Complexity of Finding Non-Planar Rectilinear Drawings of Graphs

Ján Maňuch[1,2,⋆], Murray Patterson[1,⋆⋆],
Sheung-Hung Poon[3,⋆⋆⋆], and Chris Thachuk[1,†]

[1] Dept. of Computer Science, University of British Columbia, Vancouver BC, Canada
[2] Dept. of Mathematics, Simon Fraser University, Burnaby BC, Canada
[3] Dept. of Computer Science, National Tsing Hua University, Taiwan, R.O.C.

Abstract. We study the complexity of the problem of finding non-planar rectilinear drawings of graphs. This problem is known to be NP-complete. We consider natural restrictions of this problem where constraints are placed on the possible orientations of edges. In particular, we show that if each edge has prescribed direction "left", "right", "down" or "up", the problem of finding a rectilinear drawing is polynomial, while finding such a drawing with the minimum area is NP-complete. When assigned directions are "horizontal" or "vertical" or a cyclic order of the edges at each vertex is specified, the problem is NP-complete. We show that these two NP-complete cases are fixed parameter tractable in the number of vertices of degree 3 or 4.

1 Introduction

In this paper, we study the rectilinear (or bendless orthogonal) drawing of graphs, where each edge is drawn either as a horizontal or vertical line segment. Such drawings are important for various applications such as VLSI circuit design, entity-relationship diagrams for databases, flow chart drawings in software engineering, and subway-map design. This work is also motivated by the increasing research interest in *RAC (Right Angle Crossing)* drawing [1,7]. Note that a rectilinear drawing of a graph is a RAC-drawing with the additional property that the angles between adjacent incident edges around a vertex are multiples of $\pi/2$.

Formally, a *rectilinear drawing/embedding* of a graph $G = (V, E)$ is the pair of mappings $x, y : V \to \mathbb{Z}$, where $x(u)$ and $y(u)$ represent the x and y coordinates of vertex u on the rectangular grid, such that each edge $\{u, v\} \in E$ is mapped to endpoints of a horizontal or vertical line segment that does not contain any other mapped vertices but its endpoints u and v. Observe that if the maximum degree of G is larger than 4, then it does not have a rectilinear drawing. Such a

⋆ Research supported by NSERC Discovery Grant.
⋆⋆ Research supported by NSERC PGS-D.
⋆⋆⋆ Research supported by NSC Grant 97-2221-E-007-054-MY3 in Taiwan.
† Research supported by NSERC PGS-D and MSFHR Trainee Award.

U. Brandes and S. Cornelsen (Eds.): GD 2010, LNCS 6502, pp. 305–316, 2011.

drawing/embedding is called *planar* if none of embedded edges cross; otherwise, it is called *non-planar*. We remark here that all embedded edges are straight, and thus do not contain any bends.

There are several variants of the rectilinear drawing problem which put restrictions on how each edge is drawn. The most studied is the following variant. Associated with the input graph G is a function λ which assigns each oriented edge $\vec{e} = (u, v) \in \vec{E}$ of G one of the following four labels: L, R, D, and U, where $\lambda(u, v) = \mathrm{L}$ (R) means edge $\vec{e}$ should be drawn horizontally to the left (right) of the source vertex u, and $\lambda(u, v) = \mathrm{D}$ (U) means edge $\vec{e}$ should be drawn vertically below (above) the source vertex u. A graph with edges labeled in this way will be called an *LRDU-restricted graph* and is specified as $G = (V, E, \lambda)$. An *HV-restricted graph* $G = (V, E, \lambda)$ is a graph with each of its oriented edges $\vec{e} = (u, v)$ labeled with H or V. An edge $\vec{e} = (u, v)$ labeled with $\lambda(u, v) = \mathrm{H}$ should be drawn on the plane horizontally; whereas an edge labeled $\lambda(u, v) = \mathrm{V}$ should be drawn vertically. Furthermore, we also consider a different kind of restriction on the edges of the input graph, in which the cyclic ordering of the incident edges around each vertex is fixed in every rectilinear drawing of this graph. Such graphs will be called *cyclic-restricted*. Cyclic-restricted graphs with a planar embedding are exactly the so-called graphs with a fixed embedding. On the other hand, a fixed cyclic ordering of edges around a vertex is an important constraint in the definition of a fixed embedding condition for a non-planar graph [1]. This motivates us to investigate rectilinear drawings of such a kind of graphs. A graph with no restriction on the edges will be called *unrestricted*.

