Algorithmica (2004) 39: 235–253
DOI: 10.1007/s00453-004-1091-4

Algorithmica

Minimising the Number of Bends and Volume in 3-Dimensional Orthogonal Graph Drawings with a Diagonal Vertex Layout[1]

David R. Wood[2]

Abstract. A 3-dimensional orthogonal drawing of a graph with maximum degree at most 6, positions the vertices at grid-points in the 3-dimensional orthogonal grid, and routes edges along grid-lines such that edge routes only intersect at common end-vertices. Minimising the number of bends and the volume of 3-dimensional orthogonal drawings are established criteria for measuring the aesthetic quality of a given drawing. In this paper we present two algorithms for producing 3-dimensional orthogonal graph drawings with the vertices positioned along the main diagonal of a cube, so-called *diagonal* drawings. This vertex-layout strategy was introduced in the 3-BENDS algorithm of Eades et al. [*Discrete Applied Math*. **103**:55–87, 2000]. We show that minimising the number of bends in a diagonal drawing of a given graph is NP-hard. Our first algorithm minimises the total number of bends for a fixed ordering of the vertices along the diagonal in linear time. Using two heuristics for determining this vertex-ordering we obtain upper bounds on the number of bends. Our second algorithm, which is a variation of the above-mentioned 3-BENDS algorithm, produces 3-bend drawings with $n^3 + o(n^3)$ volume, which is the best known upper bound for the volume of 3-dimensional orthogonal graph drawings with at most three bends per edge.

Key Words. Graph drawing, Orthogonal, 3-Dimensional, Diagonal layout, Vertex-ordering, Book embedding.

1. Introduction. The aim of graph drawing is to display a given graph so that the inherent relational information of the graph is clear to the user. There has been substantial research into automatically drawing graphs in two dimensions [11], [17]. Motivated by experimental evidence suggesting that displaying a graph in three dimensions is better than in two [32], [33], [26], there is a growing body of research in 3-dimensional graph drawing. In this paper we are interested in 3-dimensional *orthogonal* graph drawing; here the edges of the graph are drawn as polygonal chains composed of axis-parallel segments. This style of drawing has applications in VLSI circuit design; see for example [1] and [20].

Throughout this paper we consider n-vertex m-edge undirected graphs G, possibly with parallel edges but no loops, with vertex set $V(G)$ and edge set $E(G)$. The 3-

[1] This research was supported by NSERC. It was partially completed at Monash University (Melbourne, Australia) and at The University of Sydney (Sydney, Australia), where it was supported by the ARC. Some of these results were presented at the 8th Australasian Workshop on Combinatorial Algorithms (AWOCA '97), Noosa, Queensland, Australia, July 14–18, 1997.
[2] School of Computer Science, Carleton University, Ottawa, Ontario, Canada K1S 5B6. davidw@scs.carleton.ca.

Received December 15, 2000; revised July 19, 2002, and December 3, 2003. Communicated by G. N. Fredrickson. Online publication March 19, 2004.

dimensional orthogonal grid consists of *grid-points* in 3-dimensional space with integer coordinates, together with the axis-parallel *grid-lines* determined by these points. The $I = k$ plane, for some $I \in \{X, Y, Z\}$ and integer k, is called a *grid-plane*. A 3-*dimensional orthogonal drawing* of a graph positions each vertex at a unique grid-point, and routes each edge as a polygonal chain composed of contiguous sequences of axis-parallel segments contained in grid-lines, such that (a) the end-points of an edge route are the grid-points representing the end-vertices of the edge, and (b) no two edge routes *cross* (that is, distinct edge routes only intersect at a common end-vertex). For brevity we refer to a 3-dimensional orthogonal drawing as a *drawing*. At a vertex v the six directions, or *ports*, the edges incident with v can use are denoted by X_v^+, X_v^-, Y_v^+, Y_v^-, Z_v^+ and Z_v^-. Clearly, drawings can only exist for graphs with maximum degree at most 6, so-called 6-*graphs*. To construct orthogonal drawings of graphs with degree greater than 6, vertices can be represented by grid-boxes [5], [7], [16], [28] [34], [36] or by points in a multidimensional grid [35].

Every 6-graph has an infinite number of drawings. Various criteria have been proposed in the literature to evaluate the aesthetic quality of a particular drawing. Firstly, the volume of the drawing should be small. The *volume* of a drawing is the volume of the smallest axis-aligned box, called the *bounding box*, that encloses the drawing. For convenience, we consider the dimensions of the bounding box to be the number of grid-points along each side (which is one more than the actual side length). This enables a 2-dimensional drawing to have positive volume.

Drawings with many bends in the edges appear cluttered and are difficult to visualise, and in VLSI circuits, many bends increase the cost of production and the chance of circuit failure. Therefore minimising the number of bends is an important aesthetic criterion for orthogonal drawings. In this paper we consider bounds on both the maximum number of bends per edge and the total number of bends in 3-dimensional orthogonal graph drawings. A drawing with no more than b bends per edge is called a *b-bend drawing*.

Using straightforward extensions of the corresponding 2-dimensional NP-hardness results, minimising either the volume or the total number of bends in a drawing of a given 6-graph is NP-hard [13]. Other proposed aesthetic criteria include the length of edges and whether the drawing is "truly 3-dimensional". A number of tradeoffs between aesthetic criteria, most notably between the maximum number of bends per edge route and the bounding box volume [14], have been observed in existing algorithms.

A lower bound of $\Omega(n^{3/2})$ for the volume of a drawing was established by Kolmogorov and Barzdin [19] (also see [6] and [30]). This lower bound is asymptotically matched by algorithms of Biedl [6] and Eades and co-workers [13], [14], which all produce drawings with $\mathcal{O}(n^{3/2})$ volume. The COMPACT algorithm of Eades et al. [14], which routes each edge with at most seven bends, uses the least number of bends out of these algorithms. Other algorithms for 3-dimensional orthogonal graph drawing have been proposed by Biedl et al. [4], Closson et al. [10], Eades et al. [14], Di Battista et al. [12], Papakostas and Tollis [28] and Wood [39]. Wood [38] establishes lower bounds for the number of bends, and Lynn et al. [23] introduce a number of postprocessing techniques for the refinement of drawings.

That every 6-graph has a 3-bend drawing was established by the 3-BENDS algorithm of Eades et al. [14] and the INCREMENTAL algorithm of Papakostast and Tollis [28]. The

INCREMENTAL algorithm,[3] which supports the on-line insertion of vertices in constant time, produces drawings with $4.63n^3$ volume. The 3-BENDS algorithm produces drawings with $27n^3$ volume[4] by positioning each vertex v_i at $(3i, 3i, 3i)$ for some arbitrary vertex-ordering $(v_1, v_2, \dots, v_n)$. A similar vertex layout strategy is employed in this paper.

DEFINITION 1. For a graph G, a total order $<$ on $V(G)$ induces a numbering $(v_1, v_2, \dots, v_n)$ of $V(G)$ and vice versa. We refer to both $<$ and $(v_1, v_2, \dots, v_n)$ as a *vertex-ordering* of G. A 3-dimensional orthogonal drawing of a graph G is *diagonal* if there exists a vertex-ordering of G such that for all vertices $v, w \in V(G)$ with $v < w$, and for each dimension $I \in \{X, Y, Z\}$, the I-coordinate of v is less than the I-coordinate of w. The vertex-ordering associated with a diagonal drawing is called the *diagonal ordering*.

Diagonal drawings are examples of the wider class of *general position* 3-dimensional orthogonal drawings, in which no two vertices are in a common grid-plane.[5] This model, introduced by Papakostas and Tollis [28] and Biedl [7] in the context of 3-dimensional orthogonal box-drawings, has also been used in the quadratic-time algorithm of [39], which produces general position 4-bend drawings of simple 6-graphs with at most $\frac{16}{7}m$ bends and at most $2.13n^3$ volume. This algorithm moves the vertices from an initial diagonal layout with the aim of reducing bends.

