An Algorithm for Three-Dimensional Orthogonal Graph Drawing

David R. Wood

School of Computer Science and Software Engineering, Monash University
Wellington Road, Clayton, VIC 3168, Australia
davidw@csse.monash.edu.au

Abstract. In this paper we present an algorithm for 3-dimensional orthogonal graph drawing based on the movement of vertices from an initial layout along the main diagonal of a cube. For an n-vertex m-edge graph with maximum degree six, the algorithm produces drawings with bounding box volume at most $2.37n^3$ and with a total of $7m/3$ bends, using no more than 4 bends per edge route. For maximum degree five graphs the bounding box has volume n^3 and each edge route has two bends. These results establish new bounds for 3-dimensional orthogonal graph drawing algorithms and improve on some existing bounds.

1 Introduction

Prompted by advances in graphics workstations and applications including VLSI circuit design [4, 20, 22] and software engineering [15, 21], there has been recent interest in graph visualisation in 3-dimensional space. Proposed models include straight-line drawings [6, 13, 17], convex drawings [5, 8], spline curve drawings [14], multilevel drawings of clustered graphs [7], visibility representations [1, 12] and of interest in this paper orthogonal grid drawings [2, 9, 10, 11, 16, 18, 23, 24, 25].

The *3-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. An *orthogonal grid drawing* of a graph places the vertices at grid points and routes the edges along sequences of contiguous segments of grid lines. Edge routes are allowed to contain bends and can only intersect at a common vertex.

We shall refer to the 3-dimensional orthogonal grid as the *grid* and an orthogonal grid drawing with no more than b bends per edge as simply a *b-bend grid drawing.* At a vertex v the six directions, or *ports*, the edges incident with v can use are denoted X_v^+, X_v^-, Y_v^+, Y_v^-, Z_v^+ and Z_v^-. Clearly, grid drawings can only exist for graphs with maximum degree six. Figure 1 shows an example of a 2-bend grid drawing.

The grid has been extended to higher dimensions [23], and by representing a vertex by a cube for example, 3-dimensional grid drawing of arbitrary degree graphs has also been considered [2, 18].

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 332–346, 1998.

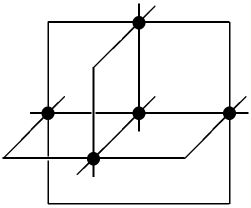


Fig. 1. A 2-bend grid drawing of K_5.

The most common proposed measures for determining a 'good' grid drawing have been the bounding box volume and the maximum number of bends per edge route. Using straightforward extensions of the corresponding 2-dimensional NP-hardness results, optimising any of these criteria is NP-hard [9]. In this paper we look at the total number of bends as a further measure of the 'goodness' of a grid drawing.

Table 1 shows the tradeoff between the bounding box volume and the maximum number of bends per edge apparent in the existing grid drawing algorithms. The algorithms of Eades, Symvonis and Whitesides [11] are denoted ESW1-4. (Their *compact* (ESW1) and *3-bends* (ESW4) algorithms originally appeared in [10].) The linear time incremental algorithm of Papakostas and Tollis [18] is denoted PT. DW refers to the algorithm presented in this paper. Also of interest is the *Reduce Forks* algorithm of Patrignani and Vargiu [19] which in experimental tests averaged less than 2.5 bends per edge route with bounding box volume $0.6n^3$.

Table 1. Upper Bounds of Grid Drawing Algorithms

	ESW1	ESW2	ESW3	DW	PT	ESW4	DW ($\Delta \leq 5$)
volume	$O(n^{3/2})$	$O(n^2)$	$O(n^{5/2})$	$2.37n^3$	$4.66n^3$	$27n^3$	n^3
max bends	7	6	5	4	3	3	2

Algorithm ESW1 requires the least volume at the cost of more bends per edge route. In fact, this volume bound is tight. Kolmogorov and Barzdin [16] and Rosenberg [22] establish a lower bound of $\Omega(n^{3/2})$ for the bounding box volume. Algorithms ESW4 and PT establish an upper bound of three for the maximum number of bends per edge route. Eades, Symvonis and Whitesides [10, 11] had conjectured that there does not exist a 2-bend grid drawing of K_7. Wood [23] presents a counterexample to this conjecture. 2-bend grid drawings of the other multi-partite 6-regular graphs $K_{6,6}$, $K_{3,3,3}$ and $K_{2,2,2,2}$ have also been found [25]. While it is unknown if the upper bound of 3 for the maximum number of bends per edge route is tight, it is easily seen that K_5 provides a lower bound of two [23]. The following open problem is therefore of interest:

2-Bends Problem: Does every maximum degree six graph admit a 2-bend grid drawing?

In this paper we solve the 2-bends problem for maximum degree five graphs. Our algorithm, applied to an n-vertex m-edge graph with maximum degree six, produces a $2.37n^3$-volume 4-bend grid drawing with at most $7m/3$ bends. The algorithm positions the vertices along the diagonal of a cube according to an 'approximately balanced' ordering. From there vertices are moved in up to two dimensions and non-intersecting edge routes determined.

Section 2 of this paper introduces balanced orderings. Section 3 describes a model for grid drawing, the types of edge routes used and how to avoid edge route intersections. Following the presentation of the algorithm in Section 4, we conclude by comparing its performance with the existing algorithms.

