Orthogonal Drawings of Series-Parallel Graphs with Minimum Bends*

Xiao Zhou and Takao Nishizeki

Graduate School of Information Sciences, Tohoku University
Aoba-yama 6-6-05, Sendai, 980-8579, Japan.
zhou|nishi@ecei.tohoku.ac.jp

Abstract. In an orthogonal drawing of a planar graph G, each vertex is drawn as a point, each edge is drawn as a sequence of alternate horizontal and vertical line segments, and any two edges do not cross except at their common end. A bend is a point where an edge changes its direction. A drawing of G is called an optimal orthogonal drawing if the number of bends is minimum among all orthogonal drawings of G. In this paper we give an algorithm to find an optimal orthogonal drawing of any given series-parallel graph of the maximum degree at most three. Our algorithm takes linear time, while the previously known best algorithm takes cubic time. Furthermore, our algorithm is much simpler than the previous one. We also obtain a best possible upper bound on the number of bends in an optimal drawing.

Keywords: orthogonal drawing, bend, series-parallel graph, planar graph.

1 Introduction

Automatic graph drawings have numerous applications in VLSI circuit layouts, networks, computer architecture, circuit schematics, etc. [3, 10]. Many graph drawing styles have been introduced [1, 3, 8, 10, 14, 16]. Among them, an "orthogonal drawing" has attracted much attention due to its various applications, specially in circuit schematics, entity relationship diagrams, data flow diagrams, etc. [13, 15, 18, 19]. An *orthogonal drawing* of a planar graph G is a drawing of G such that each vertex is mapped to a point, each edge is drawn as a sequence of alternate horizontal and vertical line segments, and any two edges do not cross except at their common end. A point where an edge changes its direction in a drawing is called a *bend* of the drawing. Figure 1(a) depicts an orthogonal drawing of the planar graph in Fig. 1(b); the drawing has exactly one bend on the edge joining vertices g and t. If a planar graph G has a vertex of degree five or more, then G has no orthogonal drawing. On the other hand, if G has no vertex of degree five or more, that is, the maximum degree Δ of G is at most

* This work is supported by JSPS grants.
M. Kaykobad and Md. Saidur Rahman (Eds.): WALCOM 2007, pp. 3–12, 2007.

four, then G has an orthogonal drawing, but may need bends. If a planar graph represents a VLSI routing, then one may be interested in an orthogonal drawing such that the number of bends is as small as possible, because bends increase the manufacturing cost of a VLSI chip. An orthogonal drawing of a planar graph G is called an *optimal* orthogonal drawing if it has the minimum number of bends among all possible orthogonal drawings of G.

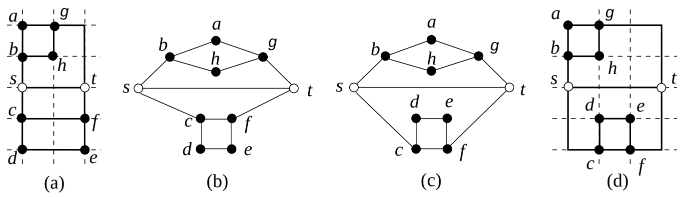

Fig. 1. (a) An optimal orthogonal drawing with one bend, (b), (c) two embeddings of the same planar graph, and (d) an orthogonal drawing with three bends.

