

On the Hardness of Orthogonal-Order Preserving Graph Drawing

Ulrik Brandes and Barbara Pampel*

Department of Computer & Information Science, University of Konstanz
Barbara.Pampel@uni-konstanz.de

Abstract. There are several scenarios in which a given drawing of a graph is to be modified subject to preservation constraints. Examples include shape simplification, sketch-based, and dynamic graph layout. While the orthogonal ordering of vertices is a natural and frequently called for preservation constraint, we show that, unfortunately, it results in severe algorithmic difficulties even for the simplest graphs. More precisely, we show that orthogonal-order preserving rectilinear and uniform edge length drawing is \mathcal{NP} -hard even for paths.

1 Introduction

In several scenarios, a graph drawing algorithm receives as input not only a graph, but also an initial (possibly partial) drawing. The task is to redraw the graph while maintaining selected features of the input drawing. Examples of this kind are embedding-constrained graph layout, shape simplification, sketch-based drawing, and dynamic graph layout.

A cartographic application of particular interest is the simplification of lines. Given a polygonal path, the task is to generate a simpler representation of the path, for instance by omitting vertices [8, 12] (level of detail) or by restricting the allowable types of segments [15, 14] (schematization).

Note that line simplification is also the base case in the design of schematic metro maps, where admissible slopes may be restricted and few bends are desired. Maintaining a user's mental map by preserving the orthogonal ordering [9] of stations and landmarks seems particularly appropriate in this scenario and has been tried, e.g., in [7]. For layout stability [3] and similarity [5] the relative position of vertices, strongly related to the orthogonal ordering, is considered, used [13] and tested [4] helpfull. Alternative constraints include preservation of the cyclic ordering of neighbors [16] and distance from original positions using various metrics [15, 14].

For two different drawing conventions we show that orthogonal ordering is \mathcal{NP} -hard to preserve, even for paths. This is in contrast to the direction-restricted models studied in [15] and [14], where paths or vertices must be within a given distance (according to the Fréchet or Euclidean metric) of the original and the number of bends can be minimized in polynomial time. For orthogonal-order preserving graph drawing, even the decision problems in the rectilinear and equal

* Corresponding author.

edge-length model are \mathcal{NP} -hard. The former implies, e.g., that bend-minimum orthogonal layout is hard under ordering constraints. The latter is also interesting, since drawing with given edge lengths is hard for general graphs [10], but easy for trees (see, e.g., [2]). With orthogonal ordering constraints, the problem is hard even for paths.

After some preliminaries, we treat the rectilinear case in Sect. 3 and the equal edge-length case in Sect. 4. For convenience, we give additional illustrations of gadgets in an appendix.

2 Preliminaries

We are interested in redrawing simple undirected paths $P = (v_1, \dots, v_n)$ using straight line edges. An original geometric position (x_v, y_v) in the plane is given for each vertex $v \in P$. Let (x'_v, y'_v) be the position of a vertex v in the resulting layout. By preserving the orthogonal ordering of the vertices we mean that if for two vertices v_i, v_j it is $x_{v_i} \leq x_{v_j}$ ($y_{v_i} \leq y_{v_j}$) in the original layout, $x'_{v_i} \leq x'_{v_j}$ ($y'_{v_i} \leq y'_{v_j}$) holds also for the resulting layout.

For a (sub)-path P of $l \geq 1$ edges we call the area between the vertical line through P 's rightmost vertex and the one through the leftmost vertex the x -range of P , and analogously the area between the horizontal line through P 's highest vertex and the one through the lowest vertex P 's y -range.

For the \mathcal{NP} -hardness proofs in this paper we use reductions from MONOTONE 3-SAT. In MONOTONE 3-SAT each clause contains exactly three literals either all negated or all non-negated. The problem is known to be \mathcal{NP} -hard [11]. Let I be an instance of the MONOTONE 3-SAT-problem with Boolean variables $X = \{x_1 \dots x_n\}$ and clauses $C = \{C_1, C_2, \dots, C_k\}$.

3 Rectilinear Drawings

The first problem we address is the following:

Orthogonal-order preserving rectilinear drawing problem: Given a graph in the plane, we want to decide whether we can draw each edge either horizontally or vertically, changing neither the horizontal nor the vertical order of endpoints, without introducing any intersection other than the common endpoint of two incident edges and keeping the edge-length positive for each edge.

