Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 21, no. 4, pp. 709–730 (2017)
DOI: 10.7155/jgaa.00436

On the Total Number of Bends for Planar Octilinear Drawings

Michael A. Bekos [1] *Michael Kaufmann* [1] *Robert Krug* [1]

[1] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany

Abstract

An *octilinear drawing* of a planar graph is one in which each edge is drawn as a sequence of horizontal, vertical and diagonal at 45° and −45° line-segments. For such drawings to be readable, special care is needed in order to keep the number of bends small. Since the problem of finding planar octilinear drawings with minimum number of bends is NP-hard, in this paper we focus on upper and lower bounds. From a recent result of Keszegh et al. on the slope number of planar graphs, we can derive an upper bound of $4n-10$ bends for planar graphs with n vertices and maximum degree 8. We considerably improve this general bound and corresponding previous ones for triconnected planar graphs of maximum degree 4, 5 and 6. We also derive non-trivial lower bounds for these three classes of graphs by a technique inspired by the network flow formulation of Tamassia for computing bend optimal orthogonal drawings.

Submitted: January 2016	Reviewed: July 2016	Revised: August 2016	Reviewed: March 2017	Revised: May 2017
	Accepted: June 2017	Final: June 2017	Published: June 2017	
	Article type: Regular paper		Communicated by: H. Meijer	

This work has been supported by DFG grant Ka812/17-1. The article is based on the preliminary version [3].
E-mail addresses: bekos@informatik.uni-tuebingen.de (Michael A. Bekos) mk@informatik.uni-tuebingen.de (Michael Kaufmann) krug@informatik.uni-tuebingen.de (Robert Krug)

1 Motivation and Background

Octilinear drawings of graphs have a long history of research, which dates back to the early thirties of the last century, when an English technical draftsman, Henry Charles Beck (also known as Harry Beck), designed the first schematic map of London Underground. His map, the so-called *Tube map*, looked more like an electrical circuit diagram (consisting of horizontal, vertical and diagonal line segments) rather than a true map, as the underlying geographic accuracy was neglected. Laying out networks in such a way is called *octilinear graph drawing* and plays an important role in map-schematization and the design of metro-maps. In particular, an octilinear drawing $\Gamma(G)$ of a graph $G = (V,E)$ is one in which each vertex occupies a point on an integer grid and each edge is drawn as a sequence of horizontal, vertical and diagonal at 45° line-segments. As a result, every vertex has eight available so-called *ports*, where its incident edges can be connected to. When G is planar, usually $\Gamma(G)$ is required to be planar.

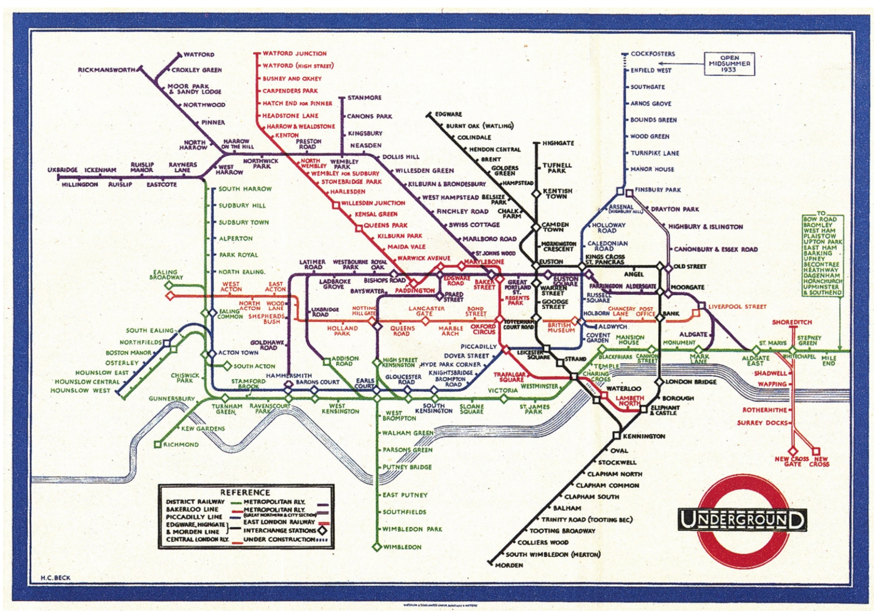

Figure 1: Henry Beck Tube Map (first edition), 1933. Printed at Waterlow & Sons Ltd., London.

In planar octilinear graph drawing, an important goal is to keep the number of bends small, so that the produced drawings can be understood easily. One can derive a non-trivial upper bound on the required number of bends from a result on the *planar slope* number of graphs by Keszegh et al. [15], who proved that every planar graph of maximum degree k has a planar drawing with at most $\lceil \frac{k}{2} \rceil$ different slopes in which each edge has at most two bends. For $3 \leq k \leq 8$, the drawings produced by the algorithm of Keszegh et al. are octilinear, which yields an upper bound of $6n-12$ on the total number of bends, where n is the number of vertices of the graph. This bound can be reduced to $4n-10$ with some effort; see Section 1.1.

Table 1: A short summary of our results.

Graph class (planar)	Lower bound	Ref.	Upper bounds			
			Previous	Ref.	New	Ref.
3-con. max.deg-4	$n/3-1$	Th. 4	$2n$	[2]	$n+5$	Th. 1
3-con. max.deg-5	$2n/3-2$	Th. 4	$5n/2$	[2]	$2n-2$	Th. 2
3-con. max.deg-6	$4n/3-6$	Th. 4	$4n-10$	[15]	$3n-8$	Th. 3

On the other hand, it is known that every planar graph of maximum degree 3 with five or more vertices admits a planar octilinear drawing in which all edges are bend-less [14, 8]. So, a natural question to ask is whether a planar graph of maximum degree $3+k$ can be drawn with $kn+O(1)$ bends. Recently, it was proved that all planar graphs of maximum degree 4 and 5 admit planar octilinear drawings with at most one bend per edge [2]. This result along with the degree sum formula implies that the total number of bends for such graphs can be upper bounded by $2n$ and $5n/2$, respectively. Note that the problem of determining whether a given embedded planar graph of maximum degree 8 admits a bend-less planar octilinear drawing is NP-complete [17].

The remainder of this paper is organized as follows. In Section 2, we considerably improve all aforementioned bounds for the classes of triconnected planar graphs of maximum degree 4, 5 and 6. In Section 3, we present corresponding lower bounds for these three classes of planar graphs. We conclude in Section 4 with open problems and future work. For a summary of our results also refer to Table 1.

1.1 Related work

As already stated, Keszegh et al. [15] have proved that every planar graph of maximum degree k admits a planar drawing with at most $\lceil \frac{k}{2} \rceil$ different slopes in which each edge has at most two bends. If one appropriately adjusts the slopes of all edge segments incident to a vertex, then one can show that any planar graph of maximum degree k, with $3 \leq k \leq 8$, admits a planar octilinear drawing in which each edge has at most two bends. This implies that any planar graph of maximum degree k on n vertices can have at most $6n-12$ bends, where $3 \leq k \leq 8$. One can easily improve this bound to $4n-10$ as follows. The edge that "enters" a vertex from its south port and the edge that "leaves" each vertex from its top port in the *s*-*t* ordering of the algorithm of Keszegh et al. can both be drawn with only one bend each. Since each vertex is incident to exactly two such edges (except for the first and last ones in the *s*-*t* ordering which are only incident to one such edge each), it follows that $2n-2$ edges can be drawn with at most one bend. Hence, the bound of $4n-10$ follows.