The problem of finding an optimal orthogonal drawing is one of the most famous problems in the graph drawing literature [3, 10] and has been studied both in the fixed embedding setting [6, 13, 15, 17, 19] and in the variable embedding setting [5, 7, 12]. A planar graph with a fixed embedding is called a *plane graph*. As a result in the fixed embedding, Tamassia [19] presented an algorithm to find an optimal orthogonal drawing of a plane graph G in time $O(n^2 \log n)$ where n is the number of vertices in G; he reduced the optimal drawing problem to a min-cost flow problem. Then Garg and Tamassia improved the complexity to $O(n^{7/4}\sqrt{\log n})$ [6]. As a result in the variable embedding setting, Garg and Tamassia showed that the problem is NP-complete for planar graphs of $\Delta \leq 4$ in the variable embedding setting [7]. However, Di Battista *et al.* [5] showed that the problem can be solved in polynomial time for planar graphs G of $\Delta \leq 3$. Their algorithm finds an optimal orthogonal drawing among all possible plane embeddings of G. They use the properties of "spirality," min-cost flow techniques, and a data structure, call a SPQ*R-tree that implicitly represents all the plane embeddings of G. The algorithm is complicated and takes time $O(n^5 \log n)$ for planar graph of $\Delta \leq 3$. Using the algorithm, one can find an optimal orthogonal drawing of a biconnected series-parallel simple graph of $\Delta \leq 4$ and of $\Delta \leq 3$ in time $O(n^4)$ [5] and in time $O(n^3)$ [4], respectively. Note that every series-parallel graph is planar. Series-parallel graphs arise in a variety of problems such as scheduling, electrical networks, data-flow analysis, database logic programs, and circuit layout [20]. The complexities $O(n^5 \log n)$, $O(n^4)$ and $O(n^3)$ above for the variable embedding setting are very high, and it is expected to obtain an efficient algorithm for a particular class of planar graphs of $\Delta \leq 3$ [2].

In this paper we deal with the class of series-parallel (multi)graphs of $\Delta \leq 3$, and give a simple linear algorithm to find an optimal orthogonal drawing in the variable embedding setting. The graph G in Fig. 1 is series-parallel, and has various plane embeddings; two of them are illustrated in Figs. 1(b) and (c); there is no plane embedding having an orthogonal drawing with no bend; however, the embedding in Fig. 1(b) has an orthogonal drawing with one bend as illustrated in Fig. 1(a) and hence the drawing is optimal; the embedding in Fig. 1(c) needs three bends as illustrated in Fig. 1(d); given G, our algorithm finds an optimal drawing in Fig. 1(a). Our algorithm works well even if G has multiple edges or is not biconnected, and is much simpler and faster than the algorithms for biconnected series-parallel simple graphs in [4, 5]; we use neither the min-cost flow technique nor the SPQ*R tree, but uses some structural features of series-parallel graphs, which have not been exploited in [20]. We furthermore obtain a best possible upper bound on the minimum number of bends.

The rest of the paper is organized as follows. In Section 2 we present some definitions and our main idea. In Section 3 we present an algorithm and an upper bound for biconnected series-parallel graphs. Finally Section 4 is a conclusion. We omit a linear algorithm for non-biconnected series-parallel graphs in this extended abstract, due to the page limitation.

2 Preliminaries

In this section we present some definitions and our main idea.

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. We denote the number of vertices in G by $n(G)$ or simply by n. For a vertex $v \in V$, we denote by $G - v$ the graph obtained from G by deleting v. An edge joining vertices u and v is denoted by uv. We denote by $G - uv$ the graph obtained from G by deleting uv. We denote the degree of a vertex v in G by $d(v, G)$ or simply by $d(v)$. We denote the maximum degree of G by $\Delta(G)$ or simply by Δ. A connected graph is *biconnected* if there is no vertex whose removal results in a disconnected graph or a single-vertex graph K_1. A *plane graph* is a fixed embedding of a planar graph.

Let G be a planar graph of $\Delta \leq 3$. We denote by $bend(G)$ the number of bends of an optimal orthogonal drawing of G in the variable embedding setting. (Thus $bend(G) = 1$ for the graph G in Fig. 1.) Let D be an orthogonal drawing of G. The number of bends in D is denoted by $bend(D)$. Of course, $bend(G) \leq bend(D)$. Let $G(D)$ be a plane graph obtained from a drawing D by replacing each bend in D with a new vertex. Figures 2(a) and (b) depict $G(D)$ for the drawings D in Figs. 1(a) and (d), respectively. An angle formed by two edges e and e' incident to a vertex v in $G(D)$ is called an *angle of vertex* v if e and e' appear consecutively around v. An angle of a vertex in $G(D)$ is called an *angle of the plane graph* $G(D)$. In an orthogonal drawing, every angle is $\pi/2$, π, $3\pi/2$ or 2π. Consider a labeling l which assigns a label 1, 0, -1 or -2 to every angle of $G(D)$. Labels 1, 0, -1 and -2 correspond to angles $\pi/2$, π, $3\pi/2$ and 2π, respectively.