Planar rectilinear drawings have been extensively investigated in the literature of graph drawing. Garg and Tamassia [5] showed that deciding whether a graph is rectilinear planar is NP-complete. However, there are efficient algorithms, in fact linear-time algorithms, to find (construct) planar rectilinear drawings of *plane* graphs of maximum vertex degree three [13], subdivisions of planar triconnected cubic graphs [11], and series-parallel graphs of maximum vertex degree three [12]. These algorithms apply to unrestricted graphs. Some other research also considered the restricted variants. Vijayan and Wigderson [14] present a linear-time decision algorithm and a $O(n^2)$-time algorithm for finding a planar rectilinear drawing of an LRDU-restricted graph. Following this work, Hoffman and Kriegel [6] present an improved linear-time algorithm that finds such a drawing. However, Patrignani [10] showed that it is NP-complete to find a planar rectilinear drawing with minimum area for an LRDU-restricted graph. Recently, Eppstein [3] investigated the rectilinear drawing problem in three dimensions, and showed that it is NP-complete to determine whether an unrestricted graph has a rectilinear drawing with the constraint that at most two vertices lie on the same axis-parallel line.

On the other hand, non-planar rectilinear drawing has not been so well studied. Formann et al. [4] showed that it is NP-hard to decide whether a graph of maximum vertex degree 4 has a straight-line drawing with angular resolution $\frac{\pi}{2}$. This is equivalent to say that it is NP-hard to decide whether there is a drawing/embedding of an unrestricted graph. Further, Eades, Hong and Poon [2]

showed that the problem is NP-hard even for an unrestricted graph consisting of 4-cycle blocks connected by single edges. In this paper, we investigate variants of the existence and the area-minimization problems of non-planar rectilinear drawings of given graphs. In particular, we show that the problem of deciding whether an LRDU-restricted graph has a rectilinear drawing (and finding such a drawing) is polynomial (Section 2.3), while the problem is NP-complete for HV-restricted graphs (Section 2.1) and cyclic-restricted graphs (Section 2.2). We then show that the NP-complete cases are fixed parameter tractable (FPT) in the number of vertices of degree 3 or 4 (Section 2.4). In addition, we show that finding a rectilinear drawing of an LRDU-restricted graph with minimum area is NP-complete as well (Section 3.1). The following table summarizes these results (the results marked with * follow immediately from the NP-completeness of the existence version):

Input Graph	Existence		Area-minimization
unrestricted	NP-c ([4])	FPT (Th. 4)	NP-c*
HV-restricted	NP-c (Th. 1)	FPT (Th. 4)	NP-c*
cyclic-restricted	NP-c (Th. 2)	FPT (Th. 4)	NP-c*
LRDU-restricted	P (Th. 3)		NP-c (Th. 5)

2 Existence of Rectilinear Drawings

2.1 Rectilinear Drawings of HV-Restricted Graphs

Theorem 1. *Given an HV-restricted graph G, the problem of deciding if G has a rectilinear drawing is NP-complete.*

Proof. It is clear that this problem is in NP. We show it is NP-hard by a reduction from the Betweenness (BTW) problem, proved to be NP-complete by Opatrny [8]. The input for the BTW problem is a set $S = \{1, \ldots, n\}$ and set of triples $t_i = (t_{i,1}, t_{i,2}, t_{i,3}) \in S^3$ for $i \in \{1, \ldots, k\}$, and the problem is to determine if there is an injective mapping $f : S \to \mathbb{Z}$ on the elements S, such that for every $i \in \{1, \ldots, k\}$, either $f(t_{i,1}) < f(t_{i,2}) < f(t_{i,3})$ or $f(t_{i,3}) < f(t_{i,2}) < f(t_{i,1})$.

Given an instance I of the BTW problem we construct an HV-restricted graph $G = (V, E, \lambda)$ on the following $3k$ vertices: for $i \in \{1, \ldots, k\}$ and every triple $t_i = (t_{i,1}, t_{i,2}, t_{i,3}) \in S^3$, we add the corresponding vertices $v_{i,1}$, $v_{i,2}$ and $v_{i,3}$. The idea of the construction is that the x-coordinates of the vertices of any rectilinear drawing x, y of G will correspond to values of their corresponding elements of S assigned by a solution f. The goal is then to add edges to the graph in such a way that (i) each triple constraint is enforced and that (ii) for every $s \in S$, the set $V_s = \{v_{i,j} \mid t_{i,j} = s\}$ of vertices in G that correspond to s are all assigned to the same x-coordinate.

To ensure (i), for $i \in \{1, \ldots, k\}$, for the set of vertices $v_{i,1}, v_{i,2}, v_{i,3}$ corresponding to each triple t_i we set $\lambda(v_{i,1}, v_{i,2}) = \lambda(v_{i,2}, v_{i,3}) = \mathrm{H}$. This ensures that for every rectilinear drawing x, y of G, that $x(v_{i,2})$ is between $x(v_{i,1})$ and