Throughout this paper G is an n-vertex m-edge undirected simple graph with maximum degree six. We define the directed graph G' with vertex set $V(G') = V(G)$ and two arcs $(v, w), (w, v) \in A(G')$ for each edge $\{v, w\} \in E(G)$. The arc (v, w) is called the *reversal* of (w, v). We use the notation vw to represent the edge $\{v, w\}$, the directed arc (v, w) or the edge route for $\{v, w\}$. The port at v used by an edge route vw is referred to as the port *assigned* to the arc vw. A total ordering $<$ of $V(G)$ induces a numbering $(v_1, v_2, \ldots, v_n)$ of $V(G)$ and vice versa. We shall refer to both $<$ and $(v_1, v_2, \ldots, v_n)$ as an *ordering* of $V(G)$.

2 Balanced Orderings

Given an ordering $<$ on $V(G)$, if $vw \in E(G)$ with $v < w$ we say v is a *predecessor* of w and w is a *successor* of v; vw is a *successor arc* of v and wv is a *predecessor arc* of w. The number of predecessors and successors of a vertex v are denoted p_v and s_v respectively. v has *cost* $c_v = |s_v - p_v|$. Note that a vertex has even cost iff it has even degree. The *total cost* of G with respect to a given ordering is the sum of the cost of each vertex. If $a = \min(s_v, p_v)$ and $b = \max(s_v, p_v)$ then v is said to be an (a, b)-vertex.

The *3-bends* algorithm of Eades, Symvonis and Whitesides [10, 11] positions the vertices along the diagonal of a cube according to an arbitrary ordering. Under this model a 2-bend grid drawing is possible iff each vertex v has $p_v \leq 3$ and $s_v \leq 3$ [24], in which case we say v is *balanced.*

We say v is *positive* if $s_v > p_v$ and *negative* if $p_v > s_v$. Clearly, if $s_v = p_v$ then v is balanced. For positive and balanced vertices v and for $k > 0$ (respectively, $k < 0$) v^k denotes the k^{th} successor (predecessor) of v to the right (left) of v in the ordering. For negative v and for $k > 0$ (respectively $k < 0$) v^k denotes the k^{th} predecessor (successor) of v to the left (right) of v in the ordering. Two adjacent vertices v, w with $v < w$ are *opposite* if v is positive and w is negative.

Given an arbitrary ordering of $V(G)$ we apply the following three rules of movement to each pair of opposite vertices v, w. Moving an unbalanced vertex v past v^i for some i $(1 \leq i \leq \lfloor c_v/2 \rfloor)$ reduces c_v by $2i$. It follows that each rule, when executed, reduces the total cost of the ordering.

M1 If $w = v^i$ for some i $(1 \leq i \leq \lfloor c_v/2 \rfloor)$ then move v to immediately past v^i, as in Figure 2.

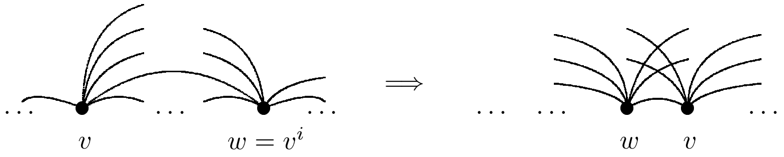


Fig. 2. The move M1 for a (1,5)-vertex v and a (2,4)-vertex w with $i = 2$.

M2 If $v < w^j < v^i < w$ for some i, j $(1 \leq i \leq \lfloor c_v/2 \rfloor,\ 1 \leq j \leq \lfloor c_w/2 \rfloor)$, then move v up to v^i and move w up to w^j, as in Figure 3.

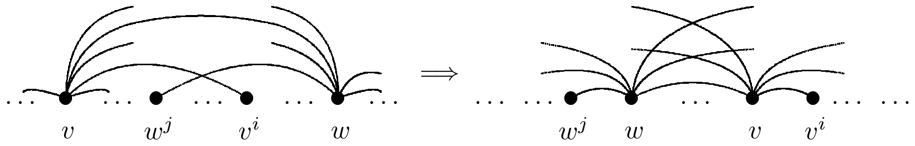


Fig. 3. The move M2 for a (1,5)-vertex v and a (2,4)-vertex w with $i = 2$ and $j = 1$.

M3 If $v < v^i = w^j < w$ for some i, j $(1 \leq i \leq \lfloor (c_v - 1)/2 \rfloor,\ 1 \leq j \leq \lfloor (c_w - 1)/2 \rfloor)$ then move v to immediately past v^i and move w to immediately past w^j, as in Figure 4 .

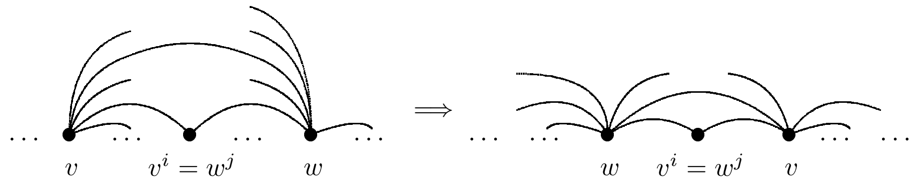


Fig. 4. The move M3 for a (0,5)-vertex v and a (1,5)-vertex w with $i = 2$ and $j = 1$.