We call l a *regular labeling* of $G(D)$ if l satisfies the following three conditions (a)–(c) [10, 19]:

(a) for each vertex v of $G(D)$,
  (a-1) if $d(v) = 1$ then the label of the angle of v is -2,
  (a-2) if $d(v) = 2$ then the labels of the two angles of v total to 0, and
  (a-2) if $d(v) = 3$ then the labels of the three angles of v total to 2;
(b) the sum of the labels of each inner face is 4; and
(c) the sum of the labels of the outer face is -4.

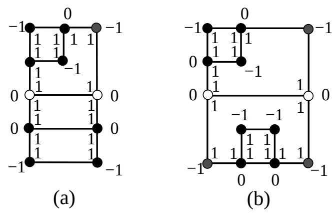

Fig. 2. Regular labelings of $G(D)$ corresponding to the drawings D in Figs. 1(a) and (d), respectively.

Figures 2(a) and (b) illustrate regular labelings for the orthogonal drawings in Figs. 1(a) and (d), respectively. If D is an orthogonal drawing of G, then clearly $G(D)$ has a regular labeling. Conversely, every regular labeling of $G(D)$ corresponds to an orthogonal drawing of G [19]. An orthogonal (geometric) drawing of G can be obtained from a regular labeling of $G(D)$ in linear time, that is, in time $O(n(G) + bend(D))$ [10, 19]. Therefore, from now on, we call a regular labeling of $G(D)$ an *orthogonal drawing of a planar graph* G or simply a *drawing of* G, and obtain a regular labeling of G in place of an orthogonal (geometric) drawing of G.

A *series-parallel graph (with terminals s and t)* is recursively defined as follows:

(a) A graph G of a single edge is a series-parallel graph. The ends s and t of the edge are called the *terminals* of G.
(b) Let G_1 be a series-parallel graph with terminals s_1 and t_1, and let G_2 be a series-parallel graph with terminals s_2 and t_2.
  (i) A graph G obtained from G_1 and G_2 by identifying vertex t_1 with vertex s_2 is a series-parallel graph, whose terminals are $s = s_1$ and $t = t_2$. Such a connection is called a *series connection.*
  (ii) A graph G obtained from G_1 and G_2 by identifying s_1 with s_2 and t_1 with t_2 is a series-parallel graph, whose terminals are $s = s_1 = s_2$ and $t = t_1 = t_2$. Such a connection is called a *parallel connection.*

For example, the graph in Fig. 1 is series-parallel.

Throughout the paper we assume that the maximum degree of a given series-parallel graph G is at most three, that is, $\Delta \leq 3$. We may assume without loss of generality that G is a simple graph, that is, G has no multiple edges, as follows. If a series-parallel multigraph G consists of exactly three multiple edges, then G has an optimal drawing of four bends; otherwise, insert a dummy vertex of degree two into an edge of each pair of multiple edges in G, and let G' be the resulting series-parallel simple graph, then an optimal drawing of the multigraph G can be immediately obtained from an optimal drawing of the simple graph G' by replacing each dummy vertex with a bend.

A drawing D of a series-parallel graph G is *outer* if the two terminals s and t of G are drawn on the outer face of D. A drawing D is called an *optimal outer drawing* of G if D is outer and $bend(D) = bend(G)$. The graph in Fig. 1 has an optimal outer drawing as illustrated in Fig. 1(a). On the other hand, the graph in Fig. 3(a) has no optimal outer drawing for the specified terminals s and t; the no-bend drawing D in Fig. 3(b) is optimal but is not outer, because s is not on the outer face; and the drawing $D^{\rm o}$ with one bend in Fig. 3(c) is outer but is not optimal.