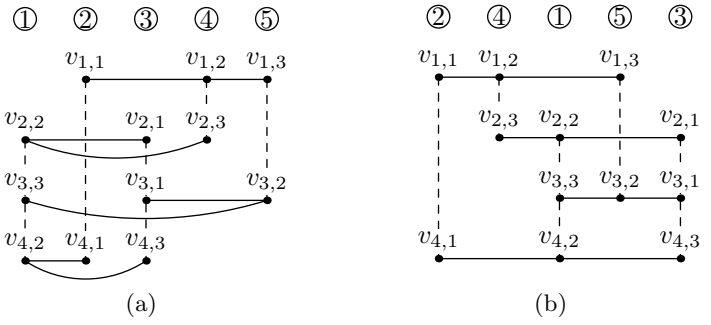

Fig. 1. (a) The input graph constructed for the instance of the BTW problem with triples $(2,4,5)$, $(3,1,4)$, $(3,5,1)$ and $(2,1,3)$. The dashed lines represent edges labeled with V. (b) Rectilinear drawing of this graph corresponding to the solution $f(2) < f(4) < f(1) < f(5) < f(3)$ of this instance.

$x(v_{i,3})$. Now to ensure (ii), for every $s \in S$, let $v_{i_1,j_1}, v_{i_2,j_2}, \ldots, v_{i_p,j_p}$ be the elements of V_s ordered such that $i_1 < i_2 < \cdots < i_p$. For $\ell \in \{1, \ldots, p-1\}$, we set $\lambda(v_{i_\ell,j_\ell}, v_{i_{\ell+1},j_{\ell+1}}) = \mathrm{V}$. This path containing vertical edges through the set V_s ensures that they have the same x-coordinate for every rectilinear drawing x, y of G. An example of a graph constructed for an instance of the BTW problem is shown in Figure 1(a). Figure 1(b) shows a rectilinear drawing of this graph. We now show that instance I has a solution if and only if G has a rectilinear drawing.

If I has a solution f, then we draw G as follows. The x-coordinates of vertices in V are set according to the injective mapping f and y-coordinates are set according to the order on the triples: $x(v_{i,j}) = f(t_{i,j})$ and $y(v_{i,j}) = i$. Obviously, the pair x, y satisfies all restrictions on edges. To show that it is a rectilinear drawing, we need the following lemma. For any $v \in V$, we denote $(x,y)(v) = (x(v), y(v))$.

Lemma 1. *For edge $\{v_{i,j}, v_{i',j'}\} \in E$, there is no vertex $v_{i'',j''} \in V \setminus \{v_{i,j}, v_{i',j'}\}$ such that $(x,y)(v_{i'',j''})$ lies on the line segment $L = [(x,y)(v_{i,j}), (x,y)(v_{i',j'})]$.*

Proof. Assume for contradiction that the image of vertex $v_{i'',j''} \in V \setminus \{v_{i,j}, v_{i',j'}\}$ lies on line segment L. If $\lambda(v_{i,j}, v_{i',j'}) = \mathrm{H}$, then $i = y(v_{i,j}) = y(v_{i',j'}) = i'$ and $j = j' \pm 1$. Since $(x,y)(v_{i'',j''})$ lies on L, we have $i'' = y(v_{i'',j''}) = i = i'$, i.e., $t_{i,j}$, $t_{i',j'}$ and $t_{i'',j''}$ are from the same triple t_i, i.e., $\{j, j', j''\} = \{1,2,3\}$. Furthermore, $x(v_{i'',j''}) = f(t_{i'',j''})$ is between $x(v_{i,j}) = f(t_{i,j})$ and $x(v_{i',j'}) = f(t_{i',j'})$. Hence, $j'' = 2$ which contradicts the fact that $j = j' \pm 1$.

If $\lambda(v_{i,j}, v_{i',j'}) = \mathrm{V}$, then we have $t_{i,j} = t_{i',j'} = s$ and $v_{i,j}, v_{i',j'} \in V_s$. Since $(x,y)(v_{i'',j''})$ lies on L, $f(t_{i'',j''}) = f(s)$. Since f is injective, $t_{i'',j''} = s$, i.e., $v_{i'',j''} \in V_s$. It follows that vertices $v_{i,j}, v_{i',j'}, v_{i'',j''}$ lie on the path of V-edges, hence, $(x,y)(v_{i'',j''})$ cannot lie on L in any rectilinear drawing of G. □

Conversely, assume G has a rectilinear drawing x, y. Such a drawing must satisfy conditions (i) and (ii) discussed above. By condition (ii), for every $s \in S$, every

vertex in V_s has the same x-coordinate. We will construct mapping f by assigning this x-coordinate of vertices in V_s to s. To ensure that f is injective, it might be necessary to modify the drawing x, y slightly: if for two different values $s, s' \in S$, points in V_s and points in $V_{s'}$ are mapped to the same x-coordinate, we will slightly offset the x-coordinate of points in one of the two sets. After this modification, the pair x, y will remain a rectilinear drawing which satisfies all edge constraints. Now, condition (i) will guarantee that f is a solution to I.