Applying M1, M2 or M3 reduces c_v by at least $2i$ and for each k $(1 \leq k \leq i-1)$ c_{v^k} is increased by at most two. For M2 and M3, c_w is reduced by at least $2j$ and for each k $(1 \leq k \leq j-1)$ c_{w^k} is increased by at most two. The cost of all other vertices remains unchanged. Thus for M1 the total cost decreases by at least two, and by at least four for M2 and M3. An ordering of $V(G)$ is said to be *approximately balanced* if the rules M1, M2 and M3 cannot be applied.

3 The 'Unique Coordinates' Model

We now present a model for grid drawing where each vertex has a unique X, a unique Y and a unique Z coordinate. Each edge route has at least two bends, and a 2-bend edge route must have three perpendicular segments, as in Figure 5.

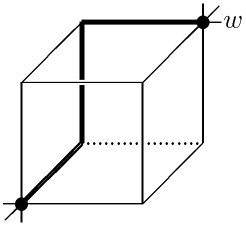


Fig. 5. 2-bend edge route vw

We shall also use particular types of 3- and 4-bend edge routes. A 3-bend edge route vw, said to be *anchored* at v, consists of a unit length segment from v followed by a 2-bend edge route to w, as in Figure 6.

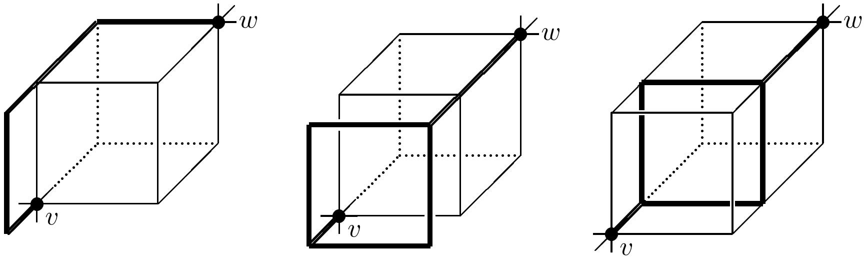


Fig. 6. 3-bend edge routes vw anchored at v

A 4-bend edge route vw, said to be *anchored* at v and at w, consists of unit length segments at v and at w with a 2-bend edge route in between, as in Figure 7.

We represent the relative coordinates of the vertices by three orderings $<_X$, $<_Y$ and $<_Z$ of $V(G)$, denoted the X-, Y- and Z-orderings. Port assignments are represented by a 3-colouring of $A(G')$ using colours $\{X, Y, Z\}$. An arc vw coloured $I \in \{X, Y, Z\}$ uses the I_v^+ or I_v^- port. We also maintain a set of anchored arcs of G' (to be specified in Section 4), such that an arc vw is anchored iff the edge route vw is anchored at v. The ports at v and w used by a 2-bend edge route vw must be perpendicular and point towards w and v respectively. Therefore the 3-colouring of $A(G')$ must satisfy:

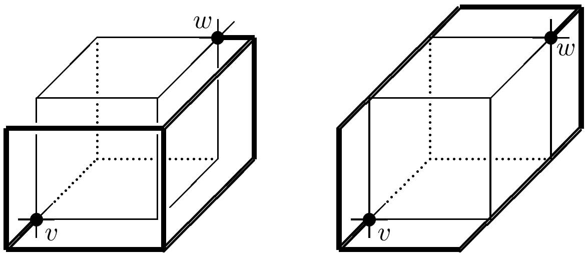


Fig. 7. 4-bend edge routes vw anchored at v and at w

- If neither an arc vw nor its reversal arc wv are anchored then they are coloured differently. (1)

- For each vertex v, for each $I \in \{X, Y, Z\}$, there are at most two outgoing arcs vu, vw coloured I, and if neither vu nor vw are anchored then $u <_I v <_I w$ or $w <_I v <_I u$. (2)

Theorem 1. *Let G be a graph. Suppose there exists X-, Y- and Z-orderings of $V(G)$, a set of k anchored arcs of G', and a 3-colouring of $A(G')$ satisfying (1) and (2). Then there exists an $(n + k/3)^3$-volume 4-bend grid drawing of G with at most $2m + k$ bends. Furthermore if $k = 0$ then all edge routes have two bends.*

Proof: Initially position each vertex v at (v_X, v_Y, v_Z) where v_I is the index of v in the I-ordering. Suppose the arc $vw \in A(G')$ is coloured $I \in \{X, Y, Z\}$ and its reversal arc wv is coloured $J \in \{X, Y, Z\}$. Let vu be the other outgoing arc at v (if any) also coloured I.