Octilinear drawings form a natural extension of the so-called *orthogonal drawings*, which allow for horizontal and vertical edge segments only. For such drawings, Tamassia [20] showed that the bend minimization problem can be solved efficiently, assuming that the input is an embedded graph. However, the corresponding minimization problem over all embeddings of the input graph is NP-hard [11]. Tamassia's algorithm [20]

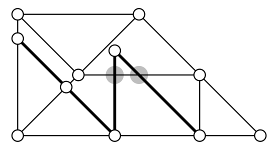

Figure 2: An octilinear drawing of a planar graph of maximum degree 4. The two intersections marked by gray circles are unavoidable when drawing the graph with the given angles. Note that these angles can be specified by a bend-optimal octilinear representation; refer also to [17].

consists of two main phases. In the first phase, a so-called *orthogonal representation* is computed, which in a sense captures the "shape" of the drawing, neglecting the exact geometry underneath. Formally, an orthogonal representation describes the sequence of bends along each edge and the angles (as multiples of 90°) between edges incident to a common vertex. In the second phase, based on the computed representation the actual coordinates for the vertices and edge-bends are computed. Note that, given an embedded planar graph of maximum degree 8, Tamassia [20] describes how one can extend his approach, so to compute a bend-optimal octilinear representation (in which the angles are multiples of 45°). However, such a representation may not be realizable by a corresponding planar octilinear drawing [6]; see Figure 2 for an example.

For orthogonal drawings, several bounds on the total number of bends are known. Biedl [4] presents lower bounds for graphs of maximum degree 4 based on their connectivity (i.e., simply connected, biconnected or triconnected), planarity (i.e., planar or not) and simplicity (i.e., simple or non-simple with multiedges or selfloops). It is also known that any planar graph of maximum degree 4 (except for the octahedron graph) admits a planar orthogonal drawing with at most two bends per edge [5, 16]. Trivially, this yields an upper bound of $4n$ bends, which can be improved to $2n+2$ [5]. Note that the best known lower bound is due to Tamassia et al. [21], who presented planar graphs of maximum degree 4 requiring $2n-2$ bends.

Finally, in metro-map visualization many approaches have been proposed that result in (nearly-)octilinear drawings (see, e.g., [12, 17, 18, 19]). However, most of them are heuristics and therefore do not focus on the bend-minimization problem explicitly.

1.2 Preliminaries

Central in our approach is the canonical order [7], which exists for any triconnected planar graph and it can be computed in linear time [13]. Since we assume familiarity with this concept, we describe only the parts of the concept which are most essential for our work. Let $G=(V,E)$ be a triconnected planar graph and let $\Pi=(P_0,\dots,P_m)$ be a partition of V into paths, such that $P_0=\{v_1,v_2\}$, $P_m=\{v_n\}$ and vertices v_2, v_1 and v_n form a path on the outerface of G. For $k=0,1,\dots,m$, let G_k be the subgraph induced by $\cup_{i=0}^{k}P_i$ and denote by C_k the outer face of G_k. Partition Π is a *canonical order* of G if for each $k=1,\dots,m-1$ the following hold (see Figure 3):

(i) G_k is biconnected,

(ii) all neighbors of $P_k \subset C_k$ in G_{k-1} are on the outer face of G_{k-1}

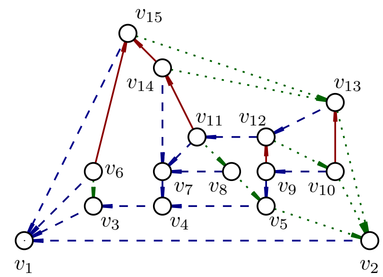

Figure 3: An example illustrating a canonical order and the corresponding direction and coloring of the edges. The graph is of maximum degree 5 (see vertex v_{13}). The canonical order consists of the following partitions: $P_0 = \{v_1, v_2\}$, $P_1 = \{v_3, v_4, v_5\}$, $P_2 = \{v_6\}$, $P_3 = \{v_7, v_8\}$, $P_4 = \{v_9, v_{10}\}$, $P_5 = \{v_{11}\}$, $P_6 = \{v_{12}\}$, $P_7 = \{v_{13}\}$, $P_8 = \{v_{14}\}$ and $P_9 = \{v_{15}\}$.

(iii) $|P_k| = 1$ or the degree of each vertex $v \in P_k$ is two in G_k and

(iv) all vertices of P_k, $0 \leq k < m$ have at least one neighbor in P_j for some $j > k$.

Path P_k is called *singleton* if $|P_k| = 1$ and *chain* otherwise. Note that if P_k is a chain, then each of the two endpoints of P_k are adjacent to G_{k-1} by only one edge.

To simplify the description of our algorithms, we direct and color the edges of G based on partition Π (similar to Schnyder colorings [9]) as follows. The first partition P_0 of Π exclusively defines one edge (that is, edge (v_1, v_2)), which we color blue and direct towards vertex v_1. For each partition $P_k = \{v_i, \ldots, v_{i+j}\} \in \Pi$ later in the order, let $v_1 = c_1, c_2, \ldots, c_q = v_2$ be the vertices on C_{k-1} as they appear from left to right and let v_ℓ and v_r be the leftmost and rightmost neighbors of P_k in G_{k-1}, respectively. In the case where P_k is a chain (that is, $j > 0$), we color edge (v_i, v_ℓ) and all edges between vertices of P_k blue and direct them towards v_ℓ, i.e., each edge (v_{i+p-1}, v_{i+p}) of P_k, $p = 1, \ldots, j$, is directed from v_{i+p} to v_{i+p-1}. The edge (v_{i+j}, v_r) is colored green and is directed from v_{i+j} to v_r. In the case where P_k is a singleton (that is, $j = 0$), we color the edges (v_i, v_ℓ) and (v_i, v_r) blue and green, respectively and we direct them towards v_ℓ and v_r. We color the remaining edges incident to P_k towards G_{k-1} (if any) red and we direct them towards v_i.

Given a vertex $v \in V$ of G, we denote by $\text{indeg}_x(v)$ ($\text{outdeg}_x(v)$, respectively) the in-degree (out-degree, respectively) of vertex v in color $x \in \{r, b, g\}$. Observe that for a vertex $v \in V \setminus \{v_1\}$, $\text{outdeg}_b(v) = 1$, which implies that the blue subgraph is a spanning tree of G. Similarly $0 \leq \text{outdeg}_g(v)$, $\text{outdeg}_r(v) \leq 1$. Hence the green and the red subgraphs form two forests of G. It also holds that $0 \leq \text{indeg}_b(v)$, $\text{indeg}_g(v)$, $\text{indeg}_r(v) \leq d(G) - 1$, where $d(G)$ is the maximum vertex-degree of G.

2 Upper Bounds

In this section, we present upper bounds on the total number of bends for the classes of triconnected planar graphs of maximum degree 4, 5 and 6.

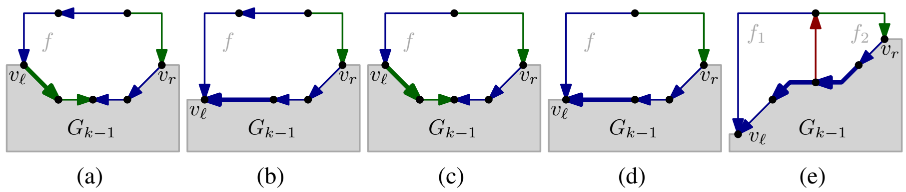

Figure 4: Illustration of the reference edge (bold) in the case of: (a-b) a chain, (c-d) a singleton of degree 2 in G_k and (e) a singleton of degree 3 in G_k.

2.1 Triconnected Planar Graphs of Maximum Degree 4

Let $G = (V, E)$ be a triconnected planar graph of maximum degree 4. Before we proceed with the description of our approach, we need to define the notions of vertical and horizontal cuts; e.g., see [10]. A *vertical cut* is a y-monotone continuous curve that crosses only horizontal line-segments of a drawing and splits it into a left and a right part (refer to the dotted curve in Figure 5f). Such a cut makes a drawing horizontally stretchable in the following sense: One can shift the right part of the drawing that is defined by the vertical cut further to the right, while keeping the left part of the drawing in place and the result is a valid octilinear drawing. Similarly, one can define a *horizontal cut*.

