Straight-Line Monotone Grid Drawings of Series-Parallel Graphs

Md. Iqbal Hossain and Md. Saidur Rahman

Graph Drawing and Information Visualization Laboratory,
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology
{mdiqbalhossain,saidurrahman}@cse.buet.ac.bd

Abstract. A monotone drawing of a graph G is a straight line drawing of G where a monotone path exists between every pair of vertices of G in some direction. Recently monotone drawings of graphs have been discovered as a new standard for visualizing graphs. In this paper we study monotone drawings of series-parallel graphs in a variable embedding setting. We show that a series-parallel graph of n vertices has a straight-line planar monotone drawing on a grid of size $O(n) \times O(n^2)$.

1 Introduction

A path P in a straight-line drawing of a graph is *monotone* if there exists a line l such that the orthogonal projections of the vertices of P on l appear along l in the order induced by P. A straight-line drawing of a graph is monotone if it contains at least one monotone path for each pair of vertices.

Upward drawings [4,8] are related to monotone drawings where every directed path is monotone with respect to vertical lines, while in a monotone drawing each monotone path, in general, is monotone with respect to a different line. Arkin *et al.* [3] showed that any strictly convex drawing of a planar graph is monotone and they gave an $O(n \log n)$ time algorithm for finding such a path from s to t. The authors in [1] showed that every biconnected planar graph has a straight-line monotone drawing in real coordinate space. Angelini *et al.* [1] showed that every tree admits a straight-line planar monotone drawing in $O(n) \times O(n^2)$ or $O(n^{1.6}) \times O(n^{1.6})$ area. Every connected plane graph admits a monotone grid drawing on an $O(n) \times O(n^2)$ grid using at most two bends per edges and an outerplane graph of n vertices admits a straight-line monotone drawing on a grid of area $O(n) \times O(n^2)$ [2]. It is also known that not every plane graph (with fixed embedding) admits a straight-line monotone drawing [1].

So the natural question is whether every connected planar graph has a straight-line monotone drawing and what is the minimum area requirement for such a drawing on a grid. In this paper, we investigate this problem for a non-trivial subclass of planar graphs called "series-parallel graphs". We show that every series-parallel graph admits a straight-line monotone drawing on an $O(n) \times O(n^2)$ grid which can be computed in $O(n \log n)$ time.

D.-Z. Du and G. Zhang (Eds.): COCOON 2013, LNCS 7936, pp. 672–679, 2013.

We now give an outline of our algorithm for constructing a monotone drawing of a series-parallel graph G. We construct an ordered "SPQ-tree" of G. We then assign a slope to each node of the SPQ-tree. We finally draw G on a grid taking into consideration the slope assigned to each node of the SPQ-tree.

The rest of the paper is organized as follows. Section 2 describes some of the definitions that we have used in our paper. Section 3 deals with straight-line monotone drawings of series-parallel graphs. Finally, Section 4 concludes our paper with discussions.

2 Preliminaries

Let $G = (V, E)$ be a connected graph with vertex set V and edge set E. A graph is *planar* if it can be embedded in the plane without edge crossings except at the vertices where the edges are incident. A *plane graph* is a planar graph with a fixed planar embedding. A plane graph divides the plane into some connected regions called *faces*. The unbounded region is called *outer face* and all the other faces are called *inner faces*. The vertices on the outer face are called *outer vertices* and all the other vertices are called *inner vertices*. A *cut vertex* is any vertex whose removal disconnects G. A *biconnected component* G' is a maximal biconnected subgraph of G.

A graph $G = (V, E)$ is called a *series-parallel* graph (with source s and sink t) if either G consists of a pair of vertices connected by a single edge or there exist two series-parallel graphs $G_i = (V_i, E_i)$, $i = 1, 2$, with source s_i and sink t_i such that $V = V_1 \cup V_2, E = E_1 \cup E_2$, and either $s = s_1, t_1 = s_2$ and $t = t_2$ or $s = s_1 = s_2$ and $t = t_1 = t_2$ [7]. A biconnected component of a series-parallel graph is also a series-parallel graph. By definition, a series-parallel graph G is a connected planar graph and G has exactly one source s and exactly one sink t.