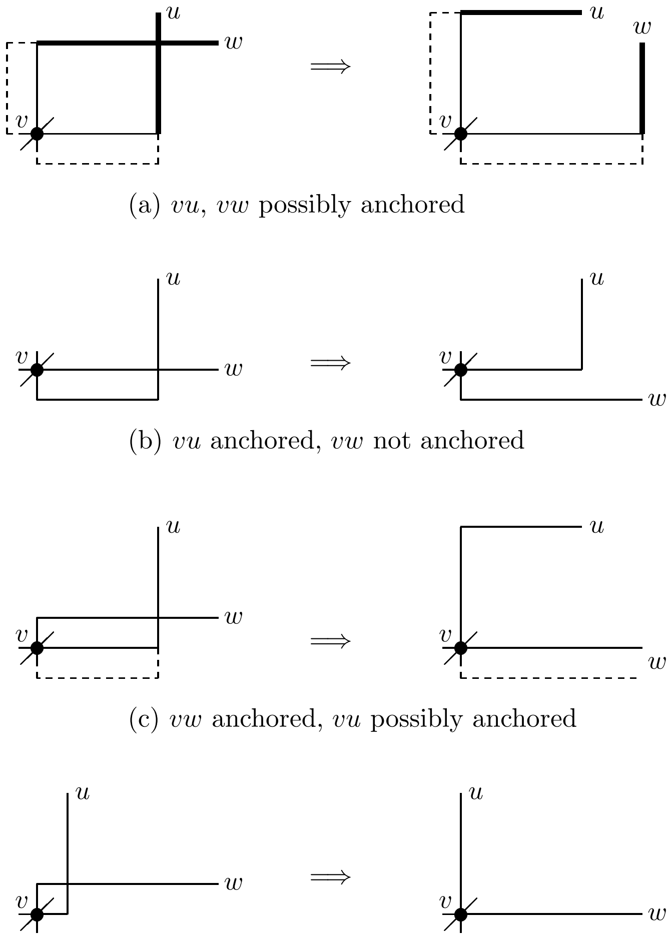
If vw is not anchored then assign vw the port I_v^+ if $v <_I w$ and I_v^- if $w <_I v$. If vu is also not anchored then by (2) vu will be assigned the opposite port to vw. If neither vw nor wv are anchored then by (1) $I \neq J$, and we route vw with two bends.

Suppose vw is anchored. If vu is not anchored then assign vw the opposite port to vu, otherwise the ports I_v^+ and I_v^- can be arbitrarily assigned to vu and vw. If wv is not anchored then route the edge vw with a 3-bend edge route anchored at v. If vw and wv are both anchored then route vw with a 4-bend edge route.

For each anchored arc vw coloured I insert a plane at v perpendicular to the I-axis so that the unit length segment of the edge route vw lies between v and the inserted plane. The plane is considered to have an I-coordinate unique to v.

A grid point on an edge route vw has two coordinates unique to v, one coordinate unique with v and one with w, or two coordinates unique to w. Therefore edge routes can only intersect if they are incident at a common vertex, and these edge routes, say vu and vw, must intersect as in Figure 8.

In each case, swapping the ports at v assigned to vu and vw reroutes the edges so that they no longer intersect. In (b) and (a) (if exactly one arc is anchored)



(a) vu, vw possibly anchored

(b) vu anchored, vw not anchored

(c) vw anchored, vu possibly anchored

(d) vu, vw both anchored

Fig. 8. Rerouting intersecting edge routes

the anchor is transfered to the other arc. In (a) and (c) swapping ports may create new edge route intersections between uv and another edge incident at u, or similarly at w. However in (a) the sum of the lengths of the middle segments of vu and vw is reduced (see the highlighted segments), and in (c) (and also in (d)) at least one anchored arc (and thus a bend) is eliminated. Since this sum and the number of bends is bounded below, by applying (a), (c) and (d) before (b) a finite number of swaps suffice for an intersection-free grid drawing.

The bounding box has volume $(n + k_X) \times (n + k_Y) \times (n + k_Z)$, where k_I is the number of anchored arcs coloured I. It is easily seen that the bounding box volume is maximised when it is a cube; i.e. $k_X = k_Y = k_Z = k/3$. Therefore the bounding box volume is at most $(n + k/3)^3$. An unanchored edge route has two bends, and each anchored arc contributes one further bend. Therefore the total number of bends is $2m+k$, and if $k = 0$ then only 2-bend edge routes are used. ■

The algorithm described in the next section shall apply Theorem 1 with one anchored arc for each degree six vertex.

4 The Algorithm

Our grid drawing algorithm initially positions the vertices along the main diagonal of a cube according to an approximately balanced ordering. At a balanced vertex v the positive (respectively, negative) ports are assigned to the successor (predecessor) arcs of v. However, at an unbalanced, say positive, vertex v the positive ports can be assigned to at most three successor arcs of v. The remaining successor arcs vw must be assigned a negative port at v. To do so we can anchor the arc vw (as referred to in Section 3), or move v past w in some ordering, in which case vw is said to be a movement arc. Table 2 defines the *movement* and *anchored* arcs for each type of unbalanced vertex.

Table 2. Definition of movement and anchored arcs for unbalanced vertices

v	(0,4)	(1,4)	(0,5)	(2,4)	(1,5)	(0,6)
vv^1	movement	movement	movement	anchored	movement	movement
vv^2	-	-	movement	-	anchored	movement
vv^3	-	-	-	-	-	anchored

If vw is a movement arc coloured I then v is moved to immediately past w in the I-ordering, thus allowing vw to be assigned the I_v^- port for positive v and the I_v^+ port for negative v. In Figure 9 we illustrate the movement and anchoring process in the case of a positive (0,6)-vertex.