Since G has maximum degree 4, it has at most $2n$ edges. By Euler's formula, it follows that G has at most $n+2$ faces. In order to construct a drawing $\Gamma(G)$ of G, which has roughly at most $n+2$ bends, we also need to associate to each face of G a so-called *reference edge*. The reference edge of a face is defined as follows. Let $\Pi = \{P_0, \ldots, P_m\}$ be a canonical order of G and assume that $\Gamma(G)$ is constructed incrementally by placing a new partition of Π each time, so that the upper envelope of the drawing constructed so far is an x-monotone path. When placing a new partition $P_k \in \Pi$, $k = 1, \ldots, m-1$, one or two bounded faces of G are formed (note that we treat the last partition P_m of Π, which might introduce three bounded faces, separately). More precisely, if P_k is a chain or a singleton of degree 2 in G_k, then only one bounded face is formed. Otherwise (that is, P_k is a singleton of degree 3 in G_k), two new bounded faces are formed. In both cases, each newly-formed bounded face consists of at least two edges incident to vertices of P_k and at least one edge of G_{k-1}. In the former case, the reference edge of the newly-formed bounded face, say f, is defined as follows. If f contains at least one green edge that belongs to G_{k-1}, then the reference edge of f is the leftmost such edge (see Figures 4a and 4c). Otherwise, the reference edge of f is the leftmost blue edge of f that belongs to G_{k-1} (see Figure 4b and 4d). In the case where P_k is a singleton of degree 3 in G_k, the reference edge of each of the newly formed faces is the edge of G_{k-1} that is incident to the endpoint of the red edge involved. Observe that, by definition, a red edge cannot be a reference edge.

As already stated, we will construct $\Gamma(G)$ in an incremental manner by placing one partition of Π at a time (refer to Figure 5 for an example). For the base case, we momentarily neglect the edge (v_1, v_2) of the first partition P_0 of Π and we start by placing the second partition, say a chain $P_1 = \{v_3, \ldots, v_{|P_1|+2}\}$, on a horizontal line from left to right. Since by definition of Π, v_3 and $v_{|P_1|+2}$ are adjacent to the two vertices,

v_1 and v_2, of the first partition P_0 of Π, we place v_1 to the left of v_3 and v_2 to the right of $v_{|P_1|+2}$. So, from left to right they form a path where all edges are drawn using horizontal line-segments that are attached to the east and west ports at their endpoints. The case where P_1 is a singleton is analogous (assuming that P_1 is a chain of zero length). Assume now that we have already constructed a drawing for G_{k-1}, which has the following invariant properties:

IP-1: The number of edges of G_{k-1} with a bend is at most equal to the number of reference edges in G_{k-1}.

IP-2: The north-west, north and north-east ports of each vertex are occupied by incoming blue/green edges and by outgoing red edges. Accordingly, the south-west, south and south-east ports of each vertex are occupied by outgoing blue and green edges and by incoming red edges. Additionally, each vertex that lies on the outerface of G_{k-1} has at least two consecutive northern ports unoccupied.

IP-3: If a horizontal port of a vertex v is occupied, then it is occupied either by an edge with a bend (to support vertical cuts) or by an edge of a chain containing v.

IP-4: A red edge is not on the outerface of G_{k-1}.

IP-5: A blue (green, respectively) edge of G_{k-1} is never incident to the north-west (north-east, respectively) port of a vertex of G_{k-1}.

IP-6: From each reference edge on the outerface of G_{k-1} one can devise a vertical cut through the drawing of G_{k-1}, i.e., each reference edge on the outerface of G_{k-1} has a horizontal segment.

The base case of our algorithm conforms with the aforementioned invariant properties. In the following, we will show how to add the next partition P_k with $k < m$, so that all aforementioned invariant properties are fulfilled. In our description, we will mainly describe the port assignment at each vertex that will always conform to IP-2–5, which fully specifies how each edge must be drawn (in other words, we describe the relative coordinates of the vertices). The exact coordinates can then be computed by adopting an approach similar to the one of Bekos et al. [2], since the *base* of each newly formed face (i.e., the part of the face that is not formed by newly introduced edges) is horizontally stretchable (follows from IP-6). Next, we consider the three main cases.

C.1: $P_k = \{v_i\}$ *is a singleton of degree* 2 *in* G_k; see Figures 6a and 6b. Let v_ℓ and v_r be the leftmost and rightmost neighbors of v_i in G_{k-1} (note that v_ℓ and v_r are not necessarily neighboring). We claim that the north-east port of v_ℓ and the north-west port of v_r cannot be simultaneously occupied. For a proof by contradiction, assume that the claim does not hold. Denote by $v_\ell \rightsquigarrow v_r$ the path from v_ℓ to v_r at the outerface of G_{k-1} (neglecting the direction of the edges implied by the canonical order Π). By IP-5, $v_\ell \rightsquigarrow v_r$ starts as blue from the north-east port of v_ℓ and ends as green at the north-west port of v_r. So, in between there is a vertex of the path $v_\ell \rightsquigarrow v_r$, which is incident to both a blue outgoing edge and a green outgoing edge. So, this vertex must have a neighbor in P_j for some $j \geq k$; a contradiction to the degree of v_i.

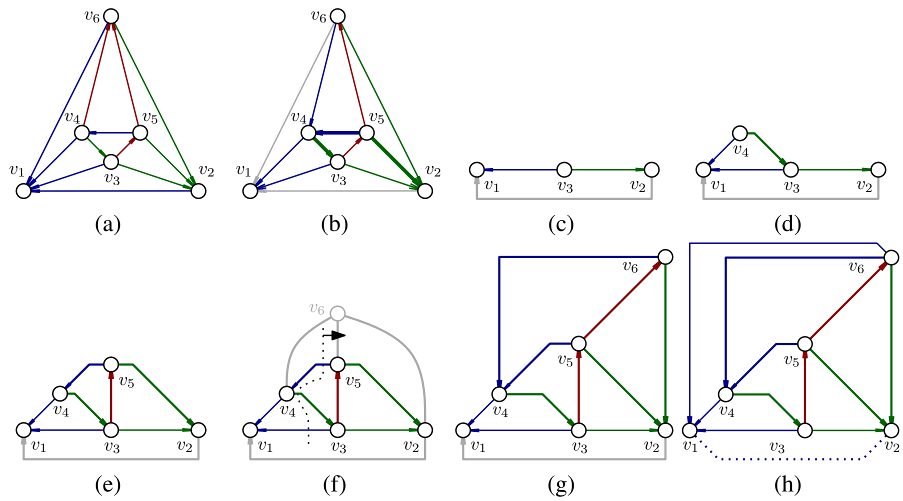

Figure 5: An illustration of our algorithm for triconnected planar graphs of maximum degree 4 by an example: the octahedron graph. The underlying canonical order consists of the following partitions: $P_0 = \{v_1, v_2\}$, $P_1 = \{v_3\}$, $P_2 = \{v_4\}$, $P_3 = \{v_5\}$ and $P_4 = \{v_6\}$. (a) The direction and the coloring of the edges. (b) The corresponding reference edges (bold); the edge (v_1, v_2) of the first partition and the edge (v_1, v_6) incident to the last (degree 4) partition are ignored. (c) The placement of the first two partitions. (d) The placement of a singleton of degree 2 incident to reference edge (v_4, v_3) that is drawn bent. (e) The placement of a singleton of degree 3 incident to reference edges (v_5, v_4) and (v_5, v_2) that are drawn bent. (f) The last singleton v_6 is not incident to reference edges. So, (v_6, v_4), (v_5, v_6) and (v_6, v_2) must be drawn bend-less. (g) Vertex v_5 is translated upwards until one of the horizontal line-segments incident to v_5 is eliminated. (h) The final layout containing (v_2, v_1) and (v_6, v_1); the dotted edge can be drawn with a single bend.

Without loss of generality, assume that the north-east port of v_ℓ is unoccupied. In order to draw the edge (v_i, v_ℓ), we distinguish two cases. If edge (v_i, v_ℓ) is the reference edge of a face, then we draw edge (v_i, v_ℓ) as a horizontal-diagonal combination from the west port of v_i towards the north-east port of v_ℓ. Otherwise, edge (v_i, v_ℓ) is drawn bend-less from the south-west port of v_i towards the north-east port of v_ℓ. To draw the second edge incident to v_i (that is, the edge (v_i, v_r)), again we distinguish two cases. If the north-west port at v_r is unoccupied, then edge (v_i, v_r) will use this port at v_r. Otherwise, edge (v_i, v_r) will use the north port at v_r. In addition, if edge (v_i, v_r) is the reference edge of a face, then edge (v_i, v_r) will use the east port at v_i. Otherwise, edge (v_i, v_r) will use either the south or the south-east port at v_i depending on whether edge (v_i, v_r) uses the north or the north-west port at v_r, respectively.