Choosing the direction of an edge can force the direction of other edges. Figure 1 shows how an edge e_i can force the direction of another edge e_j . More formal: We say an edge e_i *pulls* another edge e_j horizontally, if e_j lies completely within e_i 's y -range, hence, to keep the vertical order of endpoints, e_j has to be horizontal if e_i is horizontal. This of course also means that e_i cannot be drawn horizontally if e_j is vertical and we say e_j *pushes* e_i vertically. Analogously we say an edge e_i *pulls* another edge e_j vertically, if e_j lies within e_i 's x -range

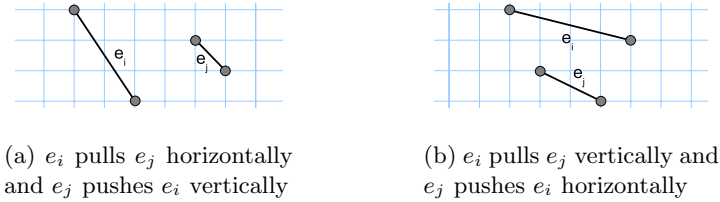


Fig. 1. Forcing to have the same direction

and therefore also e_j pushes e_i horizontally. We use this to construct the main elements of a gadget for the \mathcal{NP} -hardness proof.

Given a path $P = (e_1, e_2, e_3)$ of three edges as shown in Fig. 2. If there is a horizontal edge with one endpoint in the x -range of e_1 and one endpoint in the x -range of e_3 , at least one of P 's edges has to be drawn horizontally and we call P with the horizontal edge a *horizontal decision unit*. Analogously, if there is a vertical edge with one endpoint in the y -range of e_1 and one in the y -range of e_3 at least one of P 's edges has to be drawn vertically and we call P with the vertical edge a *vertical decision unit*. We will later use these decision units to represent the 3-SAT clauses.

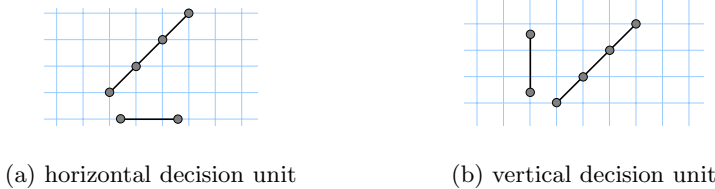


Fig. 2. Decision units

In Fig. 3 two edges $e_i \neq e_j$ are linked by a third edge $l \neq e_i, e_j$. We call l the *horizontal link* for e_i and e_j , if e_j pulls l horizontally and l pushes e_i horizontally such that e_i, e_j and l are all horizontal if e_j is horizontal (see Fig. 3(a)). Of course this also means that if e_i is vertical, also l and e_j must be vertical. A *vertical link* is defined correspondingly and shown in Fig. 3(b). We will use these links for variables which occur in more the one clause.

3.1 Unions of Paths

We now use the described edge-dependency elements to create a gadget for a given instance of MONOTONE 3-SAT to prove the following:

Theorem 1. *The orthogonal-order preserving rectilinear drawing problem is \mathcal{NP} -hard for unions of paths.*

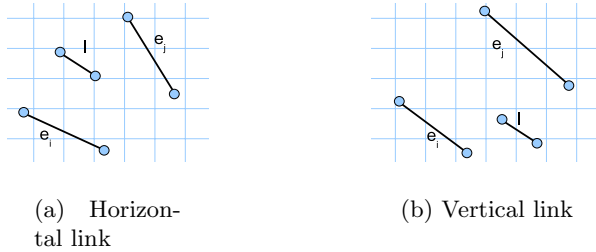


Fig. 3. Possible links l for e_i and e_j

For a given instance I of MONOTONE 3-SAT we create a union of paths as follows. Each variable will have several corresponding edges. For each positive clause C_i we place a horizontal decision unit $U(C_i)$ on the diagonal of the drawing and for each negative clause C_j we place a vertical one $U(C_j)$ as shown in Fig. 4. The horizontal and vertical edges of the decision units can be placed on a horizontal and a vertical line near the borders of the drawing. The diagonal edges in the decision units correspond to the literals in the decision unit’s clause. We then place a *variable path* (e_1, \dots, e_n) (see Fig. 4) on the diagonal of the drawing with n edges corresponding to the n variables in X . For each diagonal edge in a horizontal decision unit we add a positive link between this edge and the edge in the variable path corresponding to the same variable and for each diagonal edge in a vertical decision unit we do the same with a negative link. Because of the links an edge in the variable path is horizontal in a valid orthogonal drawing if an edge corresponding to the same variable is drawn horizontally in a horizontal decision unit and vertical if drawn vertically in a vertical decision unit. We set a variable true if the corresponding edge is drawn horizontally in the variable path and false, if it is drawn vertically, such that for a valid drawing all clauses are satisfied and all other variables can be chosen arbitrary.