For an unbalanced vertex v, if $vw = vv^i$ is a movement or anchored arc then $i \leq \lfloor c_v/2 \rfloor$, so rule M1 is applicable. Therefore w cannot be opposite to v, and hence wv cannot also be a movement or anchored arc. Consequently when edges are routed no 4-bend edge routes are immediately constructed. It is only through swapping ports to avoid intersections that a 4-bend edge route can be introduced. Furthermore, if vv^i is a movement arc then $i \leq \lfloor (c_v - 1)/2 \rfloor$, so by rules M2 and M3, if v and w are opposite unbalanced vertices then the movement arcs of v do not 'cross over' or have the same destination vertex as the movement arcs of w.

To represent the 3-colouring of $A(G')$, we construct a graph G'' with vertex set $V(G'') = A(G')$. Vertices are adjacent in G'' if the corresponding arcs must use non-parallel ports. We distinguish four types of edges of G'':

1. The first type of edge ensures that arcs which 'compete' for the same ports are coloured differently. In Table 3 we define the arcs vv^A, vv^B, vv^C, vv^D,

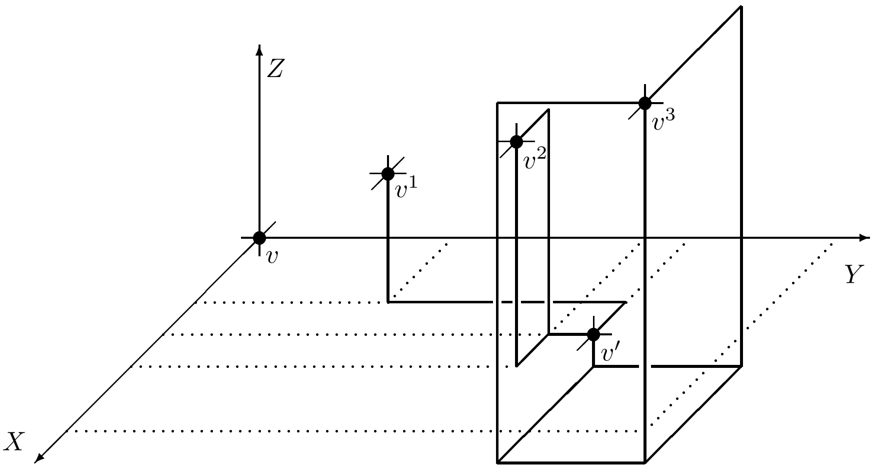


Fig. 9. v is a positive (0,6)-vertex, vv^1 is a movement arc coloured X, vv^2 is a movement arc coloured Y, vv^3 is an anchored arc coloured Z; move v to v'.

vv^E and vv^F for each type of vertex. If v is a balanced or a positive (respectively, negative) unbalanced vertex then vv^A, vv^B and vv^C will be assigned the negative (positive) ports at v. The arcs vv^D, vv^E and vv^F will be assigned the positive (negative) ports at v. Thus vv^A, vv^B and vv^C are pairwise adjacent in G'', and vv^D, vv^E and vv^F are pairwise adjacent in G''.

Table 3. Definition of vv^A, vv^B, vv^C, vv^D, vv^E and vv^F

v	vv^A	vv^B	vv^C	vv^D	vv^E	vv^F
balanced	vv^{-3}	vv^{-2}	vv^{-1}	vv^1	vv^2	vv^3
(0,4)-vertex	vv^1	-	-	vv^2	vv^3	vv^4
(1,4)-vertex	vv^{-1}	vv^1	-	vv^2	vv^3	vv^4
(2,4)-vertex	vv^{-2}	vv^{-1}	vv^1	vv^2	vv^3	vv^4
(0,5)-vertex	vv^1	vv^2	-	vv^3	vv^4	vv^5
(1,5)-vertex	vv^{-1}	vv^1	vv^2	vv^3	vv^4	vv^5
(0,6)-vertex	vv^1	vv^2	vv^3	vv^4	vv^5	vv^6

2. If neither the arc vw not its reversal arc wv are anchored then add the edge $\{vw, wv\}$ (labelled ‘r’) to G''.

3. If vw and wx are both movement arcs for some vertices v, w and x, then add the edge $\{vw, wx\}$ (labelled '$*$') to G''. This ensures that v and w do not move in the same ordering.

4. If vv^2 is a movement arc coloured I then v will move past v^1 in the I-ordering. To ensure that v^1v does not use the incorrect I_{v^1} port add the edge $\{vv^2, v^1v\}$ (labelled '$**$') to G''. Note that in Figure 9 v^1v cannot use the port $J^+_{v^1}$.

We now summarise our algorithm.

1. Determine an approximately balanced ordering $(v_1, v_2, \ldots, v_n)$ of $V(G)$.
2. Initialise the X-, Y- and Z-orderings to be $(v_1, v_2, \ldots, v_n)$.
3. Construct and 3-colour the graph G'' with colours $\{X, Y, Z\}$.
4. For each movement arc vw coloured $I \in \{X, Y, Z\}$, move v to immediately past w in the I-ordering.
5. Position each vertex v at (v_X, v_Y, v_Z).
6. For each anchored arc vw coloured I, insert a plane at v perpendicular to the I-axis.
7. Route the edges and remove edge route intersections.