The port assignment described above conforms to IP-2–5. Clearly, IP-1 also holds. IP-6 holds because the newly introduced edges that are reference edges have a horizontal line-segment, which inductively implies that vertical cuts through them are possible.

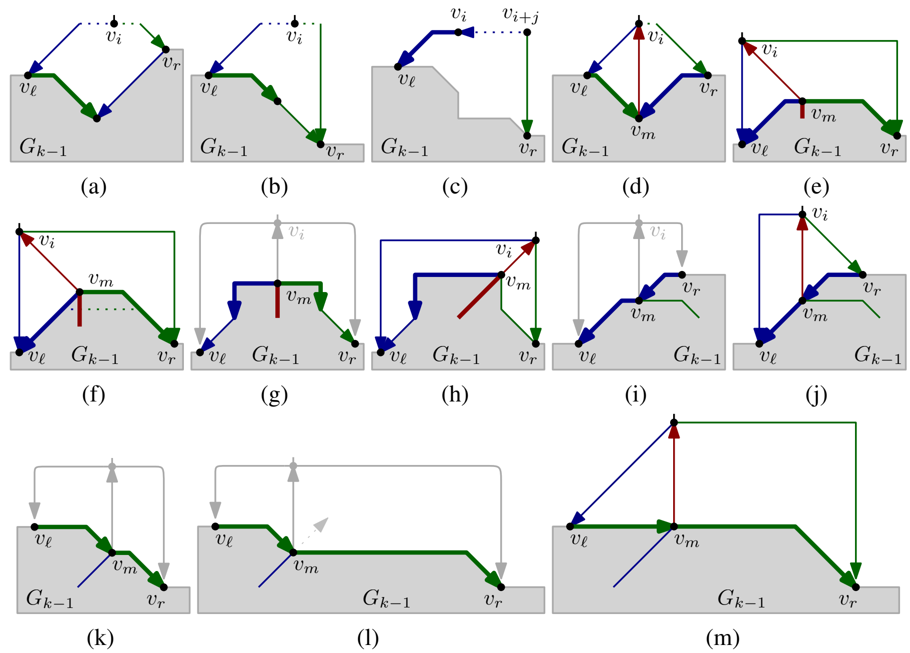

Figure 6: Illustration of: (a-b) the case of a degree-2 singleton in G_k, (c) the case of a chain, (d-m) the case of a singleton of degree 3 in G_k (dotted line-segments can have zero length).

C.2: $P_k = \{v_i, \ldots v_{i+j}\}$ *with* $j \geq 1$ *is a chain*; see Figure 6c. This case is similar to case C.1, as P_k has also exactly two neighbors in G_{k-1} (which we again denote by v_ℓ and v_r). The edges between $v_i, \ldots, v_{i+j}$ will be drawn as horizontal line-segments connecting the west and east ports of the respective vertices. The edges (v_i, v_ℓ) and (v_{i+j}, v_r) are drawn based on the rules of the case C.1 (e.g., in Figure 6c edge (v_i, v_ℓ) is a reference edge, while the edge (v_{i+j}, v_r) is not). Hence, the port assignment still conforms to IP-2–5. In addition, IP-1 and IP-6 hold, since all edges of the chain are horizontal.

C.3: $P_k = \{v_i\}$ *is a singleton of degree* 3 *in* G_k. Let v_ℓ and v_r be the leftmost and rightmost neighbors of v_i in G_{k-1} and let v_m be the third neighbor of v_i in G_{k-1}. By IP-2 and the degree restriction, the north port of v_m is unoccupied. If the north-east port of v_ℓ and the north-west port of v_r are simultaneously unoccupied, we proceed analogously to case C.1; see Figure 6d. Clearly, IP-1 and IP-6 hold. Consider now the more involved case, where the north-east port of v_ℓ is occupied and simultaneously (v_i, v_ℓ) is not a reference edge, that is, (v_i, v_ℓ) must be drawn bend-less (the case where the north-west port of v_r is occupied and simultaneously (v_i, v_r) is not a reference edge is analogous; we will only detail the deferences in the following). Since the north-east port at v_ℓ is occupied, by IP-4 it follows that the edge at the north-east port of v_ℓ is not red. Therefore, by IP-2 and IP-5, the edge at the north-east port of v_ℓ is blue. This implies that the path $v_\ell \rightsquigarrow v_m$ at the outerface of G_{k-1} consists of exclusively blue edges pointing towards v_ℓ. Hence, by IP-5 the north-west port at v_m is unoccupied. Edge (v_i, v_ℓ) can be drawn bend-less if the edge (v_i, v_r) is a reference edge (that is, by IP-6 (v_i, v_r) has a bend); see Figure 6e. In the case where the edge (v_i, v_r) is not a reference edge (that is, none of (v_i, v_ℓ) and (v_i, v_r) is a reference edge), we need a different argument. We further distinguish two sub-cases.

C.3.1: *The edge incident to v_m on the path $v_m \rightsquigarrow v_r$ on the outerface of G_{k-1} is green.* By definition, the blue (green) edge of $v_\ell \rightsquigarrow v_m$ ($v_m \rightsquigarrow v_r$) incident to v_m is a reference edge and by IP-6 has a bend. Our aim is to "eliminate" one of these bends and draw one of the edges (v_i, v_ℓ) or (v_i, v_r) with a bend and the other one bend-less, such that IP-1 holds. In this case, v_m may or may not be incident to another red edge in G_{k-1} (equivalently, v_m is either of degree 4 or 3, respectively). Without loss of generality, we assume that v_m is incident to another red edge, say (v_m, v'_m), in G_{k-1}, that is, v_m is of degree 4. In this case, we translate v_m upwards in the direction implied by the slope of the edge (v_m, v'_m), until one of the horizontal line-segments of the edges incident to v_m on the outerface of G_{k-1} is completely eliminated; see Figure 6f. The only case, where the aforementioned segment elimination is not possible, is when (v_m, v'_m) is vertical and the edges incident to v_m at the outerface of G_{k-1} are both horizontal-vertical combinations; see Figure 6g. In this particular case, however, by IP-2 it follows that either the north-west or the north-east port at v'_m is free. Since both edges incident to v_m at the outerface of G_{k-1} are bent, by IP-3 we can redraw (v'_m, v_m) so to be diagonal and then we proceed similarly to the previous case; see Figure 6h. Note that this is a local modification, which is not propagated any further. Also, observe that the port assignment still conforms to IP-2–IP-5.

C.3.2: *The edge incident to v_m on the path $v_m \rightsquigarrow v_r$ on the outerface of G_{k-1} is blue.* In this case, v_m cannot be incident to another red edge. In the case where v_m is of degree 3, we proceed similar to the case C.3.1, where v_m was of degree 3. So, we now focus on the case where v_m is of degree 4. In this case, the fourth edge attached to v_m can be either green outgoing or blue incoming. In the former case, this edge is a reference edge. In the latter case, it is part of a chain. In both cases, however, this edge has a horizontal line-segment; see Figure 6i. Hence, we can translate v_m horizontally to the left so to eliminate the bend of the edge incident to v_m on the path $v_\ell \rightsquigarrow v_m$; see Figure 6j. Clearly, all invariant properties are fulfilled once v_i is drawn.