Fact 1. *Let $G = (V, E)$ be a series-parallel graph with the source vertex s and the sink vertex t. Assume that $(s, t) \notin E(G)$. Then $G' = (V, E \cup (s, t))$ is a planar graph.*

A pair u, v of vertices of a connected graph G is a split pair if there exist two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ satisfying the following two conditions: 1. $V = V_1 \cup V_2, V_1 \cap V_2 = \{u, v\}$; and 2. $E = E_1 \cup E_2, E_1 \cap E_2 = \emptyset, |E_1| \geq 1, |E_2| \geq 1$. Thus every pair of adjacent vertices is a split pair. A *split component* of a split pair u, v is either an edge (u, v) or a maximal connected subgraph H of G such that u, v is not a split pair of H.

Let G be a biconnected series-parallel graph. Let (u, v) be an outer edge of G. The *SPQ-tree* [6,5] $\mathcal{T}$ of G with respect to a reference edge $e = (u, v)$ describes a recursive decomposition of G induced by its split pairs. Tree $\mathcal{T}$ is a rooted ordered tree whose nodes are of three types: S, P and Q. Each node x of $\mathcal{T}$ corresponds to a subgraph of G, called its pertinent graph $G(x)$. Each node x of $\mathcal{T}$ has an associated biconnected multigraph, called the *skeleton* of x and denoted by $skeleton(x)$. Tree $\mathcal{T}$ is recursively defined as follows.

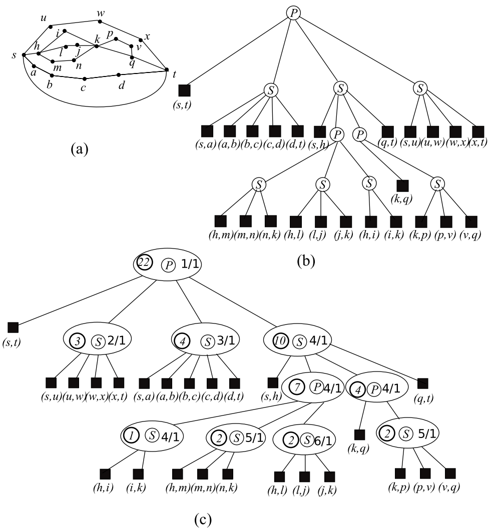

Fig. 1. (a) A series-parallel graph G, (b) an SPQ-tree $\mathcal{T}$ of G, and (c) an illustration for slope assignment in $\mathcal{T}'$ where subtrees are sorted on the number of vertices in each subtree, and number of vertices, node type and assigned slope of each node are written inside the node

Trivial Case: In this case, G consists of exactly two parallel edges e and e' joining s and t. $\mathcal{T}$ consists of a single Q-node x, and the skeleton of x is G itself. The pertinent graph $G(x)$ consists of only the edge e'.

Parallel Case: In this case, the split pair u, v has three or more split components $G_0, G_1, \ldots, G_k, k \geq 2$, and G_0 consists of only a reference edge $e = (u, v)$. The root of $\mathcal{T}$ is a P-node x. The $skeleton(x)$ consists of $k + 1$ parallel edges $e_0, e_1, \ldots, e_k$ joining s and t, where $e_0 = e = (u, v)$ and e_i, $1 \leq i \leq k$, corresponds to G_i. The pertinent graph $G(x) = G_1 \cup G_2 \cup \ldots \cup G_k$ is the union of $G_1, G_2, \ldots, G_k$.

Series Case: In this case the split pair u, v has exactly two split components, and one of them consists of the reference edge e. One may assume that the other split component has cut-vertices $c_1, c_2, \ldots, c_{k-1}$, $k \geq 2$, that partition the component into its blocks $G_1, G_2, \ldots, G_k$ in this order from t to s. Then the root of $\mathcal{T}$ is an S-node x. The skeleton of x is a cycle $e_0, e_1, \ldots, e_k$ where $e_0 = e, c_0 = u, c_k = v$, and e_i joins c_{i-1} and $c_i, 1 \leq i \leq k$. The pertinent graph $G(x)$ of node x is the union of $G_1, G_2, \ldots, G_k$. Figure 1 shows a series-parallel graph and its SPQ-tree decomposition.

Let $\mathcal{T}$ be the SPQ-tree of a series-parallel graph G and let x be a node of $\mathcal{T}$. The pertinent graph of x is denoted by $pert(x)$. For an S-node x, we denote by $n(x)$ the number of vertices in $pert(x)$ excluding s and t. For a P-node x, we denote by $n(x)$ the number of vertices in $pert(x)$ including s and t. According to the SPQ-tree decomposition, a P-node can not be the parent of another P-node and an S-node can not be the parent of another S-node. Throughout the paper, by drawing of a node x in $\mathcal{T}$ we mean the drawing of $pert(x)$ of G.