Analogously to this, setting the variables such that all clauses are satisfied will also induce a valid drawing, hence the edges in S can be drawn orthogonally without intersections keeping the horizontal and vertical order of their endpoints if and only if I is satisfiable. Thus the problem is proven to be \mathcal{NP} -hard.

The gadget is quite special but we can change it to a gadget with totally ordered vertices, i. e., no two vertices have the same x - or y -coordinates. In the horizontal decision units we used horizontal edges e_h that we can move away from their horizontal line, but force them to be later drawn horizontally again, by attaching at one endpoint a small edge e_f like shown in Fig. 5(a). e_f lies in e_h ’s x -range and e_h in e_f ’s y -range such that the only possibility of avoiding intersections is to draw e_h horizontally and e_f vertically. Since e_h must be drawn horizontally, still at least one of the diagonal edges in the decision unit must be drawn horizontally as well. We place e_f ’s endpoint that is not incident to e_h such that no vertex lies in e_f ’s x -range, so e_f cannot pull any other edge vertically. Because the horizontal edges of the decision units are placed near the borders of the drawing we can easily guarantee that e_f does not lie in any other edge’s

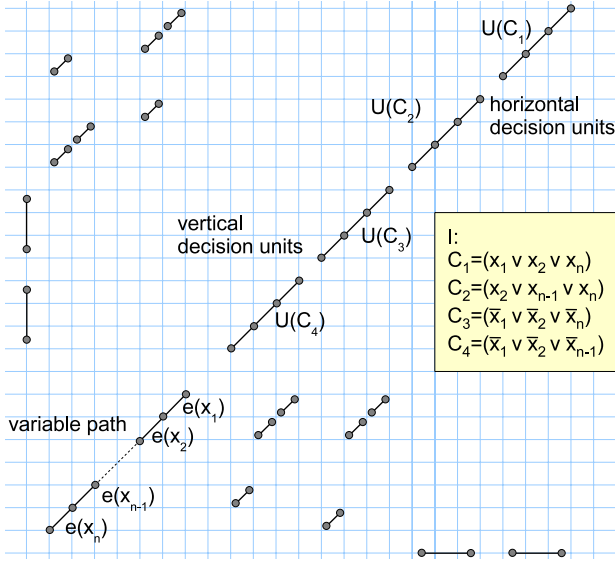


Fig. 4. Union of paths for a MONOTONE 3-SAT-instance I

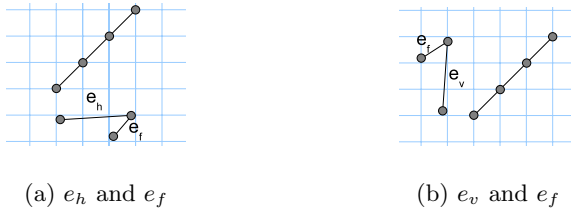


Fig. 5. Decision units without horizontal or vertical edges

y -range, so it cannot push other edges vertically. The strategy for the vertical decision units is the same and shown in Fig. 5(b). After also making sure, that no two vertices of different horizontal (vertical) links for the same edge in the variable path lie on the same vertical (horizontal) line for example by making each horizontal link shorter than the one exactly above it (likewise for vertical links), we have a gadget with total ordering that is drawable if and only if the special gadget was drawable. Thus the problem is proven to be \mathcal{NP} -hard also for unions of paths with totally ordered vertices.