Now, recall that in the case where the fourth edge attached to v_m was green outgoing, we could guarantee horizontal stretchability, because this edge was a reference edge. This leaves one case that due to symmetry cannot be covered. More precisely, when both edges incident to v_m on the outerface of G_{k-1} are green, the north-west port at v_r is occupied and the fourth edge attached to v_m is blue, we can no longer guarantee that the later edge is reference (see Figure 6k). Hence, if both edges (v_i, v_ℓ) and (v_i, v_r) are not reference (that is, bend-less), then we need a different argument. Recall that both green edges that are incident to v_m on the outerface of G_{k-1} are reference edges. Therefore, each one must have a bend. We horizontally stretch the drawing such that the length of the horizontal line-segment of the green edge attached to the west port of v_m is greater than the vertical distance between v_ℓ and v_m; see Figure 6l. This allows us to translate v_m diagonally-up, so to eliminate the diagonal line-segment of the green edge incident to the west port of v_m; see Figure 6m. So, IP-1 still holds.

Note that the coordinates of the newly introduced vertices are determined by the shape of the edges connecting them to G_{k-1}. If there is not enough space between v_ℓ and v_r to accommodate the new vertices, IP-6 allows us to stretch the drawing horizontally using the reference edge of the newly formed face.

To complete the description of our algorithm, it remains to cope with the last partition $P_m = \{v_n\}$ and describe how to draw the edge (v_1, v_2) of the first partition P_0 of Π. If v_n is of degree 3, we cope with P_m as being an ordinary singleton. However, if v_n is of degree 4, then we momentarily ignore the edge (v_n, v_1) of P_m and proceed to draw the remaining edges incident to v_n, assuming that P_m is again an ordinary singleton. The edge (v_n, v_1) can be drawn afterwards using two bends in total (see Figure 5h). Finally, since by construction v_1 and v_2 are horizontally aligned, we can draw the edge (v_1, v_2) with a single bend, emanating from the south-east port of v_1 towards the south-west port of v_2.

Theorem 1 *Let G be a triconnected planar graph of maximum degree* 4 *with n vertices. A planar octilinear drawing* $\Gamma(G)$ *of G with at most* $n+5$ *bends can be computed in* $O(n)$ *time.*

Proof: By IP-1, all bends of $\Gamma(G)$ are in correspondence with the reference edges of G, except for the bends of (v_1, v_2) and (v_n, v_1). Since the number of reference edges is at most $n+2$ and the edges (v_1, v_2) and (v_n, v_1) require 3 additional bends, the total number of bends of $\Gamma(G)$ does not exceed $n+5$. The linear running time follows from the observation that we can use the shifting method of Kant [14] to compute the actual coordinates of the vertices of G (assuming the real RAM model of computation). This is because in the canonical order the y-coordinates of the vertices that have been placed at some particular step do not change in subsequent steps. The only exceptional case is the one of two green edges incident to a singleton of degree 4 (discussed in case C.3.2). Note however that this particular case does not influence the overall running time, since it can occur at most once per vertex. A similar approach is also discussed in [2]. □

2.2 Triconnected Planar Graphs of Maximum Degree 5

Our algorithm for triconnected planar graphs of maximum degree 5 is an extension of the corresponding algorithm of Bekos et al. [2], which computes for a given triconnected planar graph G of maximum degree 5 on n vertices a planar octilinear drawing $\Gamma(G)$ of G with at most one bend per edge. Since G cannot have more than $5n/2$ edges, it follows that the total number of bends of $\Gamma(G)$ is at most $5n/2$. However, before we proceed with the description of our extension, we first provide some insights into this algorithm, which is based on a canonical order Π of G. In this algorithm central are IP-2 and IP-4 of the previous section and the so-called *stretchability invariant*, which is a simpler version of IP-3 of the previous section. More precisely, according to this invariant, all edges on the outerface of the drawing constructed at some step of the canonical order have a horizontal line-segment and therefore one can devise corresponding vertical cuts to horizontally stretch the drawing. We claim that we can appropriately modify this algorithm, so that all red edges of G are bend-less.

Since we seek to draw all red edges of G bend-less, our modification is limited to singletons. So, let $P_k = \{v_i\}$ be a singleton of Π. The degree restriction implies that v_i has at most two incoming red edges (we also assume that P_k is not the last partition of Π, that is $k \neq m$). We first consider the case where v_i has exactly one incoming red edge, say $e = (v_j, v_i)$, with $j < i$. By construction, e must be attached to one of the northern ports of v_j (that is, north-west, north or north-east). On the other hand, e can be attached to any of the southern ports of v_i, as e is its only incoming red edge [2]. This guarantees that e can be drawn bend-less.

Consider now the more involved case, where v_i has exactly two incoming red edges, say $e = (v_j, v_i)$ and $e' = (v_{j'}, v_i)$ and assume without loss of generality that v_j is to the left of $v_{j'}$ in the drawing of G_{k-1}. We distinguish three cases based on the available ports of v_j:

C.1: *The north-east port of v_j is unoccupied:* In this case, e emanates from the north-east port of v_j and leads to the south-west port of v_i (recall that all southern ports of singleton v_i are dedicated for incoming red edges; in this case e and e'). If the north-west or the north port of $v_{j'}$ is unoccupied, then e' can be easily drawn bend-less. In the former case, e' emanates from the north-west port of $v_{j'}$ and leads to the south-east port of v_i. In the latter case, e' emanates from the north port of $v_{j'}$ and leads to the south port of v_i. Hence, the aforementioned port assignment fully specifies the position of v_i. It remains to consider the case, where neither the north-west nor the north port of $v_{j'}$ is unoccupied, that is, the north-east port of $v_{j'}$ is unoccupied. By our coloring scheme and IP-2, $v_{j'}$ has already two incoming green edges, say e_g and e'_g, and e' is the last edge to be attached to $v_{j'}$; see Figure 7a. Therefore, there is no other (bend-less) red edge involved. We proceed by shifting $v_{j'}$ up in a way that makes all northern ports of $v_{j'}$ unoccupied; see Figure 7b. Note that we may have to use a second bend on the outgoing blue edge of $v_{j'}$ (in order to maintain the stretchability invariant), but on the other hand we can eliminate one bend from the second green edge e'_g; see Figure 7b. So, the total number of bends remains unchanged. In addition, the endpoints of both e_g and e'_g that are opposite to $v_{j'}$ may have to be moved horizontally to allow e_g and e'_g to be drawn planar, but by the stretchability invariant we are guaranteed that this is always possible. Finally, the stretchability invariant is maintained, since each edge besides the red ones contains a horizontal line-segment.

C.2: *The north-east port of v_j is occupied, while its north port is unoccupied:* In this case, e emanates from the north port of v_j and leads to the south port of v_i (that is, v_i and v_j are vertically aligned). We now claim that the north-west port of $v_{j'}$ is unoccupied. For the sake of contradiction, assume that the claim is not true. By our coloring scheme, the edge attached to the north-west port of $v_{j'}$ is green, which implies that there must exist a path $v_j \rightsquigarrow v_{j'}$ at the outerface face of G_{k-1}

Figure 7: (a) $e' = (v_{j'}, v_i)$ cannot be drawn bend-less. (b) Shifting $v_{j'}$ up resolves the problem.

whose first edge is blue at the north-east port of v_j and its last edge is green at the north-west port of $v_{j'}$. So, path $v_j \rightsquigarrow v_{j'}$ has a vertex which has a neighbor in P_κ for some $\kappa \geq k$. Since v_i is the only such candidate, the contradiction follows from the degree of v_i. Hence, the north-west port of $v_{j'}$ is unoccupied and therefore we can draw e' without bends by using the south-east port of v_i and the north-west port of $v_{j'}$, as desired.

C.3: *Only the north-west port of v_j is unoccupied:* We can reduce this case to case C.1 by applying an operation symmetric to the one of Figure 7a on vertex v_j. This will result in a configuration where all northern ports of v_j (including the north-east) are unoccupied.