Given an instance I to the BTW problem, we have constructed an HV-restricted graph G in time polynomial in the size of I that has a rectilinear drawing if and only if the instance I has a solution. Thus the problem of deciding if G has a rectilinear drawing is NP-hard. □

2.2 Rectilinear Drawings of Cyclic-Restricted Graphs

Theorem 2. *Given a cyclic-restricted graph G, the problem of deciding if G has a rectilinear drawing is NP-complete.*

Proof. Clearly, the problem is in NP. In order to show that the problem is NP-hard, we give a reduction from the 3SAT problem. The input instance of the 3SAT problem is a set $\{x_1, x_2, \ldots, x_n\}$ of n variables, and a collection $\{c_1, c_2, \ldots, c_m\}$ of m clauses, where each clause consists of exactly three literals. The 3SAT problem is to determine whether there exists a truth assignment to the variables so that each clause has at least one true literal. In the following, we will describe our linear-time reduction, which is based on the construction of Formann *et al.* [4].

First we construct an L-shaped skeleton (denoted by K) by connecting a series of 4-cycles together and attaching ports that connect to variable towers and clause gadgets as depicted in Figure 2. The upward spikes sitting on the base of this L-shaped skeleton are the ports that connect to the variable towers, and the 2-edge paths hanging on the right hand side of the vertical column of the skeleton are ports that connect to the clause gadgets.

The variable tower for variable x_i is constructed and connected to the skeleton as shown in Figure 3. Since the cyclic order of the incident edges to every vertex is fixed, there are only two possible configurations of this tower as depicted in Figure 3(a) and (b). These will represent the *true* and *false* states, respectively, of variable x_i. The spikes $x_{i,j}$ and $\overline{x_{i,j}}$ on this variable tower will represent the literals x_i and $\overline{x_i}$, respectively, of clause c_j (if c_j contains x_i), where the truth value in c_j of this literal will correspond to the state of variable x_i, as determined by the configuration of its variable tower. Note that in each of these two configurations, the literals pointing to the right always have the *true* values.

Each clause gadget consists of three 3-edge paths connecting the port of this clause to spikes (on the corresponding variable towers) of the literals it contains. We illustrate the construction of clause gadgets with an example of constructing the gadget for clause $c_j = x_i \vee x_l \vee \overline{x_k}$, as depicted in Figure 4. Since one of these 3 paths incident to port c_j must contain an edge pointing to the right (depending on which direction this port is bent), this path forces its corresponding variable

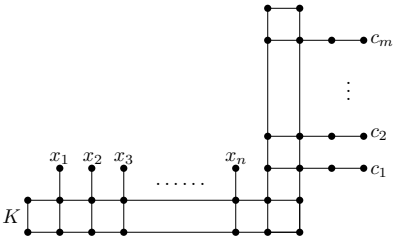

Fig. 2. L-shaped skeleton with ports that connect to variable towers and clause gadgets

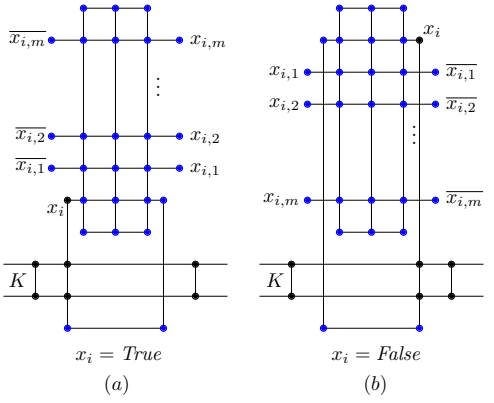

Fig. 3. Variable tower for x_i and the representation of its truth values. Note that the literals on the right have value *true*.

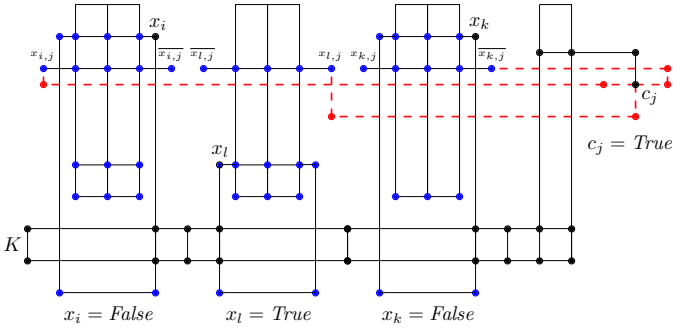

Fig. 4. An example of a clause gadget for clause $c_j = x_i \vee x_l \vee \overline{x_k}$. The port of c_j is bent down which forces the last literal $\overline{x_k}$ to have value *true*.

tower to be in the configuration that sets its literal in c_j to true, in this example, literal $\overline{x_k}$. Finally, it is easy to see that the 3SAT formula has a satisfying assignment if and only if the constructed graph has a rectilinear drawing. Thus this problem is NP-hard. □

