Dynamic Grid Embedding with Few Bends and Changes

Ulrik Brandes and Dorothea Wagner

University of Konstanz
Faculty of Mathematics and Computer Science
D–78457 Konstanz, Germany
{Ulrik.Brandes, Dorothea.Wagner}@uni-konstanz.de

Abstract. In orthogonal graph drawing, edges are represented by sequences of horizontal and vertical straight line segments. For graphs of degree at most four, this can be achieved by embedding the graph in a grid. The number of bends displayed is an important criterion for layout quality. A well-known algorithm of Tamassia efficiently embeds a planar graph with fixed combinatorial embedding and vertex degree at most four in the grid such that the number of bends is minimum [23].
When given a dynamic graph, i.e. a graph that changes over time, one has to take into account not only the static criteria of layout quality, but also the effort users spent to regain familiarity with the layout. Therefore, consecutive layouts should compromize between quality and stability. We here extend Tamassia's layout model to dynamic graphs in a way that allows to specify the relative importance of the number of bends vs. the number of changes between consecutive layouts. We also show that optimal layouts in the dynamic model can be computed efficiently by means that are very similar to the static model, namely by solving a minimum cost flow problem in a suitably defined network.

1 Introduction

One way of reducing perceptual complexity in visualized networks is to represent all edges by alternating sequences of horizontal and vertical straight line segments. Technical requirements, as in VLSI layout, may be another reason for choosing this form of representation. Along with the generally increasing interest in graph drawing (see [11] for a survey), significant attention has been devoted to orthogonal graph layout. Many efficient algorithms have been designed, and many properties of orthogonal layouts (such as the number of bends, the number of crossings, or the area needed) have been analyzed [23,4,2,20,6, and many more].

For planar graphs with vertex degree at most four, Tamassia [23] gave an efficient algorithm computing a grid layout with the minimum number of bends preserving a given combinatorial embedding of the graph. If the embedding is not fixed in advance, bend minimization becomes $\mathcal{NP}$-complete [16]. The algorithm is well-known both because of its conceptual elegance and because the number of bends seems to be an important criterion for layout quality [22]. It is based on the correspondence of flow in an associated network and angles in the embedding. The connection between angles in straight-line representations and values of certain linear programs was investigated by a number of authors [25,18,12,15], but is not one-to-one in general. Tamassia's algorithm has been extended to graphs of higher degree [14], and it builds the core of an algorithm for non-planar graphs of arbitrary degree [24].

In many settings such as user interaction, software visualization, animation of graph algorithms, or graph queries, it is required to deal with dynamic graphs, i.e. graphs that change over time. When a user analyzes a graph visually, he or she builds a mental map that ought not change dramatically when the graph is modified [13]. There are several approaches to dynamic layout [10,7,19], but most research on orthogonal layout has focused on incremental updates, in which only creation of new vertices and edges is allowed [21,5]. InteractiveGiotto [3] is a system allowing arbitrary updates, but the layout of remaining portions of the graph is fixed, which generally increases the number of bends needed. Here, we apply the Bayesian framework of [8] to obtain a dynamic model which forms an explicit and controllable compromise between layout stability and layout quality. Interestingly, the corresponding optimization problem can still be solved efficiently by network flow techniques.

This paper is organized as follows: In Sect. 2, we review the bend minimum grid embedding of [23]. After deriving an objective function for dynamic layout in Sect. 3, we show how to compute a layout accordingly in Sect. 4. Forms of user interaction are discussed briefly in Sect. 5, and examples are given in Sect. 6.

2 Static Bend Minimum Layout

A graph is said to be *4-planar*, if it is planar and has vertex degree at most four. For the orthogonal layout of connected 4-planar graphs with fixed combinatorial embedding, Tamassia [23] generates a flow network and computes a minimum cost flow corresponding to an orthogonal representation with the minimum number of bends preserving the embedding. In this section, we review briefly the concept of orthogonal representation and the construction of the flow network associated to the graph.