Theorem 2. *For a simple graph G with maximum degree six, the above algorithm will determine a 4-bend grid drawing of G with bounding box volume $2.37n^3$ using at most $7m/3$ bends. If G has maximum degree five then the bounding box has volume n^3 and each edge route has two bends.*

Proof: To prove that the graph G'' is 3-colourable we employ two operations which preserve the 3-colourability of a graph. Firstly, a degree one or two vertex v and its incident edges can be removed; v can be later coloured with the colour different from its neighbours. Secondly, if $K_4 \setminus \{v, w\}$ is a subgraph for some non-adjacent vertices v and w, then in any 3-colouring v and w must receive the same colour, so we merge these vertices and replace any multiple edges by a single edge. We shall now show that this process can be continued until G'' has maximum degree three, and is not K_4, so by Brooks' Theorem [3] is 3-colourable.

For an unbalanced vertex v, let H_v be the subgraph of G'' consisting of the vertices vv^A, vv^B and vv^C and their incident edges. We shall initially show that H_v 'reduces' to a maximum degree three subgraph.

For a degree six unbalanced vertex v, the vertex of G'' corresponding to the anchored arc vv^C is incident with at most two (unlabelled) edges, and therefore can be removed from G''. Since a (0,6)-vertex and a (0,5)-vertex v both have vv^A and vv^B as movement arcs, H_v is the same for a (0,6)-vertex v (after removing vv^C) and for a (0,5)-vertex v (see Figures 10 and 11). Similarly, for (1,5)- and (2,4)-vertices, H_v is the same as for (1,4)- and (2,3)- vertices respectively. We therefore need only consider (0,5)-, (1,4)- or (0,4)- unbalanced vertices.

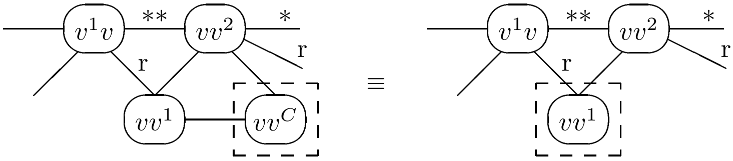


Fig. 10. The subgraph H_v for a (0,5)-vertex or a (0,6)-vertex v with v^1 balanced.

Consider a (0,5)-vertex v. v^1 may be balanced or a (1,4)-vertex. If v^1 is balanced then, as in Figure 10, vv^1 has degree two and can be removed. In the remaining graph vv^2 and v^1v have degree three.

Now, if v^1 is a (1,4)-vertex then, as in Figure 11, vv^2 and $v^1(v^1)^1$ are the non-adjacent vertices in a $K_4 \setminus \{e\}$ subgraph. If we merge these vertices then v^1v and vv^1 have degree two and can be removed. If v^2 is balanced then there is no edge $\{vv^2, v^2(v^2)^1\}$ (labelled '$*$'). If v^2 is unbalanced then v^2 must be a (1,4)-vertex, and therefore v^2v and the edge $\{vv^2, v^2v\}$ (labelled 'r') will be removed (see Figure 12). In either case vv^2 (=$v^1(v^1)^1$) has degree three.

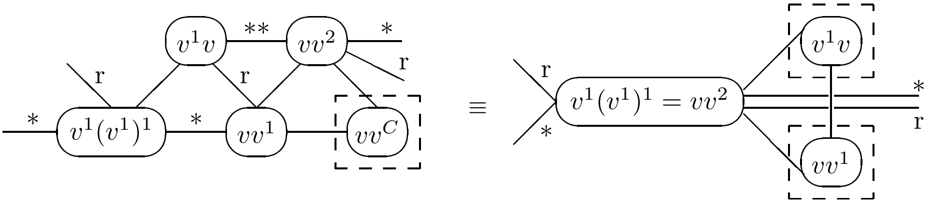


Fig. 11. The subgraph H_v for a (0,5)-vertex or a (0,6)-vertex v with v^1 a (1,4)-vertex.

Consider a (1,4)-vertex v and assume that v^{-1} is not a (0,5)-vertex with $(v^{-1})^1 = v$ (we have already considered this case). As in Figure 12, the vertex vv^{-1} has degree two and can be removed. vv^1 now has degree at most three. For a (0,4)-vertex v, H_v simply consists of the degree one vertex vv^1, which can be removed.

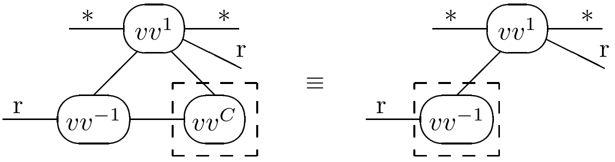


Fig. 12. The subgraph H_v of G'' for a (1,4)-vertex or a (1,5)-vertex v.