2.3 Rectilinear Drawings of LRDU-Restricted Graphs

Theorem 3. *Given an LRDU-restricted graph $G = (V, E, \lambda)$, the problem of deciding if G has a rectilinear drawing and finding such a drawing can be done in time $O(|V||E|)$.*

Proof. We will give a polynomial-time algorithm for the problem. Given LRDU-restricted graph $G = (V, E, \lambda)$, we first check to see if G satisfies the following necessary conditions: for every $u, v, w \in V$ and $X \in \{\mathrm{L}, \mathrm{R}, \mathrm{D}, \mathrm{U}\}$, (i) $\lambda(u, v) \in \{\mathrm{L}, \mathrm{R}, \mathrm{D}, \mathrm{U}\}$ iff $\{u, v\} \in E$; (ii) $\lambda(u, v) = \mathrm{L}$ iff $\lambda(v, u) = \mathrm{R}$ and $\lambda(u, v) = \mathrm{D}$

iff $\lambda(v,u) = \mathrm{U}$; and (iii) if $\lambda(u,v) = X$ then $\lambda(w,v) \neq X$, i.e., that two edges cannot start at the same vertex and have the same direction. Checking (i) and (ii) each takes time $O(|E|)$, while checking (iii) takes time $O(|V||E|)$.

If this check succeeds, we will define equivalence relations E_x and E_y on the vertices of G, and construct partial orders P_x and P_y on equivalence classes of E_x and E_y, respectively, if possible. To construct E_x (E_y), for vertices $u, v \in V$, we set $u \equiv_{E_x} v$ ($u \equiv_{E_y} v$) when $\lambda(u,v) \in \{\mathrm{D}, \mathrm{U}\}$ ($\lambda(u,v) \in \{\mathrm{L}, \mathrm{R}\}$). Each class $V|_{E_x}$ ($V|_{E_y}$) of equivalence relation E_x (E_y) specifies then a set of vertices of G which must have the same x-coordinate (y-coordinate) in any rectilinear drawing of G. To construct P_x (P_y), for classes $A, B \in V|_{E_x}$ ($V|_{E_y}$), we set $A < B$ when there exist $u \in A$ and $v \in B$ such that $\lambda(u,v) = \mathrm{R}$ ($\lambda(u,v) = \mathrm{U}$). The partial order P_x (P_y) on the classes of E_x (E_y) then specifies the partial ordering that the x-coordinates (y-coordinates) of these classes must have on the x-axis (y-axis) of the rectangular grid of this rectilinear drawing. We say that such a rectilinear drawing models (E_x, E_y, P_x, P_y). Orders P_x and P_y can be built (if they exist) independently of each other and in time $O(|V||E|)$. We will show that P_x and P_y exist if and only if G has a rectilinear drawing, i.e., to solve the problem it is enough to decide if partial orders P_x and P_y exist.

Since (E_x, E_y, P_x, P_y) express necessary conditions on any rectilinear drawing of G, if there exists a rectilinear drawing of G, then it models (E_x, E_y, P_x, P_y), i.e., P_x and P_y must exist. On the other hand, given (E_x, E_y, P_x, P_y) we can easily construct mappings x, y which satisfy all edge restrictions as follows. We first extend P_x (P_y) to any two total orders (this can be done in time $O(|V| + |E|)$). Then we assign to equivalence classes in $V|_{E_x}$ ($V|_{E_y}$) unique x (y) coordinates which respect these total orders, and for any $u \in A \in V|_{E_x}$ ($u \in B \in V|_{E_y}$) set $x(u)$ ($y(u)$) equal to the coordinate assigned to the class. Finally, we draw each edge $\{u,v\} \in E$ as line segment $L = [(x(u), y(u)), (x(v), y(v))]$. To show that mappings x, y form a rectilinear drawing, we need the following lemma.

Lemma 2. *For any pair of vertices $\{u,v\} \in E$, there is no vertex $w \in V \setminus \{u,v\}$ such that $(x(w), y(w))$ lies on the line segment $L = [(x(u), y(u)), (x(v), y(v))]$.*

Proof. Since $\{u,v\} \in E$, then $\lambda(u,v) \in \{\mathrm{L}, \mathrm{R}, \mathrm{D}, \mathrm{U}\}$, so it would be assigned to one of the equivalence classes in $V|_{E_x}$ or $V|_{E_y}$, i.e., $x(u) = x(v)$ or $y(u) = y(v)$. We can assume without loss of generality that $y(u) = y(v)$. Assume, for contradiction, that vertex $(x(w), y(w))$ lies on the line segment L. Thus $y(w) = y(u) = y(v)$, and the only way this can happen is when $w \equiv_{E_y} u \equiv_{E_y} v$, otherwise w would have been placed above or below u and v in the drawing. Now, if $(x(w), y(w))$ lies on the line segment L then either $x(u) \leq x(w) \leq x(v)$ or $x(v) \leq x(w) \leq x(u)$; without loss of generality, we assume the former. Since $x(u) \leq x(w) \leq x(v)$ and $w \equiv_{E_y} u \equiv_{E_y} v$, either there is a $u' \equiv_{E_y} u$ such that $x(u') \leq x(u)$ and $\lambda(u', w) = \mathrm{R}$ or there is a $v' \equiv_{E_y} v$ such that $x(v) \leq x(v')$ and $\lambda(w, v') = \mathrm{R}$. In either case, this contradicts the fact that the check for property (iii) succeeded. □