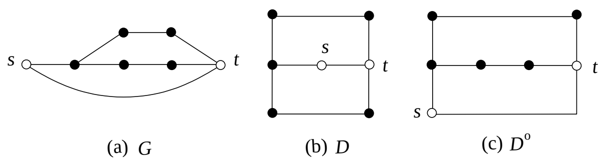

Fig. 3. (a) A biconnected series-parallel graph G, (b) an optimal drawing D, and (c) an outer drawing $D^{\rm o}$.

Our main idea is to notice that a series-parallel graph G has an optimal outer drawing if G is "2-legged." We say that G is 2-*legged* if $n(G) \geq 3$ and $d(s) = d(t) = 1$ for the terminals s and t of G. The edge incident to s or t is called a *leg* of G, and the neighbor of s or t is called a *leg-vertex*. For example, the series-parallel graphs in Figs. 4(a)–(c) are 2-legged.

We will show in Section 3 that every 2-legged series-parallel graph G has an optimal outer drawing and the drawing has one of the three shapes, "I-shape," "L-shape" and "U-shape," defined as follows. An outer drawing D of G is *I-shaped* if D intersects neither the north side of terminal s nor the south side of terminal t after rotating the drawing and renaming the terminals if necessary, as illustrated in Fig. 4(a). D is *L-shaped* if D intersects neither the north side of s nor the east side of t after rotating the drawing and renaming the terminals if necessary, as illustrated in Fig. 4(b). D is *U-shaped* if D does not intersect the north sides of s and t after rotating the drawing and renaming the terminals if necessary, as illustrated in Fig. 4(c). In Figs. 4(a)–(c) each side is shaded. The north side and the south side of a terminal contain the horizontal line

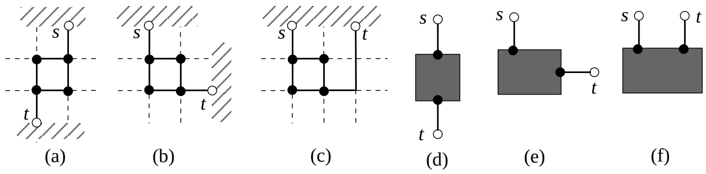

Fig. 4. (a)–(c) I-, L- and U-shaped drawings, and (d)–(f) their schematic representations.

passing through the terminal, while the east side of a terminal contains the vertical line passing through the terminal. The schematic representations of I-, L-, and U-shaped drawings are depicted in Figs. 4(d), (e) and (f), respectively. D is called an *optimal X-shaped drawing*, X=I, L and U, if D is X-shaped and $bend(D) = bend(G)$.

More precisely, we will show in Section 3 that every 2-legged series-parallel graph G with $n(G) \geq 3$ has both an optimal I-shaped drawing and an optimal L-shape drawing and that G has an optimal U-shaped drawing, too, unless G is a "diamond graph," defined as follows. A *diamond graph* is either a path with three vertices or obtained from two diamond graphs by connecting them in parallel and adding two legs.

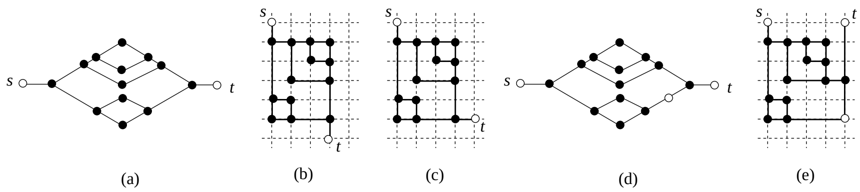

Fig. 5. (a) Diamond graph, (b) I-shaped drawing, (c) L-shaped drawing, (d) non-diamond graph, and (e) U-shaped drawing.