Let G be an embedded, connected, 4-planar graph. As shown in Fig. 1, an *orthogonal representation* $H(G)$ of $G = (V, E)$ consists of circular lists $H_f = [\,(e_0, s_0, a_0), \ldots, (e_{d_G(f)-1}, s_{d_G(f)-1}, a_{d_G(f)-1})\,]$ for each face f in G. Each triple (e_i, s_i, a_i) of list H_f consists of an edge $e_i \in E$, a string $s_i \in \{0,1\}^*$, and an integer $a_i \in \{90, 180, 270, 360\}$, such that $e_1, \ldots, e_{d_G(f)-1}$ is a clockwise (counterclockwise, if f is the outer face) traversal of the edges incident to f. Note that edges incident to vertices of degree one appear twice in this traversal. In s_i, 0's and 1's represent 90 and 270 degree bends, respectively, on the right side of e_i in the traversal. Finally, a_i is the size of the angle between e_i and $e_{i+1 \bmod d_G(f)}$.

Lemma 1 ([23]). *If a b-bend orthogonal representation of an n-vertex graph is given, a corresponding grid embedding can be computed in time* $\mathcal{O}(n + b)$.

A flow network $N = (W, A; b, l, u, c)$ consists of a directed graph (W, A) and functions b (*supply/demand*), l (*lower capacity*), u (*upper capacity*), and c (*cost*) defined on the set of arcs. We here assume that all four take integer values and that lower capacities are nonnegative. To avoid confusion with the vertices of the planar graph to be embedded, the elements of W are called *nodes*. A *flow* $x = (x_a)_{a \in A}$ with *cost* $\sum_{a \in A} c(a) \cdot x_a$ is an integer valued vector such that

$$\begin{aligned} b(w) + \sum_{(w',w) \in A} x_{(w',w)} - \sum_{(w,w') \in A} x_{(w,w')} &= 0 \quad \text{for all } w \in W \\ l(a) \leq x_a \leq u(a) & \quad\quad\; \text{for all } a \in A \end{aligned}$$

For the existence of a feasible flow, it is necessary that the total mass balance constraint, $\sum_{w \in W} b(w) = 0$, is satisfied. A vector x satisfying $\sum_{(w',w) \in A} x_{(w',w)} = \sum_{(w,w') \in A} x_{(w,w')}$, $w \in W$, and $l(a) \leq x_a \leq u(a)$, $a \in A$, is called a *circulation*. See [1] for a comprehensive treatment of network flows.

The flow network $N(G)$ used to compute some $H(G)$ is constructed such that each unit of flow corresponds to a 90 degree angle in $H(G)$. Therefore, the node set $W = W_V \cup W_F$ of $N(G)$ consists of all vertices $v \in V$ and faces $f \in F$ of G, i.e. $W_V = V$ and $W_F = F$. Likewise, two types of arcs make up $A = A_V \cup A_F$: For every occurrence of a vertex $v \in V$ in the traversal of a face $f \in F$ there is an arc $(v, f) \in A_V$. For every edge incident to neighboring faces f and g there are arcs $(f, g), (g, f) \in A_F$. Every vertex node $v \in W_V$ has supply $b(v) = 4$, while face nodes $f \in W_F$ have demand

$$b(f) = \begin{cases} -(2 \cdot d_G(f) - 4) & \text{if } f \text{ is an internal face} \\ -(2 \cdot d_G(f) + 4) & \text{if } f \text{ is the external face} \end{cases}$$

By Euler's formula, $\sum_{v \in W_V} b(v) + \sum_{f \in W_F} b(f) = 0$, i.e. the total mass balance constraint is satisfied. The lower and upper capacity of a vertex-face arc $(v, f) \in A_V$ are $l(v, f) = 1$ and $u(v, f) = 4$, respectively, while a face-face arc $(f, g) \in A_F$ has $l(f, g) = 0$ and infinite upper capacity. Arcs in A_V have zero cost, while there is an integer cost $\gamma \geq 0$ on arcs in A_F. See Fig. 1.

Theorem 2 ([23]). *If $\gamma > 0$, there is a 1-1 correspondence between feasible flows in $N(G)$ and orthogonal representations of G (up to the ordering of bends on each edge). Moreover, the total cost of a flow in $N(G)$ corresponding to an orthogonal representation with b bends equals $\gamma \cdot b$.*

3 Dynamic Layout Model

In the remainder of this paper, let G, $G^{(1)} = (V^{(1)}, A^{(1)})$, and $G^{(2)} = (V^{(2)}, A^{(2)})$ be any triple of connected, embedded, 4-planar graphs. $G^{(2)}$ succeeds $G^{(1)}$ in a sequence of graphs and is to be laid out with few bends and changes with respect to a given layout of $G^{(1)}$.