Our aim in this paper is to study the minimisation of bends and volume in diagonal drawings. We make the following contributions. First, we prove that it is NP-hard to minimise the number of bends in a diagonal drawing of a given 6-graph, and remains NP-hard for bipartite 6-graphs. The main result of this paper is an algorithm which, given a fixed diagonal ordering of a 6-graph, determines a diagonal drawing with the minimum number of bends in $\mathcal{O}(n)$ time. This algorithm is described in Section 3. In Section 4 we analyse two heuristics for determining an appropriate vertex-ordering. Using st-orderings our algorithm produces drawings of 6-regular graphs with $\frac{8}{3}m$ bends. The second heuristic produces drawings of simple 6-regular graphs with at most $\frac{31}{12}m$ bends. The final contribution of this paper, described in Section 5, is a variant of the 3-BENDS algorithm of Eades et al. [14], which, using a particular diagonal ordering and a modified edge routing strategy, produces diagonal drawings with $n^3 + o(n^3)$ volume, which is the best known upper bound for the volume of 3-bend drawings.

2. Notation.

Let $(v_1, v_2, \dots, v_n)$ be a vertex-ordering of a graph G with corresponding total order $<$. For each edge $e = vw \in E(G)$ with $v < w$ we define $R(e) = w$ and $L(e) = v$; e is called a *successor edge* of v and a *predecessor edge* of w. The number of successor and predecessor edges of a vertex v are denoted by $\text{succ}(v)$

[3] The INCREMENTAL algorithm, as stated in [28], only works for simple graphs, however, with a suitable modification it also works for multigraphs [A. Papakostas, private communication, 1998].

[4] By deleting grid-planes not containing a vertex or a bend the volume is reduced to $8n^3$.

[5] In the computational geometry literature, a set of points in 3-space are in *general position* if no three are collinear and no four are coplanar. Therefore *general grid position* may be a better term to describe a set of grid-points with no two in a common grid-plane. However, the former term has been adopted as standard in the orthogonal graph drawing literature [7], [5], [34], [39].

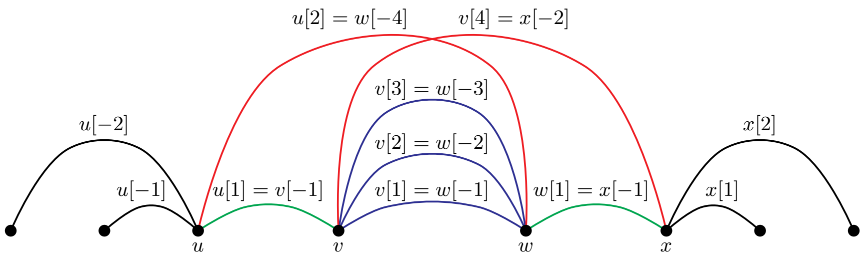

Fig. 1. Numbering edges in a vertex-ordering.

and pred(v), respectively. That is, pred$(v) = |\{vw \in E(G) : w < v\}|$ and succ$(v) = |\{vw \in E(G) : v < w\}|$. The successor edges of v are denoted by $v[1], v[2], \ldots, v[\text{succ}](v)$, where

$$R(v[1]) \leq R(v[2]) \leq \ldots \leq R(v[\text{succ}(v)]).$$

The predecessor edges of v are denoted by $v[-1], v[-2], \ldots, v[-\text{pred}(v)]$, where

$$L(v[-\text{pred}(v)]) \leq L(v[-\text{pred}(v)+1]) \leq \ldots \leq L(v[-1]).$$

Furthermore, we require the following consistent numbering of parallel edges. If $e_1 = \{v, w\}$ and $e_2 = \{v, w\}$ are parallel edges with $e_1 = v[i] = w[j]$, then we require that if $e_2 = v[i+1]$, then $e_2 = w[j-1]$, as illustrated in the example of Figure 1.

Associated with a graph G is the *arc set* $A(G) = \{(v, w), (w, v) : vw \in E(G)\}$ consisting of two anti-parallel directed arcs for each edge of G. An arc $(v, w) \in A(G)$ is denoted by $\overrightarrow{vw}$, and is called the *reversal* of $\overrightarrow{wv}$. Given a vertex-ordering of G, for each edge $e \in E(G)$, if $e = v[i] = w[j]$, then the arcs $\overrightarrow{vw}$ and $\overrightarrow{wv}$ associated with e are denoted by $\overrightarrow{v[i]}$ and $\overrightarrow{w[j]}$, respectively.

3. Bend-Minimum Algorithm. We now describe an algorithm which, given a fixed diagonal ordering of a 6-graph G, determines a bend-minimum drawing of G. First, we establish an elementary lower bound for the number of bends in a diagonal drawing. Consider a diagonal drawing of a 6-graph G with corresponding vertex-ordering $(v_1, v_2, \ldots, v_n)$. Since any two vertices differ in all three coordinates, every edge has at least two bends. Furthermore, if an edge $v_i v_j$ with $i < j$ uses a negative port at v_i or a positive port at v_j, then the edge needs at least three bends. If $v_i v_j$ uses a negative port at v_i and a positive port at v_j, then the edge needs at least four bends. Each vertex $v \in V(G)$ has at least $\max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\}$ incident edges that must use a port at v which points away from the destination of the edge. Therefore the number of bends in a diagonal drawing of an m-edge graph G is at least

$$2m + \sum_{v \in V(G)} \max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\}. \tag{1}$$

Algorithm Bend-Minimum Diagonal Drawing below, given a fixed diagonal ordering $(v_1, v_2, \ldots, v_n)$, determines a diagonal drawing with precisely this number of

Table 1. Definition of $v[A], v[B], \ldots, v[F]$ for each type of vertex v.

	v[A]	v[B]	v[C]	v[D]	v[E]	v[F]
$\text{succ}(v) = 6$	v[4]	v[5]	v[6]	v[1]	v[2]	v[3]
$\text{succ}(v) = 5$	v[-1]	v[4]	v[5]	v[1]	v[2]	v[3]
$\text{succ}(v) = 4$	v[-2]	v[-1]	v[4]	v[1]	v[2]	v[3]
$\text{succ}(v), \text{pred}(v) \leq 3$	v[-3]	v[-2]	v[-1]	v[1]	v[2]	v[3]
$\text{pred}(v) = 4$	v[-3]	v[-2]	v[-1]	v[-4]	v[1]	v[2]
$\text{pred}(v) = 5$	v[-3]	v[-2]	v[-1]	v[-5]	v[-4]	v[1]
$\text{pred}(v) = 6$	v[-3]	v[-2]	v[-1]	v[-6]	v[-5]	v[-4]

bends. The algorithm employs the following notation defined in Table 1. For each vertex $v \in V(G)$, label the edges incident to v by $v[A], v[B], \ldots, v[F]$, depending on pred(v) and succ(v), where $\{A, B, \ldots, F\} \subseteq \{-\text{pred}(v), \ldots, \text{succ}(v)\}$. Note that if $\deg(v) < 6$, then some of $v[A], v[B], \ldots, v[F]$ will not be defined.

The algorithm will assign negative ports at v to the edges $\{v[A], v[B], v[C]\}$, and positive ports at v to $\{v[D], v[E], v[F]\}$. The assignment of edges to ports is modelled by a 3-colouring of $A(G)$ with colours $\{X, Y, Z\}$. If the arc $\overrightarrow{vw}$ is coloured $I \in \{X, Y, Z\}$, then the edge vw will use an I-port at v.

LEMMA 1. *Given a vertex-ordering $(v_1, v_2, \ldots, v_n)$ of a 6-graph G there is a 3-colouring of $A(G)$ that can be determined in $\mathcal{O}(n)$ time, such that*:

(a) *reversal arcs are coloured differently*; *that is, for every edge $vw \in E(G)$, $\text{col}(\overrightarrow{vw}) \neq \text{col}(\overrightarrow{wv})$, and*
(b) *for every vertex $v \in V(G)$, $\text{col}(v[A]), \text{col}(v[B])$ and $\text{col}(v[C])$ are pairwise distinct, and $\text{col}(v[D]), \text{col}(v[E])$ and $\text{col}(v[F])$ are pairwise distinct.*

PROOF. Construct a graph H with $V(H) = A(G)$ as follows. We refer to a vertex of H by the corresponding arc in $A(G)$. For each vertex $v \in V(G)$, add cliques $\{\overrightarrow{v[A]}, \overrightarrow{v[B]}, \overrightarrow{v[C]}\}$ and $\{\overrightarrow{v[D]}, \overrightarrow{v[E]}, \overrightarrow{v[F]}\}$ to $E(H)$. These edges are called "c" edges. Note that the above-mentioned cliques may be empty or consist of a single edge. For each edge $vw \in E(G)$, add the edge $\{\overrightarrow{vw}, \overrightarrow{wv}\}$, called an "r"-edge, to $E(H)$, as illustrated in the example of Figure 2.

