also belongs to the perpendicular line to line-segment $p_u p_v$ that passes from point p. Thus, the segment corresponding to the edge (f,g) of the dual crosses at right angle the segment corresponding to the edge (u,v) of G, as required. If g is a leaf (i.e., all edges of g except (u,v) are edges of the external face), we can draw the remaining edges of face g as a polyline of appropriate number of points that goes around p_g and connects p_u and p_v.

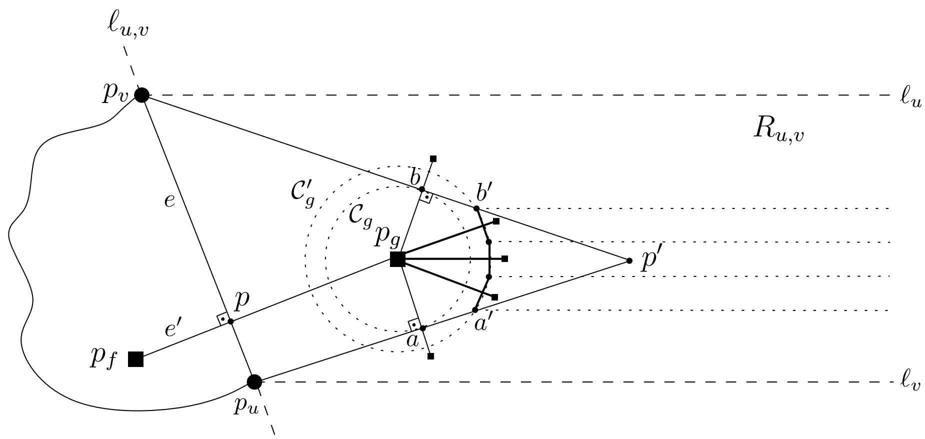


Figure 12: The recursion step of our algorithm.

Consider now the more interesting case where g is not a leaf in the dual tree of G. In this case, we draw two circles, say $\mathcal{C}_g$ and $\mathcal{C}'_g$, centered at p_g such that both lie entirely within the semi-strip $R_{u,v}$ and do not touch neither line

ℓ_u nor line ℓ_v. Assume that circle $\mathcal{C}'_g$ is the external of the two circles. From point p_u draw the tangent to circle $\mathcal{C}_g$ and let a be the point where it touches $\mathcal{C}_g$ and a' be the point to the right of a where the tangent intersects circle $\mathcal{C}'_g$ (see Figure 12). Similarly, we define points b and b' based on the tangent from point p_v to circle $\mathcal{C}_g$.

Let $k \geq 4$ be the number of vertices defining face g. The case where $k = 3$ will be examined later. Draw $k-4$ points on the (a', b') arc, which is furthest from segment $p_u p_v$. These points, say $\{p_i \mid 1 \leq i \leq k-4\}$, together with points p_u, p_v, a' and b' form face g. Observe that from point p_g, we can draw perpendicular lines towards each edge of the face. Indeed, line segments $p_g a$ and $p_g b$ are perpendicular to $p_u a'$ and $p_v b'$, respectively. In addition, the remaining edges of the face are chords of circle $\mathcal{C}'_g$ and thus, we can always draw perpendicular lines to their midpoints from the center p_g of the circle. Now, from each of the newly inserted points of face g draw a semi-line that is parallel to semi-line ℓ_u and lies entirely in the semi-strip $R_{u,v}$. We observe all invariant properties stated above hold for each child of face g in the subtree T_g of the dual of G. Thus, our algorithm can be applied recursively. The case where the number k of vertices defining face g is equal to 3 is treated as follows. We use the intersection of the two tangents, say p', as the third point of the triangular face. We have to be careful so that p' lies inside the semi-strip. However, we can always select a point p_g close to segment $p_u p_v$ and an appropriately small radius for circle $\mathcal{C}_g$, so that p' is inside $R_{u,v}$.

Now that we have described the recursive step of the algorithm, it remains to define how the recursion begins (see Figure 13). We start from any face of G that is a leaf at its dual tree, say face l. We draw the face as regular polygon, with face-vertex l mapped at its center, say p_l. Let $e = (u, v)$ be the only edge of the face that is internal to the outerplane embedding of G. Without loss of generality, assume that e is drawn vertically. Then, draw the horizontal semi-lines ℓ_u and ℓ_v from the endpoints of e in order to define the semi-strip $R_{u,v}$. From this point on, the algorithm can recursively draw the remaining faces of G and and its dual G^*.

□

Note that, the produced GDual-GRacSim drawing of G and its dual proves that producing such drawings is possible. The drawing is not particularly appealing since the height of the strips quickly becomes very small. However, it is a starting point towards algorithms that produce better layouts. Also note that, the algorithm performs a linear number of "point computations" since for each face-vertex of the dual tree the performed computations are proportional to the degree of the face-vertex. However, the coordinates of some points may be non-rational numbers.

6 Conclusion - Open Problems

In this paper, we introduced and examined geometric RAC simultaneous drawings. Our study raises several open problems. Among them are the following:

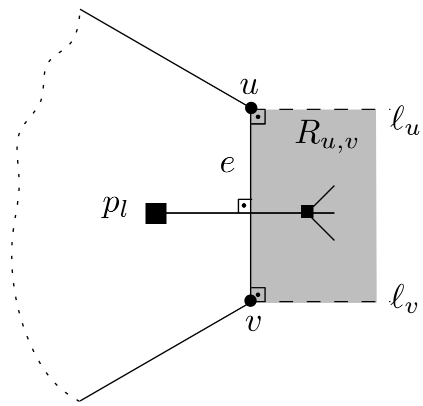


Figure 13: The initial step of our algorithm.