In order to obtain a suitable dynamic layout model we first formulate Tamassia's static model in a slightly different way: Given a connected, embedded, 4-planar graph G and its associated flow network $N(G)$, the range of variables

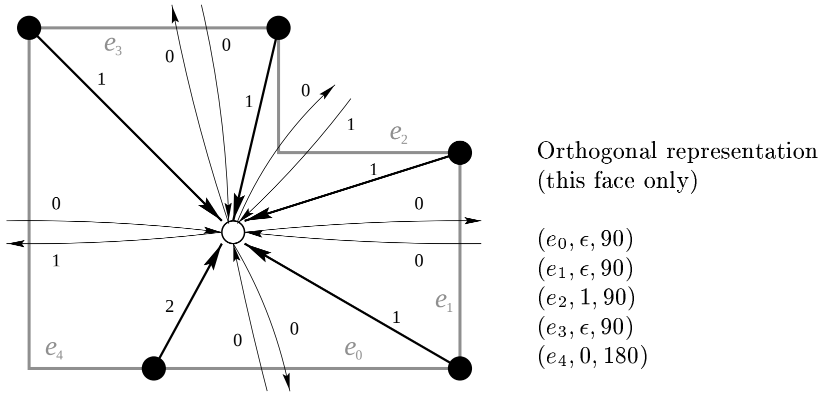


Fig. 1. Part of an embedded 4-planar graph G and its associated flow network $N(G)$. Vertex nodes $v \in W$ have $b(v) = 4$, while the face node $f \in W$ has $b(f) = -(2 \cdot 5 - 4) = -6$. Thick arcs have capacity $[1, 4]$ and cost zero, the others have capacity $[0, \infty)$ and cost γ. All arcs are labeled with values of a feasible flow, and the (*grey*) edges of G are routed according to the corresponding orthogonal representation

x_a, $a \in A$, is $\{l(a), \ldots, u(a)\} = \mathcal{X}_a$. Let $\mathcal{X}$ be the cartesian product of all $\mathcal{X}_a$, $a \in A$, and $\hat{\mathcal{X}} \subseteq \mathcal{X}$ be the set of feasible flow vectors. By Theorem 2, each $x \in \hat{\mathcal{X}}$ represents an orthogonal representation of G. The objective function of the minimum cost flow problem is $U(x) = \sum_{a \in A} c(a) \cdot x_a = \sum_{a \in A_F} \gamma \cdot x_a$, where γ is a nonnegative integer. A random variable X with distribution

$$P(X = x) = \begin{cases} \frac{1}{Z}\, e^{-U(x)} & \text{if } x \in \hat{\mathcal{X}} \\ 0 & \text{otherwise} \end{cases}$$

and $Z = \sum_{x' \in \hat{\mathcal{X}}} \exp\{-U(x')\}$ yields a constrained random field formulation of the layout problem, in which configurations corresponding to a minimum cost flow have the highest probability. The random field framework for layout models was introduced in [9], and it is shown in [8] how dynamic layout models compromizing between readability and stability can be obtained through a Bayesian approach. Therefore, let Y be a random variable for the layout of $G^{(1)}$, and let X be the random variable for its successor $G^{(2)}$. The Bayesian formula gives

$$P(X = x \mid Y = y) = \frac{P(Y = y \mid X = x) \cdot P(X = x)}{P(Y = y)}$$

for all $x \in \hat{\mathcal{X}}$ and $y \in \hat{\mathcal{Y}}$. Since y is given in advance, maximization of the left hand side is equivalent to maximization of the product in the nominator of the right hand side. But the latter consists of a prior distribution $P(X = x)$ corresponding to the static layout model for $G^{(2)}$, and the likelihood of x. It was argued in [8], that this likelihood should be used to express criteria of stability. Since the only layout variables in this model are flow values directly corresponding to features of the orthogonal representation, we choose to penalize deviation in these values.