By the above lemma, it follows that if P_x and P_y exist, we can find a rectilinear drawing of G in time $O(|V||E|)$. □

2.4 Fixed-Parameter Algorithms

In previous work, Eades, Hong and Poon showed that the problem of finding non-planar rectilinear drawings is fixed parameter tractable, where the parameter k is the number of vertices of degree 3 or 4, more precisely, they obtained that it can be solved in $O(24^k \cdot k^{2k} \cdot n)$-time [2]. In this work, we build on their idea to improve the runtime complexity of the algorithm and in addition consider rectilinear drawings for cyclic-restricted, and HV-restricted graphs. We sketch our proof here, and leave its detailed proof for the full version of this paper.

Theorem 4. *Given an unrestricted, cyclic-restricted, or HV-restricted degree-4 graph G of order n, a rectilinear drawing of G can be found in linear-time, or more precisely, in $O(24^k \cdot k^{2k+1}+n)$, $O(12^k \cdot k^{2k+1}+n)$, and $O(4^k \cdot k^{2k+1}+n)$-time, respectively, where k, the number of vertices of degree 3 or 4, is a constant.*

Proof (Sketch). Let K be the set of these k degree-3 or degree-4 vertices. We refer to any vertex $v \in K$ as a *high degree vertex*, or simply an *hd-vertex*. We call a path of degree-2 vertices connecting two *hd*-vertices an *hd-path*.

We first consider the case where G is unrestricted. Consider an *hd*-path p going from vertex u to vertex v. Suppose e_u and e_v are the two end edges of p incident to u and v, respectively. Depending on the current embedding of K, there are at most four choices of orientation for each of the two edges e_u and e_v. Thus there are at most 16 combinations. For each of these combinations, we know exactly how many edges, say m, path p needs to possess so that the connection between u and v can be built and routed around the other *hd*-vertices. We further know that m is at most five: if p possesses at least five edges, then no matter what the orientations of e_u and e_v are, the connection between u and v can be built. Hence, we perform a prepossessing step by traversing input graph G, to compute the lengths of all *hd*-paths. Since the maximum number of *hd*-paths is $2k$, this can be done in time proportional to the size of G, i.e., $O(n)$.

Since finding a routing of all *hd*-paths is sufficient for finding a rectilinear embedding of the entire graph G, we simply enumerate the number of possibilities to be checked to give us the running time of this algorithm. For any embedding of the vertices in K on the plane, it is possible that no two share a common coordinate, thus resulting in k unique horizontal and vertical coordinates. Therefore, any embedding in the plane can be considered a distribution of the vertices of K on a $k \times k$ grid. Hence, there are no more than $(k^2)^k = k^{2k}$ possible embeddings to consider. Since each vertex $v \in K$ is incident to at most four edges, each having one of four possible orientations, there are at most $4! = 24$ possible orientations of edges incident to v, and at most 24^k possible orientations for all vertices in K. Therefore, considering this number of possibilities, the time to check them and the initial length calculation of all $O(k)$ *hd*-paths, a rectilinear drawing for G (if one exists) can be found in $O(24^k \cdot k^{2k+1} + n)$-time.

Finding a rectilinear drawing when G is cyclic-restricted (resp. HV-restricted) needs to consider at most 12 (resp. 4) possible orientations of edges incident to

each hd-vertex resulting in $O(12^k \cdot k^{2k+1} + n)$ (resp. $O(4^k \cdot k^{2k+1} + n)$) time overall. Note that the HV-restricted case also considers the sequence of H and V transitions along each hd-path in the same time bound. □

3 Area-Minimization Drawings

3.1 Rectilinear Drawings of LRDU-Restricted Graphs

Theorem 5. *Given an LRDU-restricted graph G, the problem of deciding if G has a rectilinear drawing with minimum area is NP-complete.*

Proof. We show that this problem is NP-complete by reduction from 3SAT(3), a restricted version of 3SAT, proved NP-complete by Papadimitriou [9], where each variable appears exactly twice positive and once negated in the clauses. Given an instance ϕ of 3SAT(3) with n variables and m clauses, we construct a graph G which has a rectilinear drawing on a $10m \times 6n$ grid if and only if ϕ is satisfiable. Graph G will consist of $m \times n$ blocks, where each block is of three different types depicted in Figure 5. The i-th row of blocks corresponds to variable x_i and the j-th column of blocks corresponds to clause c_j. We will refer to the block in the i-th row and the j-th column, as the block at position (i, j). If neither x_i nor $\bar{x}_i$ appear in c_j, then the block at position (i, j) is the no-occurrence block depicted in Figure 5(a). If x_i appears in c_j, the positive-occurrence block in Figure 5(b) is used. If $\bar{x}_i$ appears in c_j, the negative-occurrence block in Figure 5(c) is used.