For example, the 2-legged series-parallel graph in Fig. 5(a) is a diamond graph, and has both an optimal (no-bend) I-shaped drawing and an optimal (no-bend) L-shaped drawing as illustrated in Figs. 5(b) and (c), but does not have an optimal (no-bend) U-shaped drawing. On the other hand, the 2-legged series-parallel graph in Fig. 5(d) is obtained from the diamond graph in Fig. 5(a) simply by inserting a new vertex of degree two in an edge, and is not a diamond graph any more. It has an optimal (no-bend) U-shaped drawing, too, as illustrated in Fig. 5(e). Thus the diamond graph in Fig. 5(a) has a U-shaped drawing with one bend.

3 Optimal Drawing of Biconnected Series-Parallel Graph

In this section we give a linear algorithm to find an optimal drawing of a biconnected series-parallel graph G of $\Delta \leq 3$. We first give an algorithm for 2-legged series-parallel graphs in Subsection 3.1. Using the algorithm, we then give an algorithm for biconnected series-parallel graphs in Subsection 3.2.

3.1 2-legged Series-Parallel Graph

We first have the following lemma on a diamond graph. (The proofs of all theorems and lemmas are omitted in this extended abstract, due to the page limitation.)

Lemma 1. *If G is a diamond graph, then*

(a) *G has both a no-bend* I*-shaped drawing D_{I} and a no-bend* L*-shaped drawing D_{L};*
(b) *D_{I} and D_{L} can be found in linear time; and*
(c) *every no-bend drawing of G is either* I*-shaped or* L*-shaped, and hence G does not have a no-bend* U*-shaped drawing.*

The proof of Lemma 1 immediately yields a linear algorithm **Diamond**$(G, D_{\mathrm{I}}, D_{\mathrm{L}})$ which recursively finds both a no-bend I-shaped drawing D_{I} and a no-bend L-shaped drawing D_{L} of a given diamond graph G.

The following lemma holds for a 2-legged series-parallel graph G which is not a diamond graph.

Lemma 2. *The following* (a) *and* (b) *hold for a* 2*-legged series-parallel graph G with $n(G) \geq 3$ unless G is a diamond graph:*

(a) *G has three optimal* I-, L- *and* U*-shaped drawings D_{I}, D_{L} and D_{U};*
(b) *D_{I}, D_{L} and D_{U} can be found in linear time; and*
(c) *$bend(G) \leq (n(G) - 2)/3$.*

We denote by K_n a complete graph of $n(\geq 1)$ vertices. Let G be a 2-legged series-parallel graph obtained from copies of K_2 and K_3 by connected them alternately in series, as illustrated in Fig. 6. Then $bend(G) = (n(G)-2)/3$. Thus the bound in Lemma 2(c) is best possible.

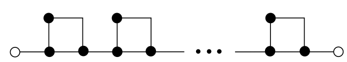

Fig. 6. A graph attaining the bound in Lemma 2(c).

The proof of Lemma 2 immediately yields a linear algorithm **Non-Diamond**(G, D_{I}, D_{L}, D_{U}) which recursively finds three optimal I-, L- and U-shaped drawings D_{I}, D_{L} and D_{U} of a given 2-legged series-parallel graph G unless G is a diamond graph. By algorithms **Diamond** and **Non-Diamond** one can find an optimal drawing of a 2-legged series-parallel graph G.

3.2 Biconnected Series-parallel Graphs

A biconnected series-parallel graph G can be defined (without specifying terminals) as a biconnected graph which has no K_4 as a minor. For every edge uv in G, G is a series-parallel graph with terminals u and v.

A cycle C of four vertices in a graph G is called a *diamond* if two non-consecutive vertices of C have degree two in G and the other two vertices of C have degree three and are not adjacent in G. We denote by G/C the graph obtained from G by contracting C to a new single vertex v_C. (Note that $G_C = G/C$ is series-parallel if G is series-parallel. One can observe that, from every diamond graph, one can obtain a path with three vertices by repeatedly contracting a diamond.)

Noting that every biconnected series-parallel graph has a vertex of degree two, one can easily observe that the following Lemma 3 holds. (Lemma 3 is also an immediate consequence of Lemma 2.1 in [9] on general series-parallel graphs.)