In other words, we count the number of changes in angles at vertices and bends at edges. Therefore, let $P(Y = y \mid X = x)$ equal

$$= \begin{cases} Z_{Y|X}^{-1} \cdot \exp\left\{ - \sum_{a \in A_V^{(1)} \cap A_V^{(2)}} \alpha \cdot |x_a - y_a| - \sum_{a \in A_F^{(1)} \cap A_F^{(2)}} \beta \cdot |x_a - y_a| \right\} & \text{if } x \in \hat{\mathcal{X}}, y \in \hat{\mathcal{Y}} \\ 0 & \text{otherwise,} \end{cases}$$

where α and β are nonnegative integers, and $Z_{Y|X}$ is the normalizing constant. The dynamic model $P(X = x \mid Y = y)$ thus compromizes between the number of bends in an orthogonal representation (the only criterion of readability), and the number of angles changed at vertices or bends (stability). This compromise can be biased by choosing specific values for parameters α (penalty for changing an angle between consecutive edges of a face), β (penalty for introducing or removing a bend on an edge), and γ (default penalty for bends).

Summarizing the above, a reasonable objective function for layouts x of $G^{(2)}$, given a layout y of $G^{(1)}$, is

$$U(x \mid y) = \sum_{a \in A_F^{(2)}} \gamma \cdot x_a + \sum_{a \in A_V^{(1)} \cap A_V^{(2)}} \alpha \cdot |x_a - y_a| + \sum_{a \in A_F^{(1)} \cap A_F^{(2)}} \beta \cdot |x_a - y_a|.$$

$U(\cdot \mid y)$ is called the *dynamic cost function*. It is only piecewise linear. However, it is shown in the next section how minimization can still be performed by solving a minimum cost flow problem in a suitably defined network.

4 Penalized Residual Networks

Given a flow network $N = (W, A; b, l, u, c)$ and a feasible flow y, the *residual network* with respect to y, $N_y = (W, A \cup \bar{A}; b_y, l_y, u_y, c_y)$, is constructed by adding a *reduction arc* $\bar{a}$ for each $a \in A$, where $\bar{a}$ connects the same nodes as a, but is oriented in the opposite direction. $\bar{A}$ is the set of all reduction arcs. We define

- residual supplies/demands

  $$b_y(w) = b(w) + \sum_{(w',w) \in A} y_{(w',w)} - \sum_{(w,w') \in A} y_{(w,w')}$$

  for all $w \in W$
- residual capacities

  $$\begin{aligned} l_y(a) &= l_y(\bar{a}) = 0 \\ u_y(a) &= u(a) - y_a \\ u_y(\bar{a}) &= y_a - l(a) \end{aligned}$$
- residual costs

  $$\begin{aligned} c_y(a) &= c(a) \\ c_y(\bar{a}) &= -c(a) \end{aligned}$$

Since, for now, y is a feasible flow, $b_y(w) = 0$ for all $w \in W$. Every flow z in N_y yields a circulation $\Delta(z)$ in N by setting

$$\Delta(z)_a \quad = \quad z_a - z_{\bar{a}}$$

Obviously, y can be altered by adding $\Delta(z)$ to give a new feasible flow in N. When a flow should change only if a notable improvement of the cost function is achieved, alteration can be rendered more difficult by adding penalties to residual costs. The resulting network is called the *penalized residual network*, $N'_y = (W, A \cup \bar{A}; b_y, l_y, u_y, c'_y)$, where $c'_y(a) = c_y(a) + \pi_a$ for all $a \in A \cup \bar{A}$. The nonnegative integers π_a represent the extra cost of changing flow along arc a. A flow z in the residual is called *proper*, if for all $a \in A$ at least one of z_a and $z_{\bar{a}}$ is zero. The following theorem summarizes the purpose of this construction.

Theorem 3. *Let y be a feasible flow in some network N. The cost of a proper flow z in N'_y equals $\sum_{a \in A} (c(a) \cdot (x_a - y_a) + \pi_a \cdot |x_a - y_a|)$, where $x = y + \Delta(z)$ is a feasible flow in N.*

For networks $N(G)$, parameters α and β of the dynamic cost function are used to penalize flow alteration on arcs in A_V and A_F, respectively. Figure 2 gives a small example. The following corollary of Theorem 3 shows that stable bend reduction can be carried out by computing a minimum cost flow in the penalized pseudo residual network.