Monotone Drawings

Let p be a point in the plane and l be a half-line starting at p. The slope of l, denoted by $slope(l)$, is the angle spanned by a counter-clockwise rotation that brings a horizontal half-line starting at p and directed towards increasing x- coordinates to coincide with l.

Let Γ be a drawing of a graph G and let (u, v) be an edge of G. The half-line starting at u and passing through v, denoted by $d(u, v)$, is the *direction of* (u, v). The direction of an edge e is denoted by $d(e)$ and slope of e is denoted by $slope(e)$.

Let $P(u_1, u_q) = (u_1, u_2, \ldots, u_q)$ be a path in a straight-line drawing of a graph. Let e_i be the edge from u_i to u_{i+1} ($1 \le i \le q-1$). Let e_j and e_k be two edges of the path $P(u_1, u_q)$ such that $slope(e_j) \ge slope(e_i)$ and $slope(e_k) \le slope(e_i)$ for $i = 1, \ldots, q-1$. We call e_j and e_k *extremal edges* of the path $P(u_1, u_q)$. The path $P(u_1, u_q)$ is a *monotone path* with respect to a direction d if the orthogonal projections of vertices $u_1, \ldots, u_q$ on a line along the direction d appear in the same order as the vertices appear in the path.

Let $P(u_1, u_q) = (u_1, u_2, \ldots, u_q)$ be a monotone path. Let e_1 and e_2 be the extremal edges of $P(u_1, u_q)$. If we draw e_i at the origin of the axes, e_1 and e_2 create a closed wedge at the origin of the axes. The closed wedge delimited by $d(e_1)$ and $d(e_2)$ and containing all the half-lines $d(u_i, u_{i+1})$, for $i = 1, \ldots, q-1$, is the *range* of $P(u_1, u_q)$ and is denoted by $range(P(u_1, u_q))$, while the closed wedge delimited by $d(e_1) - \pi$ and $d(e_2) - \pi$, and not containing $d(e_1)$ and $d(e_2)$, is the *opposite range* of $P(u_1, u_q)$ and is denoted by $opp(P(u_1, u_q))$.

We now recall some important properties of monotone paths from [1] as in the following two lemmas.

Lemma 1. *A path $P(u_1, u_q) = (u_1, u_2, \ldots, u_q)$ is monotone if and only if it contains two edges e_1 and e_2 such that the closed wedge centered at the origin of the axes, delimited by the two half-lines $d(e_1)$ and $d(e_2)$, and has an angle smaller than π, contains all the half-lines $d(u_i, u_{i+1})$, for $i = 1, \ldots, q-1$.*

Lemma 2. *Let $P(u_1, u_q) = (u_1, \ldots, u_q)$ and $P(u_q, u_{q+k}) = (u_q, ..., u_{q+k})$ be monotone paths. Then, path $P(u_1, u_{q+k}) = (u_1, \ldots, u_q,\ u_{q+1},\ \ldots, u_{q+k})$ is monotone if and only if $range(P(u_1, u_q)) \cap opp(P(u_q, u_{q+k})) = \emptyset$. Further, if $P(u_1, u_{q+k})$ is monotone, then $range(P(u_1, u_q)) \cup range(\ P(u_q, u_{q+k})) \subseteq range(P(u_1, u_{q+k}))$.*

3 Monotone Grid Drawing

In this section we give an $O(n \log n)$ time algorithm to find a straight-line planar monotone grid drawing of a series-parallel graph on an $O(n) \times O(n^2)$ grid. To get a such drawing we first construct an ordered SPQ-tree. We then assign a slope to each node. We finally draw the graph on a grid taking into consideration the slopes assigned to each node of the tree. The details of our algorithm are as follows.

We assume that an edge exists between the source s and the sink t of the input series-parallel graph G otherwise we add a dummy edge between s and t of G. (Note that the graph remains planar after adding the dummy edge (s, t) by Fact 1.) Later we will show that the drawing of G obtained by our algorithm remains monotone even after removing the dummy edge (s, t) form the drawing. Clearly, G is a biconnected series-parallel graph (with the edge (s, t)). Let $\mathcal{T}$ be the SPQ-tree of G with respect to edge (s, t). Then the root of $\mathcal{T}$ is a Q-node r and the only child of r is a P-node x. We now re-root $\mathcal{T}$ at x.