A vertex of H is incident with one "r"-edge and at most two "c"-edges. Hence H has maximum degree at most 3, and is not K_4; thus by Brooks's theorem [9], H is vertex 3-colourable. The proof of Brooks's theorem due to Lovász [22] describes an algorithm for vertex 3-colouring H in $\mathcal{O}(|E(H)|) \subseteq \mathcal{O}(n)$ time [2]. A vertex 3-colouring of H defines a 3-colouring of $A(G)$ with the claimed properties. $\square$

Each edge constructed by our algorithm consists of three consecutive perpendicular segments possibly with unit length segments attached at either end. If an edge vw has such a unit length segment attached at v, then we say the edge vw is *anchored* at v, and

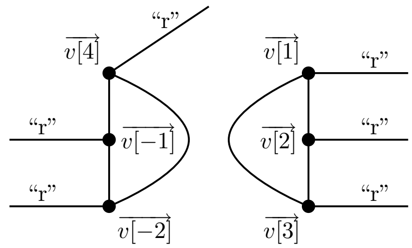

Fig. 2. Subgraph of H corresponding to a vertex v with $\mathrm{pred}(v) = 2$ and $\mathrm{succ}(v) = 4$.

the arc $\overrightarrow{vw}$ is *anchored*. (See [39] for analogous definitions in the context of arbitrary general position drawings.)

Algorithm 1. BEND-MINIMUM DIAGONAL DRAWING

Input: vertex-ordering $(v_1, v_2, \ldots, v_n)$ of a 6-graph G
Output: diagonal drawing of G

1. Initially position each vertex v_i, $1 \le i \le n$, at $(3i, 3i, 3i)$.
2. Determine a 3-colouring of $A(G)$ with colours $\{X, Y, Z\}$, as described in Lemma 1.
3. Construct an edge route for each edge $e = v_i v_j \in E(G)$ as follows. Suppose $i < j$, $e = v_i[\alpha] = v_j[\beta]$, $\mathrm{col}(\overrightarrow{v_i v_j}) = I$ and $\mathrm{col}(\overrightarrow{v_j v_i}) = J$, for some $\alpha, \beta \in \{A, B, \ldots, F\}$ and $I, J \in \{X, Y, Z\}$. By Lemma 1(a), $I \neq J$. In what follows we use (I, J, K)-coordinates, where $K \in \{X, Y, Z\} \backslash \{I, J\}$.
   (a) If $\alpha \in \{D, E, F\}$ and $\beta \in \{A, B, C\}$, then, as illustrated in Figure 3(a), route $v_i v_j$ with the 2-bend edge route $(3i, 3i, 3i) \to (3j, 3i, 3i) \to (3j, 3i, 3j) \to (3j, 3j, 3j)$ using the $I^+_{v_i}$ and $J^-_{v_j}$ ports.
   (b) If $\alpha \in \{A, B, C\}$ and $\beta \in \{A, B, C\}$, then, as illustrated in Figure 3(b), route $v_i v_j$ with the 3-bend edge route $(3i, 3i, 3i) \to (3i - 1, 3i, 3i) \to (3i - 1, 3i, 3j) \to (3j, 3i, 3j) \to (3j, 3j, 3j)$ using the $I^-_{v_i}$ and $J^-_{v_j}$ ports.
   (c) If $\alpha \in \{D, E, F\}$ and $\beta \in \{D, E, F\}$, then, as illustrated in Figure 3(c), route $v_i v_j$ with the 3-bend edge route $(3i, 3i, 3i) \to (3j, 3i, 3i) \to (3j, 3j + 1, 3i) \to (3j, 3j + 1, 3j) \to (3j, 3j, 3j)$ using the $I^+_{v_i}$ and $J^+_{v_j}$ ports.

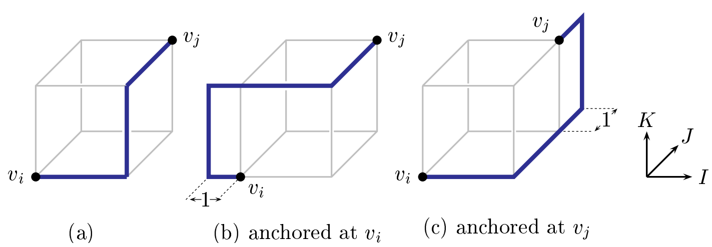

Fig. 3. 2-Bend and 3-bend edge routes.

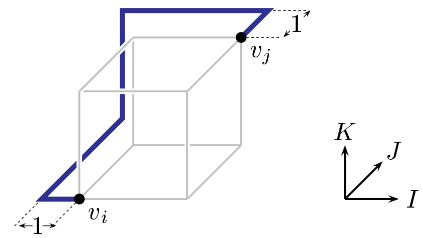

Fig. 4. 4-Bend edge route anchored at v_i and at v_j.

(d) If $\alpha \in \{A, B, C\}$ and $\beta \in \{D, E, F\}$, then, as illustrated in Figure 4, route $v_i v_j$ with the 4-bend edge route $(3i, 3i, 3i) \to (3i-1, 3i, 3i) \to (3i-1, 3j+1, 3i) \to (3i-1, 3j+1, 3j) \to (3j, 3j+1, 3j) \to (3j, 3j, 3j)$ using the $I^-_{v_i}$ and $J^+_{v_j}$ ports.

4. For each pair of vertices v, w with more than one 2- or 3-bend edge between v and w, reroute such edges as illustrated in Figure 5 depending on the set of ports at v and w assigned to these edges. Existing 4-bend edge routes vw are left as is. It is easily checked that the cases shown in Figure 5 suffice (due symmetry between v and w, the numbering of edges in Section 2 and Table 1, and the fact that reversal arcs are coloured differently).
5. Delete grid-planes not containing a vertex or a bend.

THEOREM 1. *The* BEND-MINIMUM DIAGONAL DRAWING *algorithm determines, in* $\mathcal{O}(n)$ *time, a diagonal drawing of* G *with* $2m + k$ *bends and* $(n + k/3)^3$ *volume, where*

$$k = \sum_{v \in V(G)} \max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\}$$

is the number of anchored arcs.

PROOF. We first show that each edge is assigned a unique port at its end-vertices. Before parallel edges are rerouted in Step 4, the edges $v[A]$, $v[B]$ and $v[C]$ use negative ports at a vertex v. In particular, $v[A]$ (respectively, $v[B]$ and $v[C]$) is assigned the I^-_v port where the arc $\overrightarrow{v[A]}$ (respectively, $\overrightarrow{v[B]}$ and $\overrightarrow{v[C]}$) is coloured $I \in \{X, Y, Z\}$. By Lemma 1(b), $\overrightarrow{v[A]}$, $\overrightarrow{v[B]}$ and $\overrightarrow{v[C]}$ receive distinct colours. Hence $v[A]$, $v[B]$ and $v[C]$ use distinct negative ports at v. Similarly, the edges $v[D]$, $v[E]$ and $v[F]$ use distinct positive ports at v. Therefore all edges incident to v are assigned distinct ports at v, which clearly also holds after Step 4.