3.2 Single Path

Theorem 2. *Orthogonal-order preserving rectilinear drawing is \mathcal{NP} -hard for paths.*

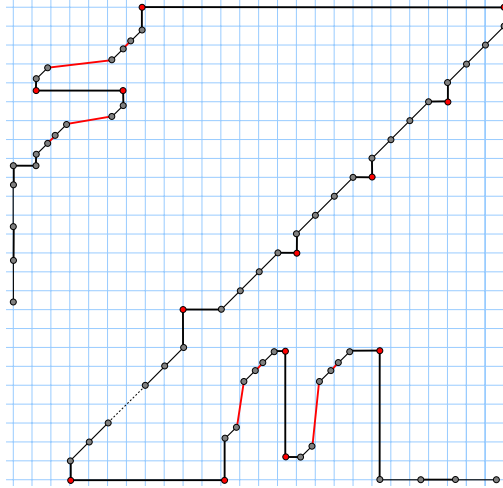


Fig. 6. Single path for the instance I

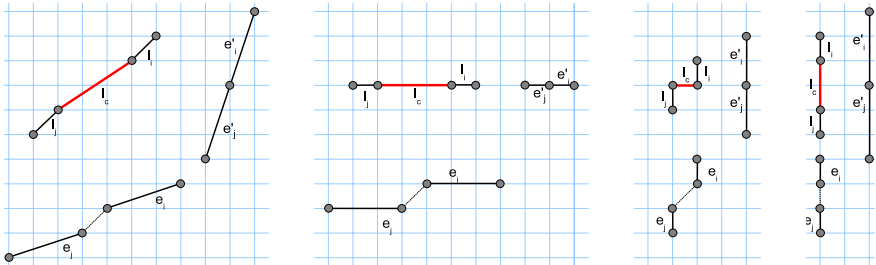


Fig. 7. l_c is connecting two positive links

To show \mathcal{NP} -hardness also for paths we have to connect the edges. We add horizontal and vertical edges connecting the decision units and the links for different decision units (see Fig. 6). They have no effect on the drawability because edges which are already horizontal or vertical cannot pull other edges. Additionally the horizontal and vertical edges added here do not lie in the range of any edge not yet horizontal or vertical and therefore cannot push other edges. Links belonging to the same decision unit are also connected. We take a closer look at these connecting edges. For two incident edges e'_i, e'_j in the same horizontal decision unit and the edges e_i, e_j in the variable path corresponding to the same variables, let $e_{ii'}$ and $e_{jj'}$ be the positive links. (See Fig. 7.) For e_i and e_j not incident let e_c be connecting the positive links. If e_i and e_j not incident let e_c be connecting the positive links. If e'_i and e'_j are both drawn horizontally they pull e_c (as well as $e_{ii'}$ and $e_{jj'}$ of course) horizontally. If e'_i, e'_j or even both edges are vertical, e_c can still be drawn horizontally. Furthermore e_c

can pull all edges $e_{i+1} \dots e_{j-1}$ vertically so we draw it vertically if and only if all edges $e_i \dots e_j$ must be vertical anyway and horizontally otherwise. The edges connecting the links have no effect on drawability. The edges in the union of paths are also kept in the connected path, hence the path is not drawable, if the union of paths had not been drawable.

With the connecting edges we again have vertices with the same x - or y -coordinates, but because they have no effect on drawability we can just slightly turn each horizontal or vertical line with two or more vertices. It can be turned back without conflicts when redrawing the path such that the new connected path is drawable if and only if the old one was. The problem is \mathcal{NP} -hard also for paths with double-totally ordered vertices.

4 Drawings with Uniform Edge Lengths

Orthogonal-order preserving equal edge lengths drawing problem:

Given a graph in the plane, we want to decide whether we can draw each edge with length one changing neither the horizontal nor the vertical order of the edges' endpoints and without introducing any intersection other than the common endpoint of two connected edges.

The constraint that all edges have length one may be exploited to force some of them to be drawn horizontally or vertically. We can use the concept of linking edges like in the previous section. Edge e_i in Fig. 8(a) forces e_j to be drawn horizontally, because otherwise it would not be short enough to have the same length as e_i . We can also define decision units. In the example in Fig. 8(b) the path $P = \{e_1, e_2, e_3\}$ must have length 3 and the only possibility of achieving this is to draw the framing edges horizontally and vertically to give the path the room of a 1×2 -rectangle in which the longest possible path monotone in x - and y -direction has length 3. It is easy to see that one edge of P has to be drawn horizontally and the other two edges vertically.

Let a *horizontal* decision unit be a 3-edge-path monotone in x - and y -direction contained into a 1×2 -rectangle while a *vertical* decision unit is also a 3-edge-path monotone in x - and y -direction, but contained into a 2×1 -rectangle

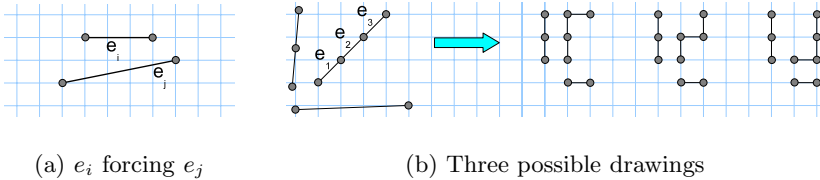


Fig. 8.