Let $\mathcal{T}'$ be an ordered SPQ-tree obtained from $\mathcal{T}$ as follows. We traverse each P-node of $\mathcal{T}$; if any Q-node exists as a child of a P-node we put the Q-node as the leftmost child of the P-node. The rest of the children of the P-node are S-nodes and we draw them from left to right according to increasing order of the number of vertices in the subtree rooted at the respective S-node. We now assign a slope to each node of $\mathcal{T}'$. Let $1/1, 2/1, \ldots, (n-1)/1$ be $n-1$ slopes in increasing order. Initially we assign the slope $1/1$ to the root of $\mathcal{T}'$. We then traverse $\mathcal{T}'$ to assign a slope to each node x of $\mathcal{T}'$. We first consider the case where x is a P-node. Let the slope assigned to x be $\mu/1$, and let $x_1, x_2, \ldots, x_k$ $(k < n)$ be the children of x in left to right order. We assign the slope $\mu/1$ to the leftmost child x_1. We next assign the slope $(\mu_{i_{max}}+1)/1$ to x_{i+1} where $\mu_{i_{max}}/1$ is the largest slope assigned in the subtree rooted at x_i. We now consider the case where x is an S-node. Let the slope assigned to x be $\mu/1$, and let $x_1, x_2, \ldots, x_k$ $(k < n)$ be the children of x in left to right order. We assign the same slope $\mu/1$ to x_i $(i \leq k)$. Thus the largest slope assigned to a vertex can be at most $(n-1)/1$.

We are now ready to draw G on a grid using the slope assigned to each node of $\mathcal{T}'$. Figure 1(c) illustrates the slope assignment to the nodes of the SPQ-tree for the graph G shown in Figure 1(a). Our algorithm uses a post-order traversal on the ordered SPQ-tree.

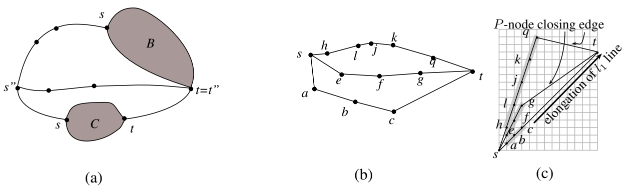

Fig. 2. (a) An example of a P-node for Cases 1 and 2, (b) a P-node x, and (c) a drawing of x on a grid

We first give a drawing algorithm for a P-node. Let x be a P-node with slope $\mu/1$ assigned to it and let s and t be the source and the sink of x, respectively. Let $x_1, x_2, \ldots, x_k$ be the children of x in left to right order. Let $\mu_1/1, \mu_2/1, \ldots, \mu_k/1$ be the slopes assigned to $x_1, x_2, \ldots, x_k$, respectively.

If x is not the root of $\mathcal{T}'$, let x' be the parent node of x, and let x'' be the parent node of x'. Clearly x' is an S-node and x'' is a P-node. Let $\mu/1$ and $\mu'/1$ be the slope assigned to x' and x'', respectively. Let s'' and t'' be the source and the sink of x'', respectively.

We denote the position of a vertex u by $p(u)$; $p(u)$ is expressed by its x- and y-coordinates as $(p_x(u), p_y(u))$ on a grid. Let $p(x)$ be a point on the grid. We place the source vertex of the P-node x on $p(x)$. Let A_x be a set of points where the neighbors of t in $pert(x)$ are to be drawn.

We now have the following two cases to consider.

Case 1: $t \neq t''$. In this case (see the node labeled C in the Figure 2), we place s on $p(x)$. If x_1 is a Q-node we leave it for now, and we draw the respective edge after placing the sink t of x. We add $p(x)$ to A_x. Otherwise all x_i are S-nodes and we follow the drawing algorithm of S-node to draw each x_i.

After drawing all x_i, we elongate the l_1 (the equation of l_1 is $y = \mu_1 \times x + p_x(x)$) line up to the point $p(x_{end}) = (p_x(x) + n(x), p_y(x) + \mu_1 \times n(x))$ and place the sink t of x on $p(x_{end})$. Since $n(x_1) \leq n(x_2) \leq n(x_3) \ldots \leq n(x_k)$ and the slope $\mu_1/1 < \mu_2/1 < \ldots < \mu_k/1$, $p(x_{i_{end}})$ is visible from $p(x_{end})$. We connect t to all points in A_x using straight line segments and we call each of these edges *P-node closing edge*. Note that the slope of edge $(p(x_{i_{end}}), p(x_{end}))$ satisfies $\pi/2 > slope(p(x_{i_{end}}), p(x_{end})) > -\pi/2$.