For each vertex $v \in V(G)$, if $\mathrm{succ}(v) > \mathrm{pred}(v)$, then an edge $v[\alpha]$ is anchored at v if and only if $\alpha \in \{A, B, C\}$; hence the arcs $\overrightarrow{v[\alpha]}$, $4 \le \alpha \le \mathrm{succ}(v)$, are anchored (refer to Table 1). If $\mathrm{pred}(v) > \mathrm{succ}(v)$, then an edge $v[\alpha]$ is anchored at v if and only if $\alpha \in \{D, E, F\}$; hence the arcs $\overrightarrow{v[-\alpha]}$, $4 \le \alpha \le \mathrm{pred}(v)$, are anchored (refer to Table 1). Therefore the number of anchored arcs

$$k = \sum_{v \in V(G)} \max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\}.$$

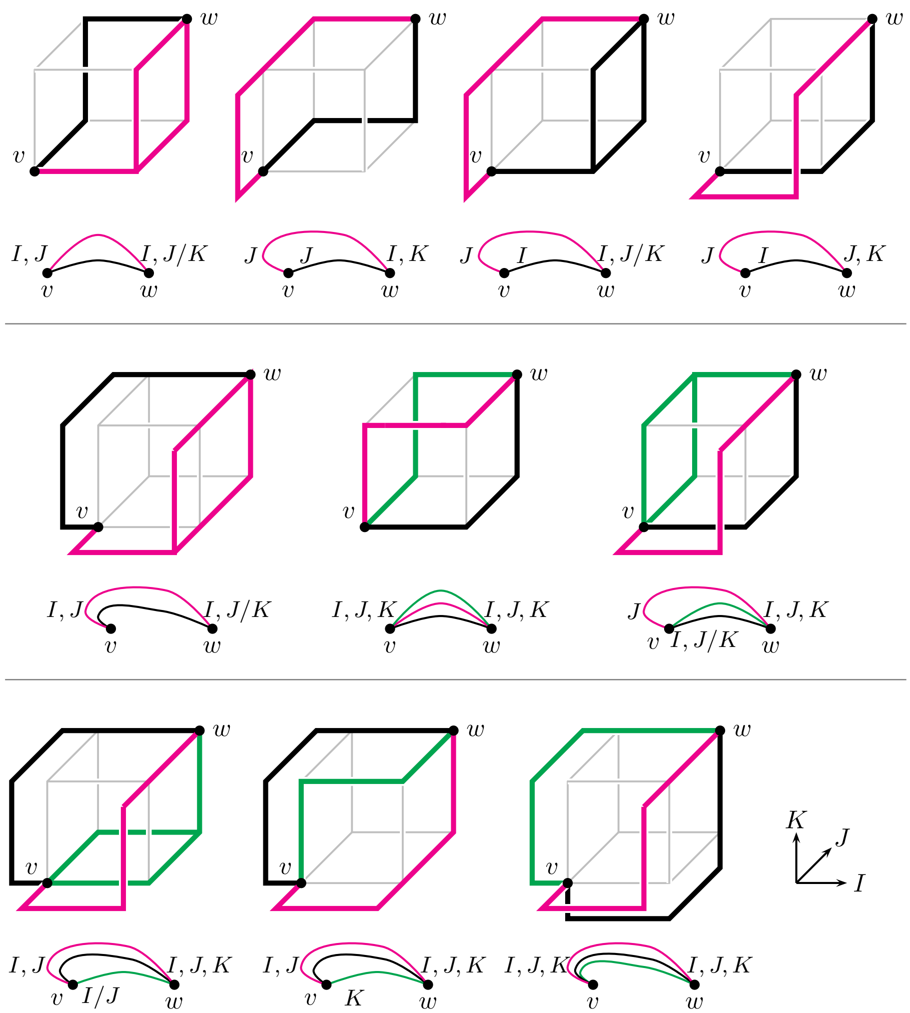

Fig. 5. Rerouting parallel 2-bend and 3-bend edge routes.

Each edge vw has two bends if neither of $\overrightarrow{vw}$ and $\overrightarrow{wv}$ is anchored, three bends if exactly one of $\overrightarrow{vw}$ and $\overrightarrow{wv}$ is anchored, and four bends if both of $\overrightarrow{vw}$ and $\overrightarrow{wv}$ are anchored. Thus the total number of bends before Step 4 of the algorithm is

$$2m + \sum_{v \in V(G)} \max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\}.$$

Step 4 preserves the total number of bends. Thus the final drawing has the claimed number of bends.

We now prove that there are no edge crossings. For each vertex v_i and dimension $I \in \{X, Y, Z\}$, we say that the $(I = 3i - 1)$-plane, the $(I = 3i)$-plane and the $(I = 3i + 1)$-plane (and the grid-points contained in these planes) *belong to* v_i. Clearly, all

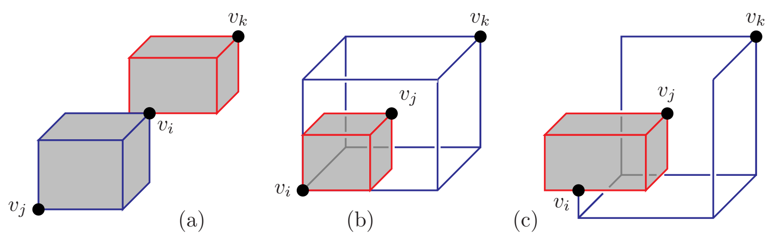

Fig. 6. Non-crossing edge routes.

grid-points contained in an edge route vw belong to both v and w. Suppose the edges vw and xy cross. Then the grid-point of intersection must belong to each of v, w, x and y. Since each grid-plane belongs to a single vertex, each grid-point belongs to at most three vertices. Therefore two of $\{v, w, x, y\}$ are equal; that is, intersecting edges must be incident to a common vertex. Suppose for the sake of contradiction that before Step 5 of the algorithm, the edges $e_1 = v_i v_j$ and $e_2 = v_i v_k$ cross.

In all edge routes, before Step 5 there are no consecutive unit length segments, and an edge crossing involving a unit-length segment must also involve the adjacent non-unit-length segment. Thus we need only consider crossings between the non-unit-length segments of e_1 and e_2. Consider the following cases which depend on the relative values of i, j and k, and whether $\overrightarrow{v_i v_j}$ and $\overrightarrow{v_i v_k}$ are anchored. As discussed above, if $i < j < k$ and $\overrightarrow{v_i v_j}$ is anchored, then $\overrightarrow{v_i v_k}$ will also be anchored. We assume without loss of generality that $\mathrm{succ}(v_i) \geq \mathrm{pred}(v_i)$.

Case 1: $j < i < k$. By construction, $\overrightarrow{v_i v_j}$ is not anchored. Thus, the edge route $v_i v_j$ is contained within the box with corners at $(3j-1, 3j-1, 3j-1)$ and $(3i, 3i, 3i)$. We can suppose without loss of generality that $\mathrm{col}(\overrightarrow{v_i v_k}) = X$, the edge route $v_i v_k$ is contained within the box with corners at $(3i-1, 3i, 3i)$ and $(3k+1, 3k+1, 3k+1)$, as illustrated in Figure 6(a). Since $v_i v_j$ and $v_i v_k$ use different ports at v_i, both $v_i v_j$ and $v_i v_k$ cannot pass through $(3i-1, 3i, 3i)$ which is the only grid-point besides v_i in both boxes. Thus e_1 and e_2 do not cross.

Case 2: *$i < j < k$ and both $\overrightarrow{v_i v_j}$ and $\overrightarrow{v_i v_k}$ are not anchored.* As illustrated in Figure 6(b), since $v_i v_j$ and $v_i v_k$ use different ports at v_i the edge routes do not cross.

Case 3: *$i < j < k$ and $\overrightarrow{v_i v_k}$ is anchored.* Suppose without loss of generality that $\mathrm{col}(\overrightarrow{v_i v_j}) = X$. As illustrated in Figure 6(c), the edge $v_i v_j$ is contained in the box with corners at $(3i-1, 3i, 3i)$ and $(3j+1, 3j+1, 3j+1)$. Since $v_i v_j$ and $v_i v_k$ use different ports at v_i, $v_i v_k$ does not enter this box. Hence e_1 and e_2 do not cross.

Case 4: $i < j = k$. Thus e_1 and e_2 are parallel edges. Two parallel 4-bend edges do not cross as illustrated in Figure 7(a). Parallel edges with at most three bends are rerouted in Step 4 of the algorithm with no crossings. As illustrated in Figure 7(b), a 4-bend edge and a parallel edge with at most three bends (which is contained in the shaded region) do not cross.