4.1 Unions of Paths

We now use these edge-dependency elements to create a gadget for a given instance of MONOTONE 3-SAT to prove the following:

Theorem 3. *The orthogonal-order preserving equal edge lengths drawing problem is \mathcal{NP} -hard for unions of paths.*

For a given instance I of MONOTONE 3-SAT we create a union of paths as follows. Like in the proof for the rectilinear graph drawing we place decision units for the clauses with edges corresponding to the variables in the clause. For each positive clause we place a horizontal decision unit and for each negative clause we place a vertical one. Similar to the gadget in Sect. 3.1 we arrange the decision units on the diagonal such that all vertical and all horizontal decision units lie next to each other without being connected. The framing rectangle edges can be placed on an almost horizontal and an almost vertical line near the borders of the drawing. For each variable in a negative clause that also occurs in a positive clause we add a link between the edge in the vertical and the edge in the horizontal decision unit such that both edges are horizontal if drawn horizontally in the horizontal decision unit and vertical if drawn vertically in the vertical decision unit (see Fig. 9). In a valid drawing there is no edge drawn horizontally in a horizontal decision unit linked to an edge drawn vertically in a vertical decision unit. We choose the variable corresponding to an edge horizontal in a horizontal decision unit true and to an edge vertical in a vertical decision unit false. With this all clauses are satisfied and the other variables can be set arbitrarily.

For a given solution of I we can create a valid drawing as follows: For each positive clause we choose one of the variables set true and draw the corresponding edge in the corresponding decision unit horizontally, the other edges in this decision unit we draw vertically. From each negative clause we also choose one variable set false and draw the corresponding edge in the corresponding decision unit vertically and the other edges horizontally. Now all decision units have a

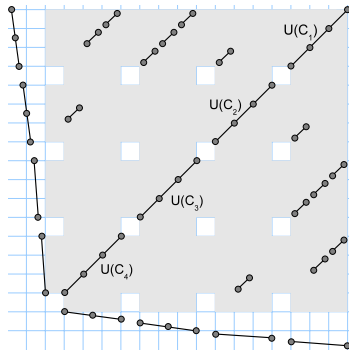


Fig. 9. The union of paths for the instance I from Sect. 3.1

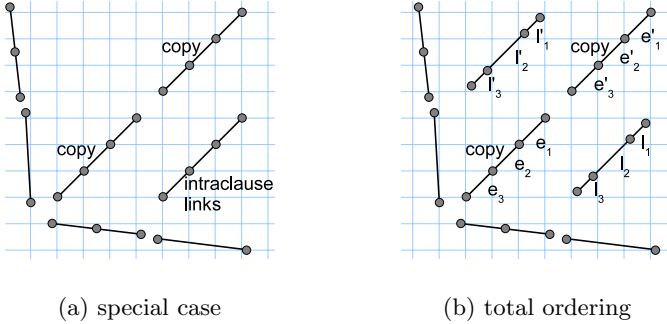


Fig. 10. Intraclause links

valid drawing. The links have a valid drawing, too, because the linked edges are both either horizontal or vertical such that the link can have length one. The union of paths can be drawn with equal edge lengths without intersections keeping the horizontal and vertical order of their endpoints if and only if I is satisfiable. Thus the problem is proven to be \mathcal{NP} -hard. Note that we did not use horizontal or vertical edges and with our arrangement of decision units we can guarantee total ordering by additionally avoiding links on the same horizontal or vertical line like we did in Sect. 3.1.

4.2 Single Path

Theorem 4. *The orthogonal-order preserving equal edge lengths drawing is \mathcal{NP} -hard for paths.*

To show \mathcal{NP} -hardness also for paths we have to add connecting edges and prove that they have no effect on the drawability. We guarantee this by making sure that the connected path stays drawable if the union of paths had been drawable. To make it easier to connect the path segments, we copy the decision units such that each copy has exactly one link to one copy of another decision unit. For $\text{count}_{neg}(e)$ being the number of times the variable corresponding to e occurs in a negative clause and $\text{count}_{pos}(e)$ the number of times it occurs in a positive clause, each vertical decision unit U must have $\sum_{e \in U} \text{count}_{pos}(e)$ copies and each horizontal decision unit U' must have $\sum_{e \in U'} \text{count}_{neg}(e)$ copies. The copies are placed next to each other on the diagonal and linked such that each copy has to be drawn equally (see Fig. 10(a)). We refer to these linking edges as *intraclause links*.