Case 2: $t = t''$. Figure 2 illustrates an example of this case (see node B).

Let $p_x(Y'')$ be the largest x-coordinate used in the drawing of x''. If $p_x(x) < p_x(Y'')$, we set $p(x) = (p_x(Y''), \frac{p_x(Y'') - p_x(x'')}{\mu})$. We then place the s on $p(x)$. We now draw all the S-nodes according to the S-node drawing algorithm described later. Since the sink vertex of x and x'' are same, we do not draw t in this step. All the end vertices of x_i will be connected at the drawing of the sink of x''. If x_1 is a Q-node, we add $p(x)$ in A_x.

We now describe an algorithm for drawing an S-node. Let x be an S-node of $\mathcal{T}'$ with assigned slope $\mu/1$. Let x' be the parent node of x and let s' be the source vertex of x'. Let $p(x')$ be the point where s' has already been placed. Clearly, x' is a P-node. Let l be a straight-line such that the equation of l is $y = \mu/1 \times x + p_x(x')$.

Assume first that all the children of x are Q-nodes. Then the $pert(x)$ is a path. In this case we place the vertices of $pert(x)$ on the line l sequentially on integer points. The last vertex of $pert(x_i)$ lies on the point $p_{end}(x) = ((p_x(x) + n(x), p_y(x) + \mu/1 \times n(x))$ $(slope(l) = \mu/1)$. Then we add the $p_{end}(x)$ in $A_{x'}$. Assume now that some of the children of x are P-nodes. We traverse left to right subtrees of x. If x_i is a Q-node, we place corresponding vertices on the line l. If x_i is a P-node, we set $p(x_i) = ((p_x(x) + i, p_y(x) + \mu/1 \times i)$ when the source vertex of x_i is not s', otherwise $p(x_i) = p(x')$. We then use the drawing algorithm for P-nodes.

We now describe how we fix the coordinates for the drawing of a Q-node. Let x be a Q-node. The pertinent graph of x is an edge (u, v). Note that u is already placed on the grid, since our drawing algorithm follows post-order traversal on the SPQ-tree. Let (α, β) be the coordinate of u. If v is a sink of any P-node, we handle this in the drawing of P-node closing edges. We thus assume that v is not a sink of any P-node. In this case we place vertex v at $(\alpha+1, \beta+\mu)$ on the line $y = \mu x + \beta - \alpha\mu$, where $\mu/1$ is the slope assigned to x.

We call the algorithm described above Algorithm *Monotone-Draw*. We now have the following theorem.

Theorem 1. *Algorithm Monotone-Draw finds a monotone drawing of a series-parallel graph on a grid of size $O(n) \times O(n^2)$ in $O(n \log n)$ time.*

Proof. Let Γ be the drawing of G constructed by Algorithm Monotone-Draw. We now show that Γ is a monotone drawing of G. To prove the claim, we show that, a monotone path exists between every pair of vertices of G in Γ.

Let s and t be the source and the sink of G. In fact we will prove that a monotone path exists between every pair of vertices of G in the drawing of $G\backslash(s,t)$ in Γ. Let v be a vertex in G such that $v \notin \{s,t\}$. We first show that there exist a monotone path between v and s, and between v and t. One can easily observe that a path $P(s,v)$ exists such that no P-node closing edge is contained in $P(s,v)$ and for each edge $e \in P(v,s)$, $\pi/2 > slope(e) \geq \pi/4$ holds. Then the path $P(s,v)$ is monotone since the $range(P(s,v))$ is smaller than π. On the other hand a path $P(v,t)$ exists such that $s \notin P(v,t), (s,t) \notin P(v,t)$ and $P(v,t)$ may contain some P-node closing edges. The path $P(v,t)$ is monotone since for each edge $e \in P(v,t)$ $\pi/2 < slope(e) < -\pi/2$ holds. Similarly, any path $P(s,t)$ in $\Gamma \setminus (s,t)$ is monotone since for each edge $e \in P(s,t)$ $\pi/2 < slope(e) < -\pi/2$ holds.

We now show that for every pair of vertices $u, v \in G$ ($v \notin \{s,t\}$, $u \notin \{s,t\}$) there is a monotone path between u and v in Γ. Let $P(u,s)$ and $P(v,s)$ be two paths such that none of them contains a P-node closing edge and assume that $e_1 = (u,u')$ lies on $P(u,s)$ and $e_2 = (v,v')$ lies on $P(v,s)$.