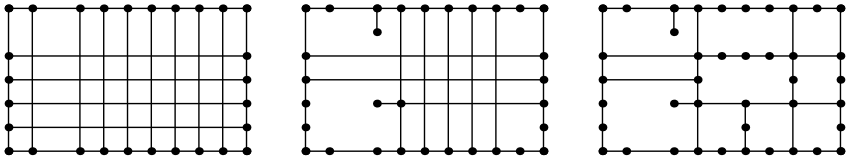

Fig. 5. Blocks: (a) a no-occurrence block — used if the variable does not occur in the clause; (b) a positive-occurrence block — used if the variable has a positive occurrence in the clause; (c) a negative-occurrence block — used if the variable has a negative occurrence in the clause

In addition, G contains a clause line and a variable line for each clause and each variable, respectively. The clause line for clause c_j starts at position $(2, 0)$ of the last block in the j-th column and ends at position $(2, 6)$ of the first block in this column. It contains $k + 2$ vertical segments, where k is the number of literals it contains. Hence, it is a vertical line spanning the whole grid with $k+1$ internal points on it. The variable line for variable x_i starts at position $(0, 5)$ of the first block in the i-th row and ends at position $(10, 5)$ of the last block in this row. It contains the following segments RDRURDRUR (two "bumps").

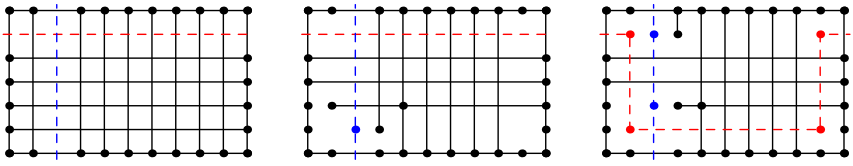

Fig. 6. (a) The clause and variable lines passing through the no-occurrence block at position (i, j), i.e., variable x_i has no occurrence in c_j. Note that if the block is in the first column (last column, first row, last row) then the variable (variable, clause, clause) line is attached to the left (right, top, bottom) of the frame. (b-c) The clause and variable lines passing through the positive-occurrence block at position (i, j), i.e., variable x_i has a positive occurrence in c_j.

Now, let us analyze how the clause line for c_j and the variable line for x_i can pass through the block at position (i, j). This will depend on the type of the block. For the no-occurrence block depicted in Figure 5(a) there is only one way that the clause and variable lines can pass through the block; see Figure 6(a). Note that no internal vertex can be placed on the clause line and no bump can occur on the variable line inside this block.

For the positive-occurrence block depicted in Figure 5(b) there are two ways that the clause and variable lines can pass through the block. They are depicted in Figure 6(b)–(c). Note that the variable line can contain at most one bump inside this block. In the case where it contains no bump (passes through directly), at most one internal vertex can be placed on the clause line inside this block. If it contains a bump, then at most two internal vertices can be placed on the clause line inside this block. Figure 6(b)–(c) depicts the possibilities with the maximal number of internal vertices on the clause line.

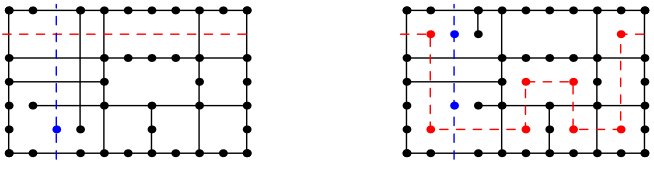

Fig. 7. The clause and variable lines passing through the negative-occurrence block at position (i, j), i.e., variable x_i has a negative occurrence in c_j

For the negative-occurrence block depicted in Figure 5(c) there are two ways that the clause and variable lines can pass through the block. They are depicted in Figure 7. Note that the variable line contains either zero or two bumps inside this block. In the case where it contains no bump (passes through directly), similarly as for the positive-occurrence block, at most one internal vertex can be placed on the clause line inside this block. If it contains two bumps, then

at most two internal vertices can be placed on the clause line inside this block. Figure 7 depicts the possibilities with the maximal number of internal vertices on the clause line.