Consider a vertex $vv^i \in V(G'')$ for some $i \in \{D, E, F\}$ or $i \in \{A, B, C\}$ if v is balanced. vv^i is incident with at most two unlabelled edges and to at most one edge labelled 'r'. Unless v^i is a (0,5)- or (0,6)-vertex and $(v^i)^1 = v$ (in which case vv^i is incident with an edge labelled '$**$' and has already been considered), vv^i has degree at most three.

We have shown that all remaining vertices in G'' have degree at most three, and it is easily seen that G'' is not K_4, by Brooks' Theorem [3], G'' is 3-colourable. We shall now show that this 3-colouring satisfies the conditions of Theorem 1.

The edges of G'' labelled 'r' guarantee condition (1) holds. The unlabelled edges in G'' ensure that at most two outgoing arcs at each vertex v receive the same colour. Suppose the arcs $vu \in \{vv^A, vv^B, vv^C\}$ and $vw \in \{vv^D, vv^E, vv^F\}$ are coloured $I \in \{X, Y, Z\}$, and vu and vw are both not anchored.

If vu is not a movement arc then v is between u and w in the initial ordering, and v does not move past u or w in any ordering. If u moves past v then it does so in the same ordering as the colour assigned to the movement arc uv. Since non-anchored reversal arcs are coloured differently uv is not coloured I, so u does not move in the I-ordering. Otherwise, if vu is a movement arc then u is between v and w in the initial ordering. In the I-ordering v moves past u and, since the movement arcs of w (if any) do not cross over or have the same destination vertex as vu, w cannot move past u in any ordering. Therefore v is between u and w in the I-ordering, and condition (2) holds.

We have thus shown that Theorem 1 is applicable. If k is the number of unbalanced degree six vertices (and therefore the number of anchored arcs) then Theorem 1 asserts G has a 4-bend grid drawing with bounding box volume $(n + k/3)^3$ and $2m + k$ bends. Since $k \leq n$ the bounding box volume is at most $(n + n/3)^3 = (4n/3)^3 \leq 2.37n^3$. If d is the average degree of those vertices without anchored outgoing arcs then $6k + d(n - k) = 2m$ and the number of bends is $2m + (2m - d(n - k))/6 = 7m/3 - d(n - k)/6$. Since $n \geq k$ the grid drawing has at most $7m/3$ total bends.

For maximum degree five graphs, no anchored arcs are introduced. By Theorem 1 the bounding box has volume n^3 and each edge route has two bends. ■

5 Experimental Results

In Table 4 we compare the performance of existing grid drawing algorithms with the algorithm presented in this paper. For the 3-bends (ESW4) algorithm [10, 11], the Papakostas and Tollis (PT) algorithm [18] and the Reduce Forks (RF) algorithm [19], we have used the implementations incorporated in the 3DCube system of Patrignani and Vargiu [19]. The Kneser graph $K^a_{b,c}$ consists of the b-subsets of $\{1, 2, \ldots, a\}$ whose intersection has cardinality c.

In most cases the algorithm presented in this paper outperforms the other algorithms both in terms of bounding box volume and the total number of bends.

Table 4. Performance of 3D orthogonal grid drawing algorithms

			Avg. (Max.) bends per edge				Bounding box volume $/n^3$			
graph	n	m	ESW4	PT	RF	DW	ESW4	PT	RF	DW
K_6	6	18	2.60 (3)	2.40 (3)	0.80 (2)	2.00 (2)	20.15	1.56	0.06	0.58
K_7	7	21	2.57 (3)	2.24 (3)	1.71 (6)	2.29 (3)	23.32	1.31	0.73	1.49
$K_{2,2,2,2}$	8	24	2.75 (3)	2.29 (3)	2.25 (8)	2.25 (3)	23.76	1.58	1.96	1.42
$K_{3,3,3}$	9	27	2.74 (3)	2.26 (3)	2.78 (8)	2.22 (3)	24.11	1.51	3.16	1.37
$K_{6,6}$	12	36	2.72 (3)	2.31 (3)	3.67 (9)	2.17 (3)	24.81	1.79	3.75	1.27
$K^5_{2,1}$	10	30	2.67 (3)	2.30 (3)	1.67 (7)	2.27 (4)	24.39	1.72	0.62	1.58
$K^6_{2,0}$	15	45	2.58 (3)	2.31 (3)	2.87 (7)	2.24 (4)	25.24	1.85	2.01	1.62
$K^{11}_{5,0}$	462	1386	-	2.21 (3)	-	2.14 (3)	-	1.77	-	1.47

6 Acknowledgements

The author acknowledges the helpful ideas and suggestions of Graham Farr.

References

[1] H. Alt, M. Godau, and S. Whitesides. Universal 3-dimensional visibility representation for graphs. In F. Brandenburg, editor, *Proc. Graph Drawing: Symp. on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Comput. Sci.*, pages 8–19, Berlin, 1996. Springer.

[2] T. Biedl, T. Shermer, S. Whitesides, and S. Wismath. Orthogonal 3-D graph drawing. In G. DiBattista, editor, *Proc. Graph Drawing : 5th International Symp. (GD'97)*, volume 1353 of *Lecture Notes in Comput. Sci.*, pages 76–86, Berlin, 1998. Springer.