Theorem 2 *Let G be a triconnected planar graph of maximum degree 5 with n vertices. A planar octilinear drawing $\Gamma(G)$ of G with at most $2n-2$ bends can be computed in $O(n)$ time.*

Proof: From our extension, it follows that the only edges of $\Gamma(G)$ that have a bend are the blue and the green ones and possibly the third incoming red edge of vertex v_n of the last partition P_m of Π. Now, recall that the blue subgraph is a spanning tree of G, while the green one is a forest on the vertices of G. So, in the worst case the green subgraph is a tree on $n-1$ vertices of G (by construction the green subgraph cannot be incident to the first vertex v_1 of Π). Therefore, at most $2n-2$ edges of $\Gamma(G)$ have a bend. In addition, the running time remains linear since the shifting technique can still be applied (again we assume the real RAM model of computation). This is because once a vertex has been placed its y-coordinate does not change anymore, except for the special case of two red edges (cases C.1 and C.3), which does not influence the overall running time, since it can occur at most once per vertex. □

2.3 Triconnected Planar Graphs of Maximum Degree 6

In this section, we present an algorithm that based on a canonical order Π of a given triconnected planar graph $G = (V, E)$ of maximum degree 6 results in a drawing $\Gamma(G)$ of G, in which each edge has at most two bends. Hence, in total $\Gamma(G)$ has at most $6n-12$ bends. Then, we show how one can appropriately adjust the produced drawing to reduce the total number of bends.

Algorithm 1 describes *rules* R1 - R6 to assign the edges to the ports of the corresponding vertices. It is not difficult to see that all port combinations implied by these

Algorithm 1: `PortAssignment(v)`

input : A vertex v of a triconnected planar graph of maximum degree 6.
output: The port assignment of the edges around v, according to the following rules.

R1: The `incoming blue edges` of v occupy consecutive ports in counterclockwise order around v starting from:

a. the south-east port of v, if $\text{indeg}_\text{b}(v) + \text{outdeg}_\text{r}(v) = 5$; see Figure 8a.
b. the east port of v, if $\text{indeg}_\text{b}(v) + \text{outdeg}_\text{r}(v) = 4$; see Figure 8b.
c. the east port of v, if $\text{outdeg}_\text{g}(v) = 0$ and (a),(b) do not hold; see Figure 8c.
d. the north-east port of v, otherwise; see Figure 8d.

R2: The `outgoing red edge` occupies the counterclockwise next unoccupied port, if v has at least one incoming blue edge. Otherwise, the north-east port of v.

R3: The `incoming green edges` of v occupy consecutive ports in clockwise order around v starting from:

a. the west port of v, if $\text{indeg}_\text{g}(v) + \text{outdeg}_\text{r}(v) + \text{indeg}_\text{b}(v) \geq 4$; see Figure 8e.
b. the north-west port of v, otherwise; see Figure 8f.

R4: The `outgoing blue edge` of v occupies the west port of v, if it is unoccupied; otherwise, the south-west port of v.

R5: The `outgoing green edge` of v occupies the east port of v, if it is unoccupied; otherwise, the south-east port of v.

R6: The `incoming red edges` of v occupy consecutively in counterclockwise direction the south-west, south and south-east ports of v starting from the first available.

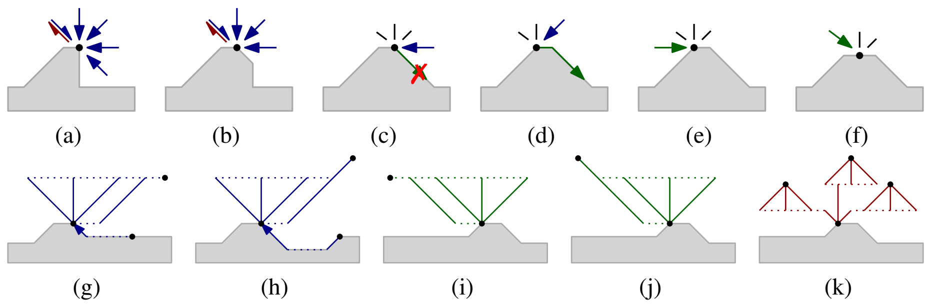

Figure 8: (a)-(f) Illustration of the port assignment computed by Algorithm 1. (g)-(k) Different line-segment combinations with at most two bends (the horizontal ones are drawn dotted)

rules can be realized with at most two bends per edge, so that all edges have a horizontal line-segment, which makes the drawing horizontally stretchable. This also fixes the shape of each edge.

(i) A blue edge emanates from the west or south-west port of a vertex (by rule R4) and leads to one of the south-east, east, north-east, north or north-west ports of its other endvertex (by rule R1); see Figure 8g and 8h,

(ii) A green edge emanates from the east or south-east port of a vertex (by rule R5) and leads to one of the west, north-west, north or north-east ports of its other endvertex (by rule R3); see Figure 8i and 8j,

(iii) A red edge emanates from one of the north-west, north, north-east ports of a vertex (by rule R2) and leads to one of the south-west, south, south-east ports of its other endvertex (by rule R6); see Figure 8k.

To compute the actual drawing $\Gamma(G)$ of G, we follow an incremental approach according to which one partition (that is, a singleton or a chain) of Π is placed at a time, similar to Kant's approach [13] and the corresponding cases of planar graph of maximum degree 4 and 5. Each edge is drawn based on its shape, while the horizontal stretchability ensures that potential crossings can always be eliminated. Note additionally that we adopt the leftist canonical order [1], according to which the leftmost partition is chosen to be placed, when there exist two or more candidates. Since each edge has at most two bends, $\Gamma(G)$ has at most $6n-12$ bends in total.

In the following, we reduce the total number of bends. This is done in two steps. In the first step, we observe that there is no reason to draw the red edges with two bends each; we draw all red edges with at most one bend each. To see this, recall that a red edge emanates from one of the north-west, north, north-east ports of a vertex and leads to one of the south-west, south, south-east ports of its other-endvertex. So, in order to prove that all red edges can be drawn with at most one bend each, we consider in total nine cases, which are illustrated in Figure 9. It is not difficult to see that in each of these cases, the red edge can be drawn with at most one bend. Note that the absence of horizontal line-segments at the red edges does not affect the

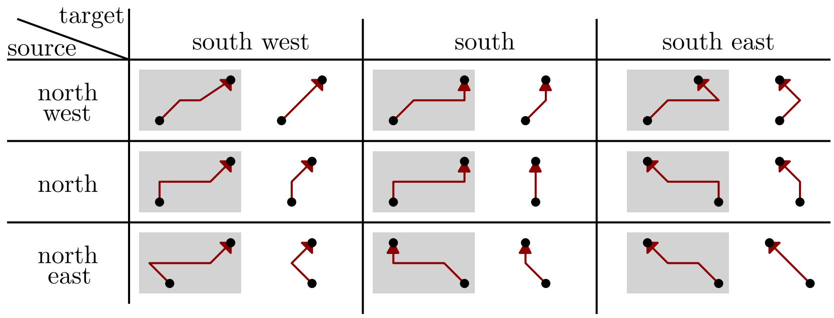

Figure 9: Red edges can be redrawn with one bend (in boxes we show their initial 2-bends shapes)

stretchability of $\Gamma(G)$, since each face of $\Gamma(G)$ has at most two such edges (which both "point upward" at a common vertex). Since a red edge cannot be incident to the outerface of any intermediate drawing constructed during the incremental construction of $\Gamma(G)$, it follows that it is always possible to use only horizontal line-segments (of blue and green edges) to define vertical cuts, thus, avoiding all red edges.

The second step of our bend reduction is more involved. Our claim is that we can "save" two bends per vertex[1], which yields a reduction by roughly $2n$ bends in total. To prove the claim, consider an arbitrary vertex $u \in V$. Our goal is to prove that there always exist two edges incident to u, which can be drawn with only one bend each. By rules R3 and R4, it follows that the west port of vertex u is always occupied, either by an incoming green edge (by rule R3) or by a blue outgoing edge (by rule R4; $u \neq v_1 \in P_0$). Analogously, the east port of vertex u is always occupied, either by a blue incoming edge (by rules R1 and R2) or by an outgoing green edge (by rule R5). Let $(u,v) \in E$ be the edge attached to the west port of u (symmetrically we cope with the edge that is attached to the east port of u). If edge (u,v) is attached to a non-horizontal port at v, then (u,v) is by construction drawn with one bend (regardless of its color; see Figures 8g and 8i) and our claim follows.