Therefore the algorithm produces a drawing with no edge crossings. It should be noted that the choice of edge numbering in Section 2 is crucial to eliminate problematic

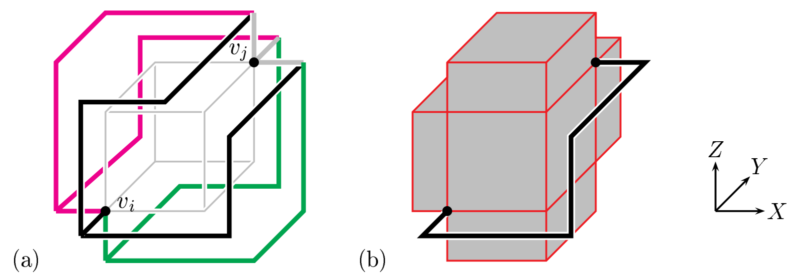

Fig. 7. Parallel 4-bend edges do not cross.

cases in Step 4 of the algorithm. For example, given a port assignment for the two parallel edges illustrated in Figure 8, it is not possible to route the edges using the given ports without possibly creating a crossing. This case will not arise since this would imply an inconsistent numbering of the edges, as illustrated in the example of Figure 8.

Finally we measure the volume of the bounding box. A grid-plane containing an edge segment also contains an end-vertex of the edge or a bend in the edge route. Thus a grid-plane not containing a vertex or a bend can be removed without affecting the drawing. Before removing such "empty" grid-planes, for each dimension $I \in \{X, Y, Z\}$, the $(I = 3i - 1)$-plane (respectively, $(I = 3i + 1)$-plane) belonging to the vertex v_i contains a bend if and only if there is an anchored edge route using the I_v^- (I_v^+) port. After removing the empty grid-planes in Step 5, the bounding box is thus $(n + k_X) \times (n + k_Y) \times (n + k_Z)$, where k_I is the number of anchored arcs coloured $I \in \{X, Y, Z\}$. It is well known that of the boxes with fixed sum of side length the cube has maximum volume (see for example [18]). Hence if k is the total number of anchored arcs, then the bounding box volume is maximised when $k_X = k_Y = k_Z = k/3$, thus the bounding box volume is at most $(n + k/3)^3$. Each step of the BEND-MINIMUM DIAGONAL DRAWING algorithm can easily be implemented in $\mathcal{O}(n)$ time. □

By (1) the number of bends in the drawings produced by the BEND-MINIMUM DIAGONAL DRAWING algorithm is optimal for a fixed diagonal ordering. For 6-graphs with minimum degree 5, $\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} \geq 3$, and hence by Theorem 1, the number of bends in a drawing produced by the BEND-MINIMUM DIAGONAL DRAWING algorithm is $2m - 3n + \sum_v \max\{\mathrm{succ}(v), \mathrm{pred}(v)\}$. Therefore minimising the number of bends

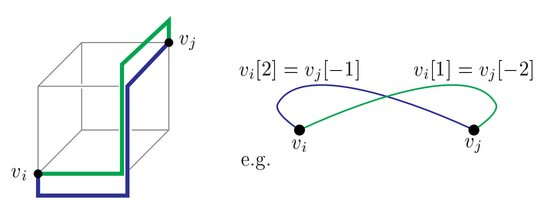

Fig. 8. A crossing between parallel edges.

in a diagonal drawing of a 6-graph with minimum degree 5 is equivalent to finding a vertex-ordering that minimises

$$\sum_{v \in V(G)} \max\{\mathrm{succ}(v), \mathrm{pred}(v)\}. \tag{2}$$

Biedl et al. [3] show that minimising (2) is NP-hard, and remains NP-hard for bipartite graphs with minimum degree 5 and maximum degree 6. The next result follows.

THEOREM 2. *Minimising the number of bends in a diagonal drawing of a given* 6*-graph is NP-hard, and remains NP-hard for bipartite* 6*-graphs.*

Since $\max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\} = 0$ if and only if $\mathrm{succ}(v) \leq 3$ and $\mathrm{pred}(v) \leq 3$, we have the following characterisation of 2-bend diagonal drawings.

COROLLARY 1. *A diagonal ordering of a* 6*-graph G admits a* 2*-bend* 3*-dimensional orthogonal drawing if and only if every vertex v has* $\mathrm{succ}(v) \leq 3$ *and* $\mathrm{pred}(v) \leq 3$.

4. Analysis of Heuristics. We now describe and analyse two methods for determining an appropriate diagonal ordering; that is, a vertex-ordering with bounds on $\sum_v \max\{\mathrm{succ}(v), \mathrm{pred}(v)\}$.

A vertex-ordering $(v_1, v_2, \ldots, v_n)$ of a graph G is an st-ordering if $v_1 = s$, $v_n = t$, and for every other vertex v_i, $1 < i < n$, $\mathrm{pred}(v_i) \geq 1$ and $\mathrm{succ}(v_i) \geq 1$. Lempel et al. [21] show that for a biconnected graph G and for all $s, t \in V(G)$, there exists an st-ordering of G. If we apply the BEND-MINIMUM DIAGONAL DRAWING algorithm with the diagonal ordering an st-ordering we obtain the following result.

THEOREM 3. *An n-vertex m-edge biconnected* 6*-graph G has a diagonal drawing, that can be determined in $\mathcal{O}(n)$ time, with at most $3m - n + \mathcal{O}(1)$ bends and at most $(\frac{1}{3}(2n + m) + \mathcal{O}(1))^3$ volume. If G is* 6*-regular, then the number of bends is at most $\frac{8}{3}m + \mathcal{O}(1)$ and the volume is at most $4.63n^3 + o(n^3)$.*

PROOF. Let n_i be the number of vertices $v \in V(G)$ with $\deg(v) = i$. Since G is biconnected, $n_0 = 0$ and $n_1 = 0$. Consider an st-ordering of G. For each $v \in \{s, t\}$, $\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} = \deg(v)$; thus

$$\max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\} = \begin{cases} 0, & \text{if } \deg(v) \leq 3; \\ 1, & \text{if } \deg(v) = 4; \\ 2, & \text{if } \deg(v) = 5; \\ 3, & \text{if } \deg(v) = 6. \end{cases}$$

For all other vertices v we have

$$\max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\} \leq \begin{cases} 0, & \text{if } \deg(v) \leq 4; \\ 1, & \text{if } \deg(v) = 5; \\ 2, & \text{if } \deg(v) = 6. \end{cases}$$

Hence, when applying the BEND-MINIMUM DIAGONAL DRAWING algorithm, by Theorem 1, the number of bends is

$$2m \;+\; \sum_{v \in V(G)} \max\{\max\{\mathrm{succ}(v), \mathrm{pred}(v)\} - 3, 0\} \;\leq\; 2m + n_5 + 2n_6 + 2.$$

Now,

$$\begin{aligned} 0 &\leq n_3 + 2n_4 + n_5, \\ 2n_5 + 4n_6 &\leq n_3 + 2n_4 + 3n_5 + 4n_6, \\ 2n_5 + 4n_6 &\leq (2n_2 + 3n_3 + 4n_4 + 5n_5 + 6n_6) - 2n, \\ 2n_5 + 4n_6 &\leq 2(m - n), \\ n_5 + 2n_6 &\leq m - n, \\ 2m + n_5 + 2n_6 &\leq 3m - n. \end{aligned}$$

Thus the number of bends is at most $3m - n + 2$, and by Theorem 1 the volume is at most $(n + \frac{1}{3}(m - n + 2))^3 = (\frac{1}{3}(2n + m + 2))^3$. If G is 6-regular, then $m = 3n$ and the claimed bounds immediately follow. Using the algorithm of [15] and by Theorem 1, the st-ordering and the drawing itself can be determined in $\mathcal{O}(n)$ time. □

Note that for a non-biconnected graph G with a constant number of end-blocks, a similar method to the above (see [3]) establishes the same upper bounds on the number of bends and the volume of a drawing of G as in Theorem 3.