1. What other non-trivial classes of graphs, besides a matching and either a path or a cycle, admit a GRacSim drawing?

2. We considered only geometric RAC simultaneous drawings. For the classes where GRacSim drawings are not possible, study drawings with bends or relax the optimality constraint on the crossing resolution of the produced drawings.

3. We showed that if two graphs admit a geometric simultaneous drawing, it is not necessary that they admit a GRacSim drawing. Finding a class of graphs (instead of a particular graph) with this property would strengthen this result.

4. A quite similar problem to the GRacSim drawing problem is the problem of drawing two (or more) graphs on the same vertex set on the plane, such that each graph is drawn RAC (i.e., only edges of different graphs may introduce non-right angle crossings). Note that, the class of graphs that admit such drawings contains the class of graphs for which a simultaneous drawing is possible.

5. Obtain more appealing GDual-GRacSim drawings for an outerplanar graph and its dual. Study the required drawing area. The characterization of the classes, other than outerplanar graphs, admitting a GDual-GRacSim drawing is also of interest.

Acknowledgements

The work of E.N. Argyriou has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference

Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

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.

References

[1] P. Angelini, L. Cittadini, G. Di Battista, W. Didimo, F. Frati, M. Kaufmann, and A. Symvonis. On the perspectives opened by right angle crossing drawings. *Journal of Graph Algorithms and Applications*, 15(1):53–78, 2011. doi:10.7155/jgaa.00217.

[2] P. Angelini, M. Geyer, M. Kaufmann, and D. Neuwirth. On a tree and a path with no geometric simultaneous embedding. In *Proc. of 18th International Symposium on Graph Drawing*, LNCS, pages 38–49, 2010. doi:10.1007/978-3-642-18469-7_4.

[3] E. N. Argyriou, M. A. Bekos, and A. Symvonis. The straight-line RAC drawing problem is NP-hard. In *Proc. of 37th International Conference on Current Trends in Theory and Practice of Computer Science, (SOFSEM)*, LNCS, pages 74–85, 2011. doi:10.1007/978-3-642-18381-2_6.

[4] K. Arikushi, R. Fulek, B. Keszegh, F. Moric, and C. Toth. Graphs that admit right angle crossing drawings. In *Graph Theoretic Concepts in Computer Science*, volume 6410 of *LNCS*, pages 135–146, 2010. doi:10.1007/978-3-642-16926-7_14.

[5] 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. doi:10.1016/j.comgeo.2006.05.006.

[6] G. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal Discrete Mathematics*, 6(2):214–229, 1993. doi:10.1137/0406017.

[7] S. Cabello, M. J. v.Kreveld, G. Liotta, H. Meijer, B. Speckmann, and K. Verbeek. Geometric simultaneous embeddings of a graph and a matching. *JGAA*, 15(1):79–96, 2011. doi:10.7155/jgaa.00219.

[8] E. Di Giacomo, W. Didimo, G. Liotta, and H. Meijer. Area, curve complexity, and crossing resolution of non-planar graph drawings. In *Proc. of 17th International Symposium on Graph Drawing*, volume 5849 of *LNCS*, pages 15–20, 2009. doi:10.1007/978-3-642-11805-0_4.

[9] W. Didimo, P. Eades, and G. Liotta. A characterization of complete bipartite graphs. *Information Processing Letters*, 110(16):687–691, 2010. doi:10.1016/j.ipl.2010.05.023.

[10] W. Didimo, P. Eades, and G. Liotta. Drawing graphs with right angle crossings. *Theoretical Computer Science*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.

[11] P. Eades and G. Liotta. Right angle crossing graphs and 1-planarity. volume 7034 of *Lecture Notes in Computer Science*, pages 148–153. Springer, 2011. doi:10.1007/978-3-642-25878-7_15.

[12] C. Erten and S. G. Kobourov. Simultaneous embedding of a planar graph and its dual on the grid. *Theory Computing Systems*, 38(3):313–327, 2005. doi:10.1007/s00224-005-1143-4.

[13] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *JGAA*, 9(3):347–364, 2005. doi:10.7155/jgaa.00113.

[14] A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In *Proc of 15th International Symposium on Graph Drawing*, LNCS, pages 280–290, 2007. doi:10.1007/978-3-540-77537-9_28.

[15] F. Frati, M. Kaufmann, and S. G. Kobourov. Constrained simultaneous and near-simultaneous embeddings. *JGAA*, 13(3):447–465, 2009. doi:10.7155/jgaa.00194.

[16] M. Geyer, M. Kaufmann, and I. Vrto. Two trees which are self-intersecting when drawn simultaneously. *Discrete Mathematics*, 309(7):1909–1916, 2009. doi:10.1016/j.disc.2008.01.033.

[17] W. Huang, S.-H. Hong, and P. Eades. Effects of crossing angles. In *Proc. of IEEE Pacific Visualization Symposium*, IEEE, pages 41–46, 2008. doi:10.1109/PACIFICVIS.2008.4475457.

[18] W. T. Tutte. How to draw a graph. *Proc Lond Math Soc*, 13:743–767, 1963.

[19] M. van Kreveld. The quality ratio of RAC drawings and planar drawings of planar graphs. In *Proc. of 18th International Symposium on Graph Drawing*, LNCS, pages 371–376, 2010. doi:10.1007/978-3-642-18469-7_34.