Lemma 3. *Every biconnected series-parallel graph G of $\Delta \leq 3$ has, as a subgraph, one of the following three substructures* (a)–(c)*:*

(a) *a diamond C;*
(b) *two adjacent vertices u and v such that $d(u) = d(v) = 2$; and*
(c) *a complete graph K_3 of three vertices u, v and w such that $d(v) = 2$.*

Our idea is to reduce the optimal drawing problem for a biconnected series-parallel graph G to that for a smaller graph G' as in the following Lemma 4.

Lemma 4. *Let G be a biconnected series-parallel graph with $n(G) \geq 6$.*

(a) *If G has a diamond C, then $bend(G) = bend(G')$ for $G' = G/C$.*
(b) *If G has a substructure (b) in Lemma 3(b), then $bend(G) = bend(G')$ for $G' = G - uv$.*
(c) *If G has a substructure (c) in Lemma 3(c), then $bend(G) = bend(G') + 1$ for $G' = G - v - uw$.*

From the proof of Lemma 4 we have the following algorithm **Biconnected**(G, D) to find an optimal drawing D of a biconnected series-parallel graph G.

Biconnected(G, D);
begin
One may assume that $n(G) \geq 6$ (otherwise, one can easily find an optimal drawing D of G in linear time);
{ By Lemma 3 G has one of the three substructures (a)–(c) in Lemma 3. }
Case 1: G has a diamond C;
Let $G' = G/C$; { G' is a biconnected series-parallel graph. }
Biconnected(G', D');
Extend an optimal drawing D' of G' to an optimal drawing D of G simply by replacing v_C by a rectangular drawing of C;
Case 2: G has no diamond, but has a substructure (b);

Let $G' = G - uv$;
{ G' is a 2-legged series-parallel graph with terminals u and v, and is not a diamond graph. }
Find an optimal U-shaped drawing D'_{U} of G' by **Non-Diamond**;
{ cf. Lemma 2 }
Extend D'_{U} to an optimal drawing D of G by drawing uv as a straight line segment; { cf. Lemma 4 }
Case 3: G has neither a diamond nor a substructure (b), but has a substructure (c);
Let $G' = G - v - uw$;
{ G' is a 2-legged series-parallel graph with terminals u and w, and is not a diamond graph. }
Find an optimal U-shaped drawing D'_{U} of G' by **Non-Diamond**;
Extend D'_{U} to an optimal drawing D of G by drawing $K_3 = uvw$ as a rectangle with one bend; { cf. Lemma 4 }
end

All substructures (a)–(c) can be found total in time $O(n)$ by a standard bookkeeping method to maintain all degrees of vertices together with all paths of length two with an intermediate vertex of degree two. One can thus observe that **Biconnected** can be executed in linear time.

We thus have the following theorem.

Theorem 1. *An optimal orthogonal drawing of a series-parallel biconnected graph G of $\Delta \leq 3$ can be found in linear time.*

4 Conclusions

In this paper, we gave a linear algorithm to find an optimal orthogonal drawing of a series-parallel graph G of $\Delta \leq 3$ in the variable embeddings setting. Our algorithm works well even if G has multiple edges or is not biconnected, and is simpler and faster than the previously known one for biconnected series-parallel simple graphs [4, 5]. One can easily extend our algorithm so that it finds an optimal orthogonal drawing of a partial 2-tree of $\Delta \leq 3$. Note that the so-called block-cutvertex graph of a partial 2-tree is a tree although the block-cutvertex graph of a series-parallel graph is a path. One can prove that $bend(G) \leq \lceil n(G)/3 \rceil$ for every biconnected series-parallel graph and $bend(G) \leq (n(G) + 4)/3$ for every series-parallel graph. The bounds on $bend(G)$ are best possible.

In an *orthogonal grid drawing*, every vertex has an integer coordinate. The *size* of an orthogonal grid drawing is the sum of width and height of the minimum axis-parallel rectangle enclosing the drawing. One can prove that every biconnected series-parallel graph G of $\Delta \leq 3$ has an optimal orthogonal grid drawing of size $\leq 2N/3 + 1$, where $N = n(G) + bend(G)$.