Our second heuristic for determining a balanced vertex-ordering of a simple graph inserts each vertex, in turn, midway between its already inserted neighbours, and in the case of an odd number of already inserted neighbours, minimises the imbalance of the median neighbour. We call this the MEDIAN PLACEMENT algorithm; see [3] for details. Similar methods were used for graph drawing in [8] for example. Employing the MEDIAN PLACEMENT algorithm to determine the diagonal ordering, we obtain the following result.

THEOREM 4. *A simple n-vertex m-edge 6-graph G has a diagonal drawing, that can be determined in $\mathcal{O}(n)$ time, with at most $\frac{5}{2}m + \frac{1}{4}n$ bends and at most $(\frac{1}{6}m + \frac{13}{12}n)^3$ volume. For 6-regular graphs, the number of bends is at most $\frac{31}{12}m$ and the volume is at most $3.97n^3$.*

PROOF. Biedl et al. [3] prove that the MEDIAN PLACEMENT algorithm, inserting the vertices in an arbitrary order, determines a vertex-ordering $(v_1, v_2, \ldots, v_n)$ with

$$\sum_{v \in V(G)} \max\{\mathrm{succ}(v), \mathrm{pred}(v)\} \;\leq\; \frac{3m}{2} + \frac{n}{4}. \tag{3}$$

If we determine a diagonal drawing of G using the BEND-MINIMUM DIAGONAL DRAWING algorithm with the vertex-ordering $(v_1, v_2, \ldots, v_n)$, by Theorem 1, the number of anchored arcs is $\sum_v \max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\}$. A degree 1 or 2 vertex v has $\max\{\text{succ}(v), \text{pred}(v)\} \geq 1$, thus

$$\max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\} \leq (\max\{\text{succ}(v), \text{pred}(v)\} - 3) + 2.$$

A degree 3 or 4 vertex v has $\max\{\text{succ}(v), \text{pred}(v)\} \geq 2$, thus

$$\max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\} \leq (\max\{\text{succ}(v), \text{pred}(v)\} - 3) + 1.$$

A degree 5 or 6 vertex v has $\max\{\text{succ}(v), \text{pred}(v)\} \geq 3$, thus

$$\max\{\max\{\text{succ}(v), \text{pred}(v)\} - 3, 0\} = \max\{\text{succ}(v), \text{pred}(v)\} - 3.$$

Let n_i be the number of vertices with degree i. The number of anchored arcs is thus at most

$$\sum_{v \in V(G)} (\max\{\text{succ}(v), \text{pred}(v)\} - 3) + 2n_1 + 2n_2 + n_3 + n_4,$$

which by (3) is at most

$$\begin{aligned}
&\frac{3m}{2} + \frac{n}{4} - 3n + 2n_1 + 2n_2 + n_3 + n_4 \\
&\qquad = \frac{m}{2} + \frac{1}{2}(n_1 + 2n_2 + 3n_3 + 4n_4 + 5n_5 + 6n_6) \\
&\qquad\quad - \frac{11}{4}(n_1 + n_2 + n_3 + n_4 + n_5 + n_6) + 2n_1 + 2n_2 + n_3 + n_4 \\
&\qquad = \frac{m}{2} + \frac{1}{4}(-n_1 + n_2 - n_3 + n_4 - n_5 + n_6) \\
&\qquad \leq \frac{m}{2} + \frac{n}{4}.
\end{aligned}$$

By Theorem 1, the drawing has at most $\frac{5}{2}m + \frac{1}{4}n$ bends, and at most $(n + \frac{1}{3}(\frac{1}{2}m + \frac{1}{4}n))^3 = (\frac{1}{6}m + \frac{13}{12}n)^3$ volume. For 6-regular graphs $m = 3n$, and the claimed bounds immediately follow. The MEDIAN PLACEMENT algorithm can be implemented in $\mathcal{O}(n)$ time [3], and by Theorem 1, the drawing itself can be determined in $\mathcal{O}(n)$ time. Thus the overall algorithm runs in $\mathcal{O}(n)$ time. $\square$

5. 3-Bend Algorithm. We now present a modification of the 3-BENDS algorithm of Eades et al. [14] that produces 3-bend diagonal drawings with $n^3 + o(n^3)$ volume. This is the best known upper bound for the volume of 3-bend drawings. In the drawings produced the vertices are not "spaced out" as in the BEND-MINIMUM DIAGONAL DRAWING algorithm; that is, vertex v_i is at (i, i, i). The box with corners at $(1, 1, 1)$ and (n, n, n), which contains all the vertices, is called the *inner box*. The same type of 2-bend edges used in the BEND-MINIMUM DIAGONAL DRAWING algorithm are again used here, but 3-bend edges are routed out of the inner box into so-called *outer boxes* illustrated in Figure 9.

This approach has the advantage that some edges routed in a particular outer box can share a grid-plane, thus reducing the volume. A similar edge routing strategy is used in

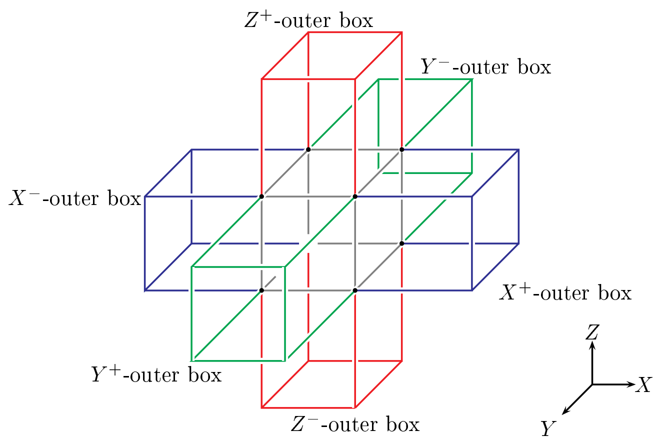

Fig. 9. Inner and outer boxes.

the algorithm of Wood [35], which, given a fixed 3-dimensional general position vertex layout, determines a 3-bend drawing.

The algorithm exploits two well-known tools from graph theory. A *cycle cover* of a directed graph is a spanning subgraph consisting of directed cycles. The following result, which can be considered as a repeated application of the classical result of Petersen [29] that "every regular graph of even degree has a 2-factor", has an algorithmic proof by Eades et al. [14].

LEMMA 2 [14]. *If G is an n-vertex* 6*-graph, then there exists a directed graph G' (possibly with loops) such that*:

1. *G is a subgraph of the underlying undirected graph of G'.*
2. *Each vertex of G' has in-degree* 3 *and out-degree* 3.
3. *The arcs of G' can be partitioned into edge-disjoint cycle covers (C_X, C_Y, C_Z).*

G' and the edge-disjoint cycle covers can be computed in $O(n)$ time.

The second tool is that of a book-embedding of a graph. A *book* consists of a line in 3-space, called the *spine*, and some number of *pages* (each a half-plane with the spine as boundary). A *book embedding* of a graph is a spatial embedding consisting of a vertex-ordering, called the *spine ordering*, along the spine of a book and an assignment of edges to pages so that edges in the same page can be drawn on that page without crossings; that is, for any two edges vw and xy, if $v < x < w < y$ in the spine ordering, then vw and xy are assigned different pages. The minimum number of pages in which a graph can be embedded is its *pagenumber*. Malitz [24] proved that the pagenumber of an m-edge graph is $\mathcal{O}(\sqrt{m})$, and presented a Las Vegas algorithm to (almost certainly) compute the book embedding in $\mathcal{O}(m \log^2 n \log\log m)$ time (also see [37]). Shahrokhi and Shi [31] gave a deterministic algorithm to compute, in $\mathcal{O}(m^2 n^{3/2})$ time, a book-embedding of a k-colourable graph with $\mathcal{O}(\sqrt{k \cdot m})$ pages. 6-Graphs have $\mathcal{O}(n)$ edges, and are 7-colourable in $\mathcal{O}(n)$ time using the sequential greedy algorithm. Thus a book-embedding of a 6-graph with $\mathcal{O}(\sqrt{n})$ pages can be determined in $\mathcal{O}(n^{5/2})$ time or (almost certainly) in $\mathcal{O}(n \log^2 n \log\log n)$ time.