An *interclause link* between a copy of a horizontal and a vertical decision unit lies within a 2x2-rectangle. We first connect the link to two *inner anchor vertices* outside of this rectangle. We take a close look at the case where the third edge of a vertical decision unit is linked with the second edge of a vertical decision unit (see Fig. 11). For the eight combinations of possible drawings of the decision units, except for one combination the anchor vertices lie exactly

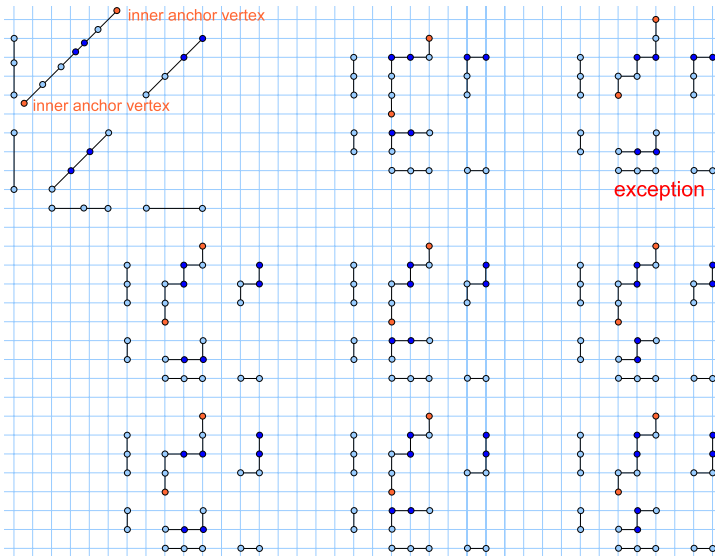


Fig. 11. Eight possible combinations

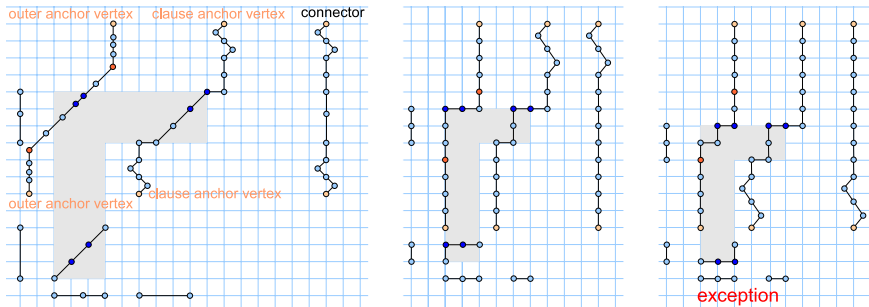


Fig. 12. Outer anchor vertices

at the same distance from the border of the horizontal decision units's y -range and always at the same distance from the border of the vertical decision units's x -range. Because of the one exception, we connect the inner anchor vertices to *outer anchor vertices* that can always be placed each at the same y -coordinates with a *clause anchor vertex* for the horizontal decision unit and at the same x -coordinates with a clause anchor point of the positive decision unit (see Fig. 12). This guarantees, that we can later connect these parts of the path, without having an effect on the drawability.

It is possible to move the horizontal decision unit and possible intraclass links by one edge length. Whenever we have to cross the x - or y -range between the outer anchor vertices when connecting other parts of the path, we can use

a *connector* like shown in Fig. 12 and outside the ranges edges that are either horizontal or vertical with length already 1. Analogously for any interclause link such anchor points can be defined because only one of the decision units has to be movable by at least one edge length. Thus the union of paths can be connected to a path, that stays drawable, if the unions of paths was drawable.