Corollary 4. *Let $G^{(1)} = G^{(2)} = G$, and y be a feasible flow in $N(G)$. The cost of a proper flow z in $N'_y(G)$ equals $U(x \mid y) - U(y)$, where $x = y + \Delta(z)$ is a feasible flow in $N(G)$.*

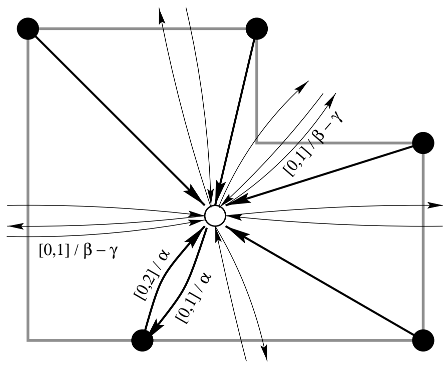


Fig. 2. Penalized residual network with respect to the flow in Fig. 1. Arcs $\bar{a} \in \bar{A}$ with capacities $l(\bar{a}) = 0 = u(\bar{a})$ are not shown. Unlabeled thick arcs have capacity $[0, 3]$ and cost α, unlabeled thin arcs have capacity $[0, \infty)$ and cost β

The flow y in $N(G^{(1)})$ defining the orthogonal representation of $G^{(1)}$ is only a *pseudo-flow* in $N(G^{(2)})$, since its index set no longer equals the set of network arcs. In a pseudo-flow, capacity constraints are obeyed, but flow balance constraints at nodes need not. However, the definition of the penalized residual graph is easily generalized to flow vectors defined on a different index set by setting $y_a = 0$ for all a not in the index set of y. Observe that residual demands/supplies no longer equal zero in general.

Corollary 5. *Given a grid embedding of $G^{(1)}$, a grid embedding of $G^{(2)}$ according to the dynamic layout model can be computed in $\mathcal{O}((n+b)^{3/4} n \sqrt{\log n} + b)$ time, where n is the maximum number of vertices in $G^{(1)}$ and $G^{(2)}$, and b is the maximum number of bends in the orthogonal representations of $G^{(1)}$ and $G^{(2)}$.*

5 Graph Modification

In this section, we sketch one way of maintaining the penalized residual network used to evaluate the dynamic cost function for a modified graph. For convenience, we restrict updates to the *elementary* modifications of inserting and deleting an edge, where the edge may be a *bridge*, i.e. incident to a vertex of degree one. An arbitrary number of elementary modifications in the graph may be traced before a new embedding is computed. The only restriction is that the underlying graph need to be connected, embedded and planar at all times, and 4-planar in the end. Therefore, more powerful update operations can be provided by combining elementary modifications into more complex ones. For example, vertex deletion is a simple sequence of edge deletions that can be carried out in one step. Note that modifications changing the embedding of the graph require more elaborate techniques and conventions for maintaining the network.

Because of space limitations we do not go into details, but rather give an example for the construction of $N'_y(G^{(2)})$ when an edge is inserted between existing vertices of $G^{(1)}$. See Fig. 3 and note that the residual demands/supplies of the two resulting copies of the face node need to be computed by restricting the sum of flow values to those arcs that remain incident to the node. The computational demand of this modification is thus proportional to the size of the face in $G^{(1)}$. The other three elementary modifications can be performed in constant time.

6 Examples

Figure 4(a) shows a bend-minimum grid embedding of a small graph, computed according to Tamassia's model. $G^{(1)}$ is modified elementary by inserting edge $\{7, 2\}$. A bend-minimum grid embedding of the resulting graph is shown in Fig. 4(b). Figures 4(c) and 4(d) show grid embeddings according to our dynamic model with $\alpha = \beta = \gamma = 1$, and with $\alpha = 0$, $\beta = \gamma = 1$, respectively. Changes with respect to the orthogonal representation of $G^{(1)}$ are highlighted in all three drawings of $G^{(2)}$. A vertex is rendered darker, if any angle between consecutive edges incident to it