The following algorithm uses the same method as the 3-BENDS algorithm of Eades et al. [14] to determine a port-assignment based on a cycle cover decomposition. It differs in the choice of a spine ordering as the diagonal ordering and in the routing of 3-bend edges in the outer boxes.

Algorithm 2. 3-BEND DIAGONAL DRAWING

Input: 6-graph G
Output: 3-bend diagonal drawing of G

1. Determine a book-embedding of G using the algorithm of [24] or [31]. Suppose $(v_1, v_2, \ldots, v_n)$ is the spine ordering, and $p : E(G) \to \{1, 2, \ldots, P\}$ is the page numbering where $P \in \mathcal{O}(\sqrt{n})$.
2. Position each vertex v_i, $1 \le i \le n$, at (i, i, i).
3. Compute G' and a cycle cover decomposition (C_X, C_Y, C_Z) of G'.
4. For dimension $I \in \{X, Y, Z\}$, for each directed cycle $Q \in C_I$, and for each directed edge $v_a v_b \in Q$ in the original graph with $v_b v_c$ the next edge in Q, route $v_a v_b$ as follows, depending on the relative values of a, b and c.

   In what follows edge routes are expressed as (I, J, K) coordinates where (a) $J = Y$ and $K = Z$ if $I = X$, (b) $J = Z$ and $K = X$ if $I = Y$ and (c) $J = X$ and $K = Y$ if $I = Z$:

   (a) If $a < b < c$, then we say $v_a v_b$ is *normal increasing*; as illustrated in Figure 10(a), route $v_a v_b$ with the 2-bend edge: $(a, a, a) \to (b, a, a) \to (b, b, a) \to (b, b, b)$.
   (b) If $a > b > c$, then we say $v_a v_b$ is *normal decreasing*; as illustrated in Figure 10(b), route $v_a v_b$ with the 2-bend edge: $(a, a, a) \to (a, a, b) \to (a, b, b) \to (b, b, b)$.
   (c) If $a < b > c$, then we say $v_a v_b$ is *increasing to a local maximum*; as illustrated in Figure 11(a), route $v_a v_b$ with the 3-bend edge: $\big(a, a, a\big) \to \big(p(v_a v_b) + n, a, a\big) \to \big(p(v_a v_b) + n, b, a\big) \to \big(p(v_a v_b) + n, b, b\big) \to \big(b, b, b\big)$.
   (d) If $a > b < c$ then we say $v_a v_b$ is *decreasing to a local minimum*; as illustrated in Figure 11(b), route $v_a v_b$ with the 3-bend edge: $\big(a, a, a\big) \to \big(a, a, 1 - p(v_a v_b)\big) \to \big(a, b, 1 - p(v_a v_b)\big) \to \big(b, b, 1 - p(v_a v_b)\big) \to \big(b, b, b\big)$.

THEOREM 5. *The algorithm* 3-BEND DIAGONAL DRAWING *determines a* 3*-bend drawing of G with $n^3 + o\big(n^3\big)$ bounding box volume. The running time is $\mathcal{O}(n^{5/2})$ using the algorithm of Shahrokhi and Shi* [31] *or* (*almost certainly*) $\mathcal{O}(n \log^2 n \log\log n)$ *using the algorithm of Malitz* [24].

PROOF. A normal increasing edge is followed in its cycle by an increasing edge, a normal decreasing edge is followed by a decreasing edge, an edge that is increasing to a local maximum is followed by a decreasing edge, and an edge that is decreasing to a

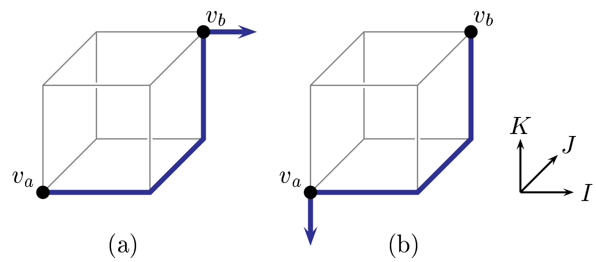

Fig. 10. Normal 2-bend edge routes: (a) increasing and (b) decreasing.

local minimum is followed by an increasing edge. Hence it is easily checked that the two edges incident to a vertex v in the cycle cover C_I use the I_v^+ and K_v^- ports at v. Therefore each edge is assigned unique ports at its end-vertices.

We call an edge segment incident to a vertex v a v-*segment*. The segment of a 2-bend edge not incident to a vertex is called an *inner middle* segment, and the two segments of a 3-bend edge not incident to a vertex are called *outer middle* segments.

Clearly, two v-segments for some vertex v do not cross. A v-segment and a w-segment for some distinct vertices v and w do not cross as otherwise, as illustrated in Figure 12(a), v and w would be in a common grid-plane. As illustrated in Figure 12(b), a v-segment and an inner middle segment of an edge incident to v do not cross. As illustrated in Figure 12(c), a v-segment and an inner middle segment of an edge uw not incident to v do not cross as otherwise v would be in a grid-plane containing u or w. In each outer box there is at most one edge incident to a vertex v. Hence a v-segment does not cross an outer middle segment of an edge incident to v. As illustrated in Figure 12(d), a v-segment and an outer middle segment of an edge uw not incident to v do not cross as otherwise v would be in a grid-plane containing u or w.

By the same argument used in the proof of Theorem 1, no two inner middle segments cross. An inner middle segment is contained within the inner box, and an outer middle segment is contained within one of the outer boxes, and hence do not cross. Two outer middle segments could only cross if routed in the same outer box and with their corresponding edges in the same page of the book embedding. However, crossing edges

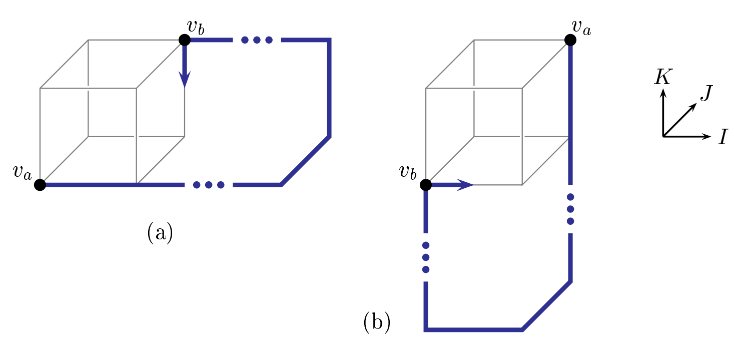

Fig. 11. 3-Bend edge routes: (a) increasing to a local maximum and (b) decreasing to a local minimum.

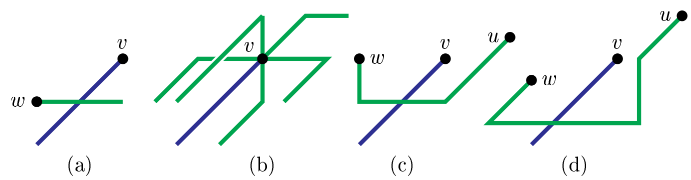

Fig. 12. Intersecting v-segment.

routed in the same page would also intersect in the book embedding, as illustrated in Figure 13. Hence there are no edge crossings.

The bounding box is $(P+n+P)\times(P+n+P)\times(P+n+P)$. Since $P\in\mathcal{O}(\sqrt{n})$ the volume is $(n+\mathcal{O}(\sqrt{n}))^3 = n^3+\mathcal{O}(n^{5/2}) = n^3+o(n^3)$. All steps except for Step 1 run in $\mathcal{O}(n)$ time. The claimed running times follow from the previous discussion. □

Obviously the volume of a diagonal drawing is at least n^3. Hence the 3-Bend Diagonal Drawing algorithm produces diagonal drawings with optimal volume, up to an additive lower-order term.