We can also achieve total ordering for a single path. Before connecting the union of paths in the special case, we changed it by copying and linking the clauses. To guarantee that the copies of the decision units for the same clause are all drawn in the same way, we used intraclause links with vertices on the same horizontal and vertical line with the vertices of the decision units (Fig. 10(a)). We now have to link the copies without using vertices with equal x - or y -coordinates. We can do this by using shorter and longer links as shown in Fig. 10(b) on both sides of the units. The shorter links l_1, l'_1 and l_3, l'_3 are pulled by e_1 and e_3 but also push them, such that e_1, l_1, l'_1 and e'_1 must always have the same direction, just as e_3, l_3, l'_3 and e'_3 . The longer links l_2 and l'_2 pull e_2 and e'_2 and are also pushed by them, hence also e_2, l_2, l'_2 and e'_2 all have the same direction and the two copies have to be drawn in the same way. We now have a new union of paths that is drawable if and only if the old one had been drawable as well. We connect the paths exactly like in Sect. 4.2 and turn the horizontal and vertical lines through more than one vertex like in Sect. 3. The connecting edges are always drawable and do not force any other edge, while the edges of the union of paths are still contained, hence the path is drawable if and only if the union of paths had been drawable. The problem is \mathcal{NP} -hard also for a single path with totally ordered vertices along each axis.

References

- [1] Agarwal, P., Har-Peled, S., Mustafa, N., Wang, Y.: Near-linear time approximation algorithms for path simplification. *Algorithmica* 42, 203–219 (2000)
- [2] Bachmaier, C., Brandes, U., Schlieper, B.: Drawing phylogenetic trees. In: Deng, X., Du, D.-Z. (eds.) *ISAAC 2005*. LNCS, vol. 3827, pp. 1110–1121. Springer, Heidelberg (2005)
- [3] Boehringer, K.-F., Newbery Paulisch, F.: Using constraints to achieve stability in automatic graph algorithms. In: *Proc. of the ACM SIGCHI Conference on Human Factors in Computer Systems*, pp. 43–51. WA (1990)
- [4] Bridgeman, S., Tamassia, R.: A user study in similarity measures for graph drawing. *JGAA* 6(3), 225–254 (2002)
- [5] Bridgeman, S., Tamassia, R.: Difference metrics for interactive orthogonal graph drawing algorithms. In: Whitesides, S.H. (ed.) *GD 1998*. LNCS, vol. 1547, pp. 57–71. Springer, Heidelberg (1999)
- [6] Carlson, J., Eppstein, D.: Trees with convex faces and optimal angles. In: Kaufmann, M., Wagner, D. (eds.) *GD 2006*. LNCS, vol. 4372, pp. 77–88. Springer, Heidelberg (2007)
- [7] Dwyer, T., Koren, Y., Marriott, K.: Stress majorization with orthogonal order constraints. In: Healy, P., Nikolov, N.S. (eds.) *GD 2005*. LNCS, vol. 3843, pp. 141–152. Springer, Heidelberg (2006)

- [8] Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canad. Cartog.* 10(2), 112–122 (1973)
- [9] Eades, R., Lai, W., Misue, K., Sugiyama, K.: Layout adjustment and the mental map. *J. Visual Lang. Comput.* 6, 183–210 (1995)
- [10] Eades, P., Wormald, N.: Fixed edge-length graph drawing is \mathcal{NP} -hard. *Discrete Appl. Math.* 28, 111–134 (1990)
- [11] Garey, M., Johnson, D.: *Computers and Intractability*. W. H. Freeman and Company, New York (1979)
- [12] Imai, H., Iri, M.: An optimal algorithm for approximating a piecewise linear function. *J. Inform. Process.* 9(3), 159–162 (1986)
- [13] Lee, Y.-Y., Lin, C.-C., Yen, H.-C.: Mental map preserving graph drawing using simulated annealing. In: *Proc. of the 2006 Asia-Pacific Symposium on Information Visualisation*, pp. 179–188. Australian Computer Science (2006)
- [14] Merrick, D., Gudmundsson, J.: Path simplification for metro map layout. In: Kaufmann, M., Wagner, D. (eds.) *GD 2006*. LNCS, vol. 4372, pp. 258–269. Springer, Heidelberg (2007)
- [15] Neyer, G.: Line simplification with restricted orientations. In: Dehne, F., Gupta, A., Sack, J.-R., Tamassia, R. (eds.) *WADS 1999*. LNCS, vol. 1663, pp. 13–24. Springer, Heidelberg (1999)
- [16] Nöllenburg, M., Wolff, A.: A mixed-integer program for drawing high-quality metro maps. In: Healy, P., Nikolov, N.S. (eds.) *GD 2005*. LNCS, vol. 3843, pp. 321–333. Springer, Heidelberg (2006)