Let M and N be the two Q-nodes in $\mathcal{T}'$ such that $(u,u') \in pert(M)$ and $(v,v') \in pert(N)$, and let W be the lowest common ancestor of M and N in $\mathcal{T}'$. Let U and V be the children of W and ancestors of M and N, respectively. Let μ_W, μ_U, μ_V, μ_M, μ_N be the slopes assigned to the nodes W, U, V, M and N, respectively. Since e_1 and e_2 are not P-node closing edges, the slopes of $d(e_1)$ and $d(e_2)$ are $-\mu_M$ and $-\mu_N$, respectively.

We now have the following two cases to consider.

Case 1: *W is a P-node.* Without loss of generality we may consider $\mu_M > \mu_N$. So according to the slope assignment $\mu_M \geq \mu_U > \mu_V \geq \mu_N$. Let w and w' be the source and sink vertices of W in Γ. The path $P(u,v)$ ($w' \notin P(u,v)$) is composed of path $P(u,w)$ and of path $P(w,v)$. Clearly, for each edge $e \in P(u,w)$, and $e' \in P(w,v)$ it holds $\mu_M \geq slope(e) \geq \mu_U$ and $\mu_N \geq slope(e') \geq \mu_V$, respectively. So we have $range(P(u,w)) \cap opp(P(w,v)) = \emptyset$. Then by Lemma 2, $P(u,v)$ is a monotone path.

Case 2: *W is an S-node.* If M and N are children of the same S-node then the case is straight-forward, the path $P(u,v)$ lies on a straight-line. Otherwise the path $P(u,v)$ could have some P-node closing edge. let W' be the parent of W. Clearly W' is a P-node. Let w' be the source vertex of W' in Γ. Then the path $P(u,v)(w' \notin P(u,v))$ is monotone with respect to a horizontal half-line.

Thus we have proved that Γ is a monotone drawing of G.

We are placing the sink of each P-node on the $x = n(x)$ line. The drawings of all child nodes are inside the drawing of its parent P-node. So the largest x-coordinate of the drawing can be at most n. On the other hand we might get B type nodes (see Figure 2) recursively, and hence the y-coordinate can be up to $O(n^2)$. Thus the total grid size is $O(n) \times O(n^2)$.

We now analyze the required time for our algorithm. We construct SPQ-tree in linear-time, and $O(n \log n)$ time is required to sort. We assign slopes to $\mathcal{T}'$ in linear time. Thus the overall time complexity of the algorithm is $O(n \log n)$. □

4 Conclusion

In this paper we have studied monotone grid drawings of series-parallel graphs. We have shown that a series-parallel graph of n vertices has a straight-line planar monotone drawing on an $O(n) \times O(n^2)$ grid and such a drawing can be found in $O(n \log n)$ time. Finding straight-line monotone grid drawings of larger classes of planar graphs is remained as our future work.

Acknowledgment. We thank CodeCrafters International and Investortools, Inc. for supporting this research under the grant "CodeCrafters-Investortools Research Grant for CSE BUET".

References

1. Angelini, P., Colasante, E., Di Battista, G., Frati, F., Patrignani, M.: Monotone drawings of graphs. Journal of Graph Algorithms and Applications 16(1), 5–35 (2012)
2. Angelini, P., Didimo, W., Kobourov, S., Mchedlidze, T., Roselli, V., Symvonis, A., Wismath, S.: Monotone drawings of graphs with fixed embedding. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 379–390. Springer, Heidelberg (2011)
3. Arkin, E.M., Connelly, R., Mitchell, J.S.: On monotone paths among obstacles with applications to planning assemblies. In: Proceedings of the Fifth Annual Symposium on Computational Geometry, SCG 1989, pp. 334–343. ACM, New York (1989)
4. Di Battista, G., Tamassia, R.: Algorithms for plane representations of acyclic digraphs. Theoretical Computer Science 61(2-3), 175–198 (1988)
5. Di Battista, G., Tamassia, R.: On-line maintenance of triconnected components with spqr-trees. Algorithmica 15(4), 302–318 (1996)
6. Di Battista, G., Tamassia, R., Vismara, L.: Output-sensitive reporting of disjoint paths. Algorithmica 23, 302–340 (1999)
7. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of series-parallel graphs. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 409–420. Springer, Heidelberg (2006)
8. Samee, M.A.H., Rahman, M.S.: Upward planar drawings of series-parallel digraphs with maximum degree three. In: WALCOM 2007, pp. 28–45 (2007)