6. Conclusion. In this paper we have studied 3-dimensional orthogonal graph drawings with the vertices positioned along the main diagonal of a cube. Minimising the number of bends in so-called diagonal drawings is NP-hard. We present two algorithms for producing a diagonal drawing. The first minimises the total number of bends for a fixed ordering of the vertices along the diagonal. Using two heuristics for determining this vertex-ordering we obtain upper bounds on the number of bends. For simple graphs with average degree at least 5, it is easily seen that the upper bound on the number of bends in a diagonal drawing gained by applying the Median Placement algorithm (see Theorem 4) is lower than the upper bound gained by using an st-ordering (see Theorem 3). Our second algorithm produces 3-bend drawings with $n^3+o(n^3)$ volume,

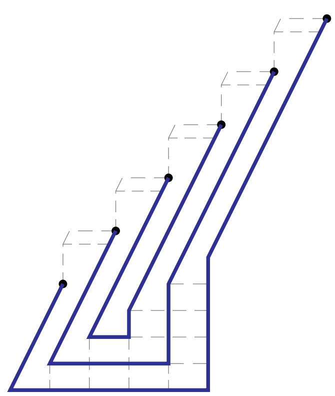

Fig. 13. Edges in the same page that are also routed in the same outer box.

which is the best known upper bound for the volume of 3-dimensional orthogonal graph drawings with at most three bends per edge.

Acknowledgements. The author gratefully acknowledges the invaluable advice of his Ph.D. supervisor Graham Farr. Many thanks to Therese Biedl for fruitful discussions, especially regarding the proofs of Theorems 3 and 4.

References

[1] A. Aggarwal, M. Klawe, and P. Shor, Multilayer grid embeddings for VLSI. *Algorithmica*, **6**(1):129–151, 1991.

[2] B. Baetz and D. R. Wood, Brooks' Vertex-Colouring Theorem in Linear Time. Technical Report CS-AAG-2001-05, Basser Department of Computer Science, The University of Sydney, 2001.

[3] T. Biedl, T. Chan, Y. Ganjali, M. Hajiaghayi, and D. R. Wood, Balanced vertex-orderings of graphs, submitted. See Technical Report TR-2002-01, School of Computer Science, Carleton University, Ottawa, 2002.

[4] T. Biedl, J. R. Johansen, T. Shermer, and D. R. Wood, Orthogonal drawings with few layers. In [27], pp. 297–311.

[5] T. Biedl, T. Thiele, and D. R. Wood, Three-dimensional orthogonal graph drawing with optimal volume. In [25], pp. 284–295.

[6] T. C. Biedl, Heuristics for 3D-orthogonal graph drawings. In *Proc. 4th Twente Workshop on Graphs and Combinatorial Optimization*, pp. 41–44, 1995.

[7] T. C. Biedl, Three approaches to 3D-orthogonal box-drawings. In S. Whitesides, ed., *Proc. 6th International Symp. on Graph Drawing* (*GD '98*), vol. 1547 of Lecture Notes in Computer Science, pp. 30–43. Springer-Verlag, Berlin, 1998.

[8] T. C. Biedl and M. Kaufmann, Area-efficient static and incremental graph drawings. In R. Burkhard and G. Woeginger, eds., *Proc. 5th Ann. European Symp. on Algorithms* (*ESA '97*), vol. 1284 of Lecture Notes in Computer Science, pp. 37–52. Springer-Verlag, Berlin, 1997.

[9] R. L. Brooks, On colouring the nodes of a network. *Proc. Cambridge Philos. Soc.*, **37**:194–197, 1941.

[10] M. Closson, S. Gartshore, J. Johansen, and S. K. Wismath, Fully dynamic 3-dimensional orthogonal graph drawing. *J. Graph Algorithms Appl.*, **5**(2):1–34, 2001.

[11] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Englewood Cliffs, NJ, 1999.

[12] G. Di Battista, M. Patrignani, and F. Vargiu, A split&push approach to 3D orthogonal drawing. *J. Graph Algorithms Appl.*, **4**(3):105–133, 2000.

[13] P. Eades, C. Stirk, and S. Whitesides, The techniques of Kolmogorov and Barzdin for three dimensional orthogonal graph drawings. *Inform. Process. Lett.*, **60**(2):97–103, 1996.

[14] P. Eades, A. Symvonis, and S. Whitesides, Three dimensional orthogonal graph drawing algorithms. *Discrete Applied Math.*, **103**:55–87, 2000.

[15] S. Even and R. E. Tarjan, Computing an st-numbering. *Theoret. Comput. Sci.*, **2**(3):339–344, 1976.

[16] K. Hagihara, N. Tokura, and N. Suzuki, Graph embedding on a three-dimensional model. *Systems-Comput.-Controls*, **14**(6):58–66, 1983.

[17] M. Kaufmann and D. Wagner, eds., *Drawing Graphs: Methods and Models*, vol. 2025 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2001.

[18] N. D. Kazarinoff, *Analytic Inequalities*. Holt, Rinehart and Winston, New York, 1961.

[19] A. N. Kolmogorov and Y. M. Barzdin, On the realization of nets in 3-dimensional space. *Probl. Cybernet.*, **8**:261–268, 1967.

[20] F. T. Leighton and A. L. Rosenberg, Three-dimensional circuit layouts. *SIAM J. Comput.*, **15**(3):793–813, 1986.

[21] A. Lempel, S. Even, and I. Cederbaum, An algorithm for planarity testing of graphs. In *Proc. Internat. Symp. on Theory of Graphs*, pp. 215–232. Gordon and Breach, New York, 1967.

[22] L. Lovász, Three short proofs in graph theory. *J. Combin. Theory Ser. B*, **19**:269–271, 1975.
[23] B. Y. S. Lynn, A. Symvonis, and D. R. Wood, Refinement of three-dimensional orthogonal graph drawings. In [25], pp. 308–320.
[24] S. M. Malitz, Graphs with E edges have pagenumber $O(\sqrt{E})$. *J. Algorithms*, **17**(1):71–84, 1994.
[25] J. Marks, ed., *Proc. 8th Internat. Symp. on Graph Drawing* (*GD* '00), vol. 1984 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2001.
[26] K. Misue, P. Eades, W. Lai, and K. Sugiyama, Layout adjustment and the mental map. *J. Visual Languages Comput.*, **6**:183–210, 1995.
[27] P. Mutzel, M. Jünger, and S. Leipert, eds., *Proc. 9th Internat. Symp. on Graph Drawing* (*GD* '01), vol. 2265 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2002.
[28] A. Papakostas and I. G. Tollis, Algorithms for incremental orthogonal graph drawing in three dimensions. *J. Graph Algorithms Appl.*, **3**(4):81–115, 1999.
[29] J. Petersen, Die Theorie der regulären Graphen. *Acta. Math.*, **15**:193–220, 1891.
[30] A. L. Rosenberg, Three-dimensional VLSI: A case study. *J. Assoc. Comput. Mach.*, **30**(2):397–416, 1983.
[31] F. Shahrokhi and W. Shi, On crossing sets, disjoint sets, and pagenumber. *J. Algorithms*, **34**(1):40–53, 2000.
[32] C. Ware and G. Franck, Viewing a graph in a virtual reality display is three times as good as a 2D diagram. In A. L. Ambler and T. D. Kimura, eds., *Proc. IEEE Symp. Visual Languages* (*VL* '94), pp. 182–183. IEEE, New York, 1994.
[33] C. Ware and G. Franck, Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graphics*, **15**(2):121–140, 1996.
[34] D. R. Wood, Multi-dimensional orthogonal graph drawing with small boxes. In J. Kratochvil, ed., *Proc. 7th Internat. Symp. on Graph Drawing* (*GD* '99), vol. 1731 of Lecture Notes in Computer Science, pp. 311–222. Springer-Verlag, Berlin, 1999.
[35] D. R. Wood, Three-Dimensional Orthogonal Graph Drawing. Ph.D. thesis, School of Computer Science and Software Engineering, Monash University, Melbourne, 2000.
[36] D. R. Wood, Bounded degree book embeddings and three-dimensional orthogonal graph drawing. In [27], pp. 312–327.
[37] D. R. Wood, Degree constrained book embeddings. *J. Algorithms*, **45**(2):144–154, 2002.
[38] D. R. Wood, Lower bounds for the number of bends in three-dimensional orthogonal graph drawings. *J. Graph Algorithms Appl.*, **7**(1):33–77, 2003.
[39] D. R. Wood, Optimal three-dimensional orthogonal graph drawing in the general position model. *Theoret. Comput. Sci.*, **299**(1-3):151–178, 2003.