It remains to consider the case where edge (u,v) is attached to a horizontal port at v. Assume first that edge (u,v) is blue (we will discuss the case where edge (u,v) is green later). By Algorithm 1, it follows that edge (u,v) is either the first blue incoming edge attached to v (by rules R1b and R1c) or the second one (by rule R1a). We consider each of these cases separately. In rule R1c, observe that edge (u,v) is part of a chain (because $\text{outdeg}_g(u) = 0$). Hence, when placing this chain in the canonical order, we will place u directly to the right of v. This implies that (u,v) will be drawn as a horizontal line-segment (that is, bend-less). Similarly, we cope with rule R1b, when additionally $\text{outdeg}_g(u) = 0$. So, there are still two cases to consider: rule R1a and rule R1b, when additionally $\text{outdeg}_g(u) = 1$; see the left part of Figure 10. In both cases, the current degree of vertex u is 3 and vertex v (and other vertices that are potentially horizontally-aligned with v) must be shifted diagonally up, when u is placed based on the canonical order, such that (u,v) is drawn as a horizontal line-segment (that is, bend-less; see the

[1]Except for vertex v_1 of the first partition P_0 of Π, which has no outgoing blue edge.

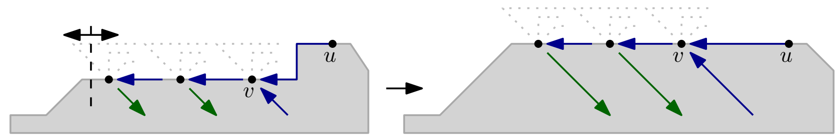

Figure 10: Aligning vertices u and v.

right part of Figure 10). Note that when v is shifted up, vertex v and all vertices that are potentially horizontally-aligned with v are also of degree 3, since otherwise one of these vertices would not have a neighbor in some later partition of Π, which contradicts the definition of Π.

We complete our case analysis with the case where edge (u,v) is green. By rule R3a, it follows that (u,v) is the first green incoming edge attached to u. In addition, when (u,v) is placed based on the canonical order, there is no red outgoing edge attached to u (otherwise u would not be at the outerface of the drawing constructed so far). The leftist canonical order also ensures that there is no blue incoming edge at u drawn before (u,v). Hence, vertex u is of degree 2, when edge (u,v) is placed. Hence, it can be shifted up (potentially with other vertices that are horizontally-aligned with u), such that (u,v) is drawn as a horizontal line-segment (that is, bend-less). We summarize our approach in the following theorem.

Theorem 3 *Let G be a triconnected planar graph of maximum degree 6 with n vertices. A planar octilinear drawing $\Gamma(G)$ of G with at most $3n-8$ bends can be computed in $O(n^2)$ time.*

Proof: Before the two bend-reduction steps, $\Gamma(G)$ contains at most $6n-12$ bends. In the first reduction step, all red edges are drawn with one bend. Hence, $\Gamma(G)$ contains at most $5n-9$ bends. In the second reduction step, we "save" two bends per vertex (except for $v_1 \in P_0$, which has no outgoing blue edge), which yields a reduction by $2n-1$ bends. Therefore, $\Gamma(G)$ contains at most $3n-8$ bends in total. On the negative side, we cannot keep the running time of our algorithm linear. The reason is the second reduction step, which yields changes in the y-coordinates of the vertices. In the worst case, however, quadratic time suffices (under the real RAM model of computation). □

Note that there exist planar graphs of maximum degree 6 that do not admit planar octilinear drawings with at most one bend per edge [2]. Theorem 3 implies that on average one bend per edge suffices.

3 Lower Bounds

In this section, we present lower bounds on the total number of bends for the classes of triconected planar graphs of maximum degree 4, 5 and 6, assuming a fixed outerface in their planar embeddings.

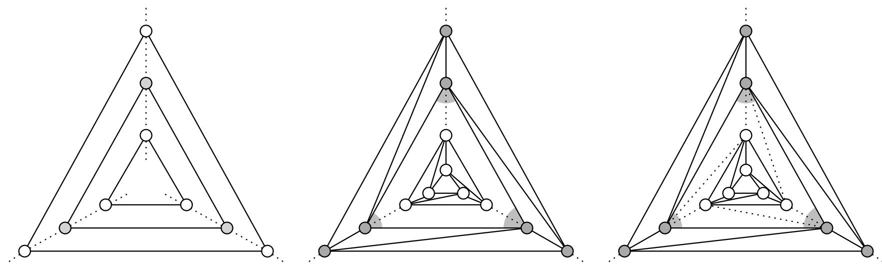

Figure 11: Planar graphs of different degrees that require (a) $n/3-1$, (b) $2n/3-2$ and (c) $4n/3-6$ bends, when drawn in the octilinear model.

3.1 Triconected Planar Graphs of Maximum Degree 4

We start our study with the case of octilinear drawings of planar graphs of maximum degree 4. Our main observation is that if a 3-cycle $\mathscr{C}_3$ of a graph has at least two vertices, each of which has at least one neighbor in the interior of $\mathscr{C}_3$, then at least one edge of $\mathscr{C}_3$ must contain a bend. This is because the sum of the interior angles at the corners of $\mathscr{C}_3$ exceeds $180°$. In fact, elementary geometry implies that a k-cycle, say $\mathscr{C}_k$ with $k \geq 3$, whose vertices have $\sigma \geq 0$ neighbors in the interior of $\mathscr{C}_k$ requires (at least) $\max\{0, \lceil(\sigma - 3k + 8)/3\rceil\}$ bends. Therefore, a bend is necessary. Now, refer to the planar graph of Figure 11a, which clearly has maximum degree 4 as it contains $n/3$ nested triangles, where n is the number of its vertices. It follows that this particular graph requires at least $n/3 - 1$ bends in total.

3.2 Triconected Planar Graphs of Maximum Degree 5 and 6

For planar graphs of maximum degree 5 and 6, our proof becomes more complex and it is actually based on an ILP formulation and a corresponding output obtained by a solver. Our approach is inspired by Tamassia's min-cost flow formulation [20] for computing bend-minimum representations of embedded planar graphs of bounded degree. Since it is rather difficult to implement this algorithm in the case where the underlying drawing model is not the orthogonal model, we developed an ILP instead (refer to Algorithm 2). Recall that a representation describes the "shape" of a drawing without specifying its exact geometry. This is enough to determine a lower bound on the number of bends, even if a bend-optimal octilinear representation may not be realizable by a corresponding (planar) octilinear drawing.

In our formulation, variable $\alpha(u,v) \cdot 45°$ corresponds to the angle formed at vertex u by edge (u,v) and its cyclic predecessor around vertex u. Hence, the following inequalities must hold:

$$1 \leq \alpha(u,v) \leq 8$$

Since the sum of the angles around a vertex is $360°$, it follows that the corresponding sum of the α-variables must be equal to 8, or equivalently:

$$\sum_{v \in N(u)} \alpha(u,v) = 8$$

Given an edge $e = (u,v)$, variables $\ell_{45}(u,v)$, $\ell_{90}(u,v)$ and $\ell_{135}(u,v)$ correspond to the number of left turns at 45°, 90° and 135° when moving along (u,v) from vertex u towards vertex v. Similarly, variables $r_{45}(u,v)$, $r_{90}(u,v)$ and $r_{135}(u,v)$ are defined for right turns. All aforementioned variables are integers lower-bounded by zero. For a face f, we assume that its edges are directed according to the clockwise traversal of f. This implies that each (undirected) edge of the graph appears twice in our formulation. For reasons of symmetry, we require:

$$\begin{aligned} \ell_{45}(u,v) &= \quad r_{45}(v,u) \\ \ell_{90}(u,v) &= \quad r_{90}(v,u) \\ \ell_{135}(u,v) &= \quad r_{135}(v,u) \end{aligned}$$