It is left as a future work to obtain a linear algorithm for a larger class of planar graphs.

Acknowledgments. We thank Dr. Md. Saidur Rahman for fruitful discussions.

References

1. P. Bertolazzi, R. F. Cohen, G. Di Battista, R. Tamassia and I. G. Tollis, *How to draw a series-parallel digraph*, Proc. of SWAT '92, pp. 272-283, 1992.
2. F. Brandenburg, D. Eppstein, M. T. Goodrich, S. Kobourov, G. Liotta and P. Mutzel, *Selected open problems in graph drawings*, Proc. of GD '03, LNCS, 1912, pp. 515-539, 2004.
3. G. Di Battista, P. Eades, R. Tamassia and I. G. Tollis, *Graph Drawing: Algorithm for the Visualization of Graphs*, Prentice-Hall Inc., Upper Saddle River, New Jersey, 1999.
4. G. Di Battista, G. Liotta and F. Vargiu, *Spirality and optimal orthogonal drawings*, Tech. Rept. 07.94, Dipartimento di Informatica e Sistemistica, Universita' di Roma "La Sapienza," 1994.
5. G. Di Battista, G. Liotta and F. Vargiu, *Spirality and optimal orthogonal drawings*, SIAM J. Comput., 27(6), pp. 1764–1811, 1998.
6. A. Garg and R. Tamassia, *A new minimum cost flow algorithm with applications to graph drawing*, Proc. of GD '96, LNCS, 1190, pp. 201–226, 1997.
7. A. Garg and R. Tamassia, *On the computational complexity of upward and rectilinear planarity testing*, SIAM J. Comput., 31(2), pp. 601–625, 2001.
8. S. H. Hong, P. Eades and S. H. Lee, *Drawing series-parallel digraphs symmetrically*, Comput. Geom., 17, pp. 165–188, 2000.
9. M. Juvan, B. Mohar and R. Thomas, *List edge-colorings of series-parallel graphs*, The Electronic J. Combinatorics, 6(42), pp. 1–6, 1999.
10. T. Nishizeki and M. S. Rahman, *Planar Graph Drawing*, World Scientific, Singapore, 2004.
11. K. Nomura, S. Taya and S. Ueno, *On the orthogonal drawing of outerplanar graphs*, Proc. of COCOON '04, LNCS, 3106, pp. 300–308, 2004.
12. M. S. Rahman, N. Egi and T. Nishizeki, *No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs*, IEICE Trans. Inf. & Syst., E88-D(1), pp. 23-30, 2005.
13. M. S. Rahman and T. Nishizeki, *Bend-minimum orthogonal drawings of plane 3-graphs*, Proc. of WG '02, LNCS, 2573, pp. 265–276, 2002.
14. M. S. Rahman, S. Nakano and T. Nishizeki, *Rectangular grid drawings of plane graphs*, Comp. Geom. Theo. App., 10(3), pp. 203–220, 1998.
15. M. S. Rahman, S. Nakano and T. Nishizeki, *A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs*, J. of Graph Alg. App., 3(4), pp. 31-62, 1999.
16. M. S. Rahman, S. Nakano and T. Nishizeki, *Box-rectangular drawings of plane graphs*, J. of Algorithms, 37, pp. 363–398, 2000.
17. M. S. Rahman, S. Nakano and T. Nishizeki, *Orthogonal drawings of plane graphs without bends*, J. of Graph Alg. App., 7(4), pp. 335–362, 2003.
18. J. Storer, *On minimum node-cost planar embeddings*, Networks, 14, pp. 181–212, 1984.
19. R. Tamassia, *On embedding a graph in the grid with the minimum number of bends*, SIAM J. Comput., 16, pp. 421–444, 1987.
20. K. Takamizawa, T. Nishizeki and N. Saito, *Linear-time computability of combinatorial problems on series-parallel graphs*, J. Assoc. Comput. Mach., 29, pp. 623–641, 1982.