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

[4] T. Calamoneri and A. Massini. On three-dimensional layout of interconnection networks. In G. DiBattista, editor, *Proc. Graph Drawing : 5th International Symp. (GD'97)*, volume 1353 of *Lecture Notes in Comput. Sci.*, pages 64–75, Berlin, 1998. Springer.

[5] M. Chrobak, M. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annual ACM Symp. on Computational Geometry*, pages 319–328, 1996.

[6] R.F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17(2):199–208, 1996.

[7] P. Eades and Q.-W. Feng. Multilevel visualization of clustered graphs. In S. North, editor, *Proc. Graph Drawing : Symp. on Graph Drawing (GD'96)*, volume 1190 of *Lecture Notes in Comput. Sci.*, pages 101–112, Berlin, 1997. Springer.

[8] P. Eades and P. Garvan. Drawing stressed planar graphs in three dimensions. In F. Brandenburg, editor, *Proc. Graph Drawing: Symp. on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Comput. Sci.*, pages 212–223, Berlin, 1996. Springer.

[9] P. Eades, C. Stirk, and S. Whitesides. The techniques of Komolgorov and Bardzin for three dimensional orthogonal graph drawings. *Information Processing Lett.*, 60(2):97–103, 1996.

[10] P. Eades, A. Symvonis, and S. Whitesides. Two algorithms for three dimensional orthogonal graph drawing. In S. North, editor, *Proc. Graph Drawing : Symp. on Graph Drawing (GD'96)*, volume 1190 of *Lecture Notes in Comput. Sci.*, pages 139–154, Berlin, 1997. Springer.

[11] P. Eades, A. Symvonis, and S. Whitesides. Three dimensional orthogonal graph drawing algorithms. 1998, submitted.

[12] S. Fekete, M. Houle, and S. Whitesides. New results on a visibility representation of graphs in 3D. In F. Brandenburg, editor, *Proc. Graph Drawing: Symp. on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Comput. Sci.*, pages 234–241, Berlin, 1996. Springer.

[13] A. Garg, R. Tamassia, and P. Vocca. Drawing with colors. In *Proc. 4th Annual European Symp. on Algorithms (ESA'96)*, volume 1136 of *Lecture Notes in Comput. Sci.*, pages 12–26, Berlin, 1996. Springer.

[14] P.L. Garvan. Drawing and labelling graphs in three-dimensions. In M. Patel, editor, *Proc. 20th Australasian Comput. Sci. Conf. (ACSC'97)*, volume 19 (1) of *Australian Comput. Sci. Comms.*, pages 83–91. Macquarie University, 1997.

[15] H. Koike. An application of three-dimensional visualization to object-oriented programming. In *Proc. Advanced Visual Interfaces (AVI'92)*, volume 36 of *World Scientific Series in Comput. Sci.*, pages 180–192, 1992.

[16] A.N. Kolmogorov and Ya.M. Barzdin. On the realization of nets in 3-dimensional space. *Problems in Cybernetics*, 8:261–268, March 1967.

[17] J. Pach, T. Thiele, and G. Toth. Three-dimensional grid drawings of graphs. In G. DiBattista, editor, *Proc. Graph Drawing : 5th International Symp. (GD'97)*, volume 1353 of *Lecture Notes in Comput. Sci.*, pages 47–51, Berlin, 1998. Springer.

[18] A. Papakostas and I.G. Tollis. Incremental orthogonal graph drawing in three dimensions. In G. DiBattista, editor, *Proc. Graph Drawing : 5th International Symp. (GD'97)*, volume 1353 of *Lecture Notes in Comput. Sci.*, pages 52–63, Berlin, 1998. Springer.

[19] M. Patrignani and F. Vargiu. 3DCube: a tool for three dimensional graph drawing. In G. DiBattista, editor, *Proc. Graph Drawing : 5th International Symp. (GD'97)*, volume 1353 of *Lecture Notes in Comput. Sci.*, pages 284–290, Berlin, 1998. Springer.

[20] F.P. Preparata. Optimal three-dimensional VLSI layouts. *Math. Systems Theory*, 16:1–8, 1983.

[21] S.P. Reiss. 3-D visualization of program information. In R. Tamassia and I. Tollis, editors, *Proc. Graph Drawing : DIMACS International Workshop (GD'94)*, volume 894 of *Lecture Notes in Comput. Sci.*, pages 12–24, Berlin, 1995. Springer.

[22] A.L. Rosenberg. Three-dimensional VLSI: A case study. *J. ACM*, 30(2):397–416, 1983.

[23] D.R. Wood. On higher-dimensional orthogonal graph drawing. In J. Harland, editor, *Proc. Computing : the Australasian Theory Symp.(CATS'97)*, volume 19 (2) of *Australian Comput. Sci. Comms.*, pages 3–8. Macquarie University, 1997.

[24] D.R. Wood. Towards a 2-bends algorithm for three-dimensional orthogonal graph drawing. In V. Estivill-Castro, editor, *Proc. Australasian Workshop on Combinatorial Algorithms (AWOCA'97)*, pages 102–107. Queensland University of Technology, 1997.

[25] D.R. Wood. Two-bend three-dimensional orthogonal grid drawing of maximum degree five graphs. Technical Report 98/03, School of Computer Science and Software Engineering, Monash University, 1998, available at `ftp.csse.monash.edu.au`.