Since the sum of the angles formed at the vertices and at the bends of a bounded face f equals to $180° \cdot (p(f) - 2)$, where $p(f)$ denotes the total number of such angles, it follows that:

$$\begin{aligned} \textstyle\sum_{(u,v)\in E(f)} \alpha(u,v) + & \quad (\ell_{45}(u,v) + 2\ell_{90}(u,v) + 3\ell_{135}(u,v)) \\ - & \quad (r_{45}(u,v) + 2r_{90}(u,v) + 3r_{135}(u,v)) = 4a(f) - 8, \end{aligned}$$

where $a(f)$ denotes the total number of vertex angles in f, and, $E(f)$ the directed arcs of f in its clockwise traversal. If f is unbounded, the respective sum is increased by 16. Of course, the objective is to minimize the total number of bends over all edges, or:

$$\min \textstyle\sum_{(u,v)\in E} \ell_{45}(u,v) + \ell_{90}(u,v) + \ell_{135}(u,v) + r_{45}(u,v) + r_{90}(u,v) + r_{135}(u,v)$$

Now, consider the planar graph of Figure 11b (which clearly has maximum degree 5) and observe that each "layer" of this graph consist of six vertices that form an octahedron (solid-drawn), while octahedrons of consecutive layers are connected with three edges (dotted-drawn). Using our ILP formulation, we proved that each octahedron subgraph requires at least 4 bends, when drawn in the octilinear model (except for the innermost one for which we could guarantee only two bends). Note that in order to prove so, we had to appropriately lower-bound the gray-colored angles of each octahedron, as we know that there exist edges to be attached in between. This implies that $2n/3 - 2$ bends are required in total to draw the graph of Figure 11b. For the case of planar graphs of maximum degree 6, we applied our ILP approach to a similar graph consisting of nested octahedrons that are connected by six edges each; see Figure 11c. This led to a lower bound of $4n/3 - 6$ bends, as each octahedron except for the innermost one required 8 bends. Summarizing we obtain the following theorem.

Theorem 4 *There exists a class $G_{n,k}$ of triconnected embedded planar graphs of maximum degree k, with $4 \leq k \leq 6$, whose octilinear drawings require at least: (i) $n/3 - 1$ bends, if $k = 4$, (ii) $2n/3 - 2$ bends, if $k = 5$ and (iii) $4n/3 - 6$ bends, if $k = 6$.*

Algorithm 2: ILP to compute bend-optimal octilinear representations.

Input : A planar embedded graph $G = (V,E)$ with maximum degree 8.
Output: A bend-optimal octilinear representation of G.

$$\textbf{min} \sum_{(u,v)\in E} \ell_{45}(u,v) + \ell_{90}(u,v) + \ell_{135}(u,v) + r_{45}(u,v) + r_{90}(u,v) + r_{135}(u,v)$$

s.t:

$$1 \leq a_{(u,v)} \leq 8, \qquad \forall (u,v) \in E$$
$$\sum_{v\in N(u)} a_{(u,v)} = 8, \qquad \forall u \in V$$

$$\ell_{45}(u,v) = r_{45}(v,u), \qquad \forall (u,v) \in E$$
$$\ell_{90}(u,v) = r_{90}(v,u), \qquad \forall (u,v) \in E$$
$$\ell_{135}(u,v) = r_{135}(v,u), \qquad \forall (u,v) \in E$$

$$\sum_{(u,v)\in E(f)} \alpha(u,v) + (\ell_{45}(u,v) + 2\ell_{90}(u,v) + 3\ell_{135}(u,v)) - (r_{45}(u,v) + 2r_{90}(u,v) + 3r_{135}(u,v)) = 4a(f) - 8, \forall f \in F$$

4 Conclusions

In this paper, we studied bounds on the total number of bends of octilinear drawings of triconnected planar graphs. We showed how one can adjust an algorithm of Keszegh et al. [15] to derive an upper bound of $4n - 10$ bends for general planar graphs of maximum degree 8. Then, we adjusted this general bound and previously-known ones for the classes of triconnected planar graphs of maximum degree 4, 5 and 6. For these classes of graphs, we also presented corresponding (but not matching) lower bounds.

We mention two major open problems in this context. The first one is to extend our results to biconnected and simply connected graphs and to further tighten the bounds. Since our drawing algorithms might require super-polynomial area (cf. arguments from [2]), the second problem is to study trade-offs between the total number of bends and the required area.

References

[1] M. Badent, U. Brandes, and S. Cornelsen. More canonical ordering. *Journal of Graph Algorithms and Applications*, 15(1):97–126, 2011. doi:10.7155/jgaa.00219.

[2] M. A. Bekos, M. Gronemann, M. Kaufmann, and R. Krug. Planar octilinear drawings with one bend per edge. *Journal of Graph Algorithms and Applications*, 19(2):657–680, 2015. doi:10.7155/jgaa.00369.

[3] M. A. Bekos, M. Kaufmann, and R. Krug. On the total number of bends for planar octilinear drawings. In E. Kranakis, G. Navarro, and E. Chávez, editors, *LATIN 2016*, volume 9644 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2016. doi:10.1007/978-3-662-49529-2_12.

[4] T. C. Biedl. New lower bounds for orthogonal graph drawings. In F. J. Brandenburg, editor, *Graph Drawing*, volume 1027 of *LNCS*, pages 28–39. Springer, 1996. doi:10.1007/bfb0021788.

[5] T. C. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. *Computational Geometry*, 9(3):159–180, 1998. doi:10.1016/s0925-7721(97)00026-6.

[6] H. L. Bodlaender and G. Tel. A note on rectilinearity and angular resolution. *Journal of Graph Algorithms and Applications*, 8(1):89–94, 2004. doi:10.1142/9789812773289_0006.

[7] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/bf02122694.

[8] E. Di Giacomo, G. Liotta, and F. Montecchiani. The planar slope number of subcubic graphs. In A. Pardo and A. Viola, editors, *LATIN*, volume 8392 of *LNCS*, pages 132–143. Springer, 2014. doi:10.1007/978-3-642-54423-1_12.

[9] S. Felsner. Schnyder woods or how to draw a planar graph? In *Geometric Graphs and Arrangements*, Advanced Lectures in Mathematics, pages 17–42. Vieweg/Teubner Verlag, 2004. doi:10.1007/978-3-322-80303-0_2.

[10] U. Fössmeier, C. Hess, and M. Kaufmann. On improving orthogonal drawings: The 4M-algorithm. In S. Whitesides, editor, *Graph Drawing*, volume 1547 of *LNCS*, pages 125–137. Springer, 1998. doi:10.1007/3-540-37623-2_10.

[11] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001. doi:10.1007/3-540-58950-3_384.

[12] S. Hong, D. Merrick, and H. A. D. do Nascimento. Automatic visualisation of metro maps. *Journal of Visual Languages and Computing*, 17(3):203–224, 2006. doi:10.1016/j.jvlc.2005.09.001.

[13] G. Kant. Drawing planar graphs using the lmc-ordering. In *FOCS*, pages 101–110. IEEE, 1992. doi:10.1109/sfcs.1992.267814.

[14] G. Kant. Hexagonal grid drawings. In E. W. Mayr, editor, *WG*, volume 657 of *LNCS*, pages 263–276. Springer, 1992. doi:10.1007/3-540-56402-0_53.

[15] B. Keszegh, J. Pach, and D. Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. *SIAM Journal of Discrete Mathematics*, 27(2):1171–1183, 2013. doi:10.1137/100815001.

[16] Y. Liu, A. Morgana, and B. Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Mathematics*, 81(1-3):69–91, 1998. doi:10.1016/s0166-218x(97)00076-0.

[17] M. Nöllenburg. Automated drawings of metro maps. Technical Report 2005-25, Fakultät für Informatik, Universität Karlsruhe, 2005.

[18] M. Nöllenburg and A. Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):626–641, 2011. doi:10.1109/tvcg.2010.81.

[19] J. M. Stott, P. Rodgers, J. C. Martinez-Ovando, and S. G. Walker. Automatic metro map layout using multicriteria optimization. *IEEE Transactions on Visualization and Computer Graphics*, 17(1):101–114, 2011. doi:10.1109/tvcg.2010.24.

[20] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal of Computing*, 16(3):421–444, 1987. doi:10.1137/0216030.

[21] R. Tamassia, I. G. Tollis, and J. S. Vitter. Lower bounds for planar orthogonal drawings of graphs. *Information Processing Letters*, 39(1):35–40, 1991. doi:10.1016/0020-0190(91)90059-q.