Now, we are ready to show that the constructed graph has a rectilinear drawing if and only if the 3SAT(3) instance is satisfiable. First, let us consider the variable lines. Each variable line contains two bumps. Since the variable line for x_i cannot make any bumps in the no-occurrence blocks, it must make bumps in the remaining three blocks. Let c_{j_1}, c_{j_2} (c_{j_3}) be the clauses containing a positive (negative) occurrence of x_i. The variable line for x_i can contain zero or one bump in the blocks at positions (i, j_1) and (i, j_2), and zero or two bumps in the block at position (i, j_3). It follows that we have two mutually-exclusive possibilities: either the variable line makes bumps in its positive-occurrence blocks (one in each of them) or it contains two bumps in its negative-occurrence block. In the first case, we set the value of variable x_i to *true*, in the second case, to *false*.

Next, we show that each clause is satisfied in the constructed assignment. Recall that each clause line contains $k+1$ internal vertices, where k is the number of literals in the clause. Consider a clause c_j. Since the clause line cannot contain any internal vertices inside the no-occurrence blocks it passes through, the $k+1$ internal points have to appear in the remaining k blocks. It follows that in at least one of them the clause line will contain two internal vertices. However, by the above analysis and definition of the assignment, the corresponding literal must be set to *true*, hence, clause c_j is satisfied.

Given a truth assignment for the 3SAT(3) instance, we can construct a rectilinear drawing for the graph as follows. First, for the variable line for x_i, we place the bumps in the blocks at position (i, j), where c_j contains an occurrence of the variable x_i and this occurrence (positive or negative) has value *true*. Second, for the clause line for c_j, we place two internal vertices on the clause line inside the block at position (i, j), where c_j contains an (positive or negative) occurrence of x_i which makes c_j satisfied. We might have several choices for i, but we pick one of them. Then we place one internal vertex on the clause line in the remaining blocks at positions (i', j), where c_j contains an occurrence of $x_{i'}$ and $i' \neq i$. Now, each variable line contains exactly two bumps and each clause line for a k-clause contains exactly $k+1$ internal vertices, hence, we have a rectilinear drawing of the graph. □

4 Conclusions

Previous work has shown the problem of finding non-planar rectilinear drawings for graphs to be NP-complete [2,4]. In this work, we have resolved the complexity of a number of natural restrictions where constraints are placed on the possible orientations of edges. In particular, we show that determining the existence of a non-planar rectilinear drawing for a graph when directions are prescribed to each edge is polynomial, while determining a minimum-area drawing for the same case is NP-complete. When edges are prescribed to be either horizontal or vertical, or when a cyclic order of the incedent edges around each vertex is prescribed,

we show that the existence problem, and thus the area-minimization problem, is NP-complete. Finally, we have shown the NP-complete existence problems to be fixed parameter tractable in the number of vertices of degree 3 or 4.

It remains open whether or not the corresponding minimum-area drawing cases are also fixed parameter tractable. Since there are polynomial time algorithms for finding planar rectilinear drawings for several different classes of maximum degree 3 graphs [13,11,12], it would be interesting to find such classes for the non-planar case. Further interesting open questions are the complexity of finding planar rectilinear drawings of HV-restricted and cyclic-restricted graphs.

References

1. Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. In: Dehne, F., Gavrilova, M., Sack, J.-R., Tóth, C.D. (eds.) WADS 2009. LNCS, vol. 5664, pp. 206–217. Springer, Heidelberg (2009)
2. Eades, P., Hong, S.-H., Poon, S.-H.: On rectilinear drawing of graphs. In: Eppstein, D., Gansner, E.R. (eds.) GD 2009. LNCS, vol. 5849, pp. 232–243. Springer, Heidelberg (2010)
3. Eppstein, D.: The topology of bendless three-dimensional orthogonal graph drawing. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 78–89. Springer, Heidelberg (2009)
4. Formann, M., Hagerup, T., Haralambides, J., Kaufmann, M., Leighton, F.T., Symvonis, A., Welzl, E., Woeginger, G.: Drawing graphs in the plane with high resolution. SIAM Journal on Computing 22(5), 1035–1052 (1993)
5. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. SIAM Journal of Computing 31(2), 601–625 (2001)
6. Hoffman, F., Kriegel, K.: Embedding rectilinear graphs in linear time. Information Processing Letters 29(2), 75–79 (1988)
7. Huang, W., Hong, S., Eades, P.: Effects of crossing angles. In: Proc. of IEEE Pacific Visualization Symposium (PacificVis 2008), pp. 41–46 (2008)
8. Opatrny, J.: Total ordering problem. SIAM Journal of Computing 8(1), 111–114 (1979)
9. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
10. Patrignani, M.: On the complexity of orthogonal compaction. Computational Geometry 19, 47–67 (2001)
11. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 387–392. Springer, Heidelberg (2004)
12. 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)
13. Rahman, M.S., Naznin, M., Nishizeki, T.: Orthogonal drawings of plane graphs without bends. Journal of Graph Algorithms and Applications 7(4), 335–362 (2003)
14. Vijayan, G., Wigderson, A.: Rectilinear graphs and their embeddings. SIAM Journal of Computing 14(2), 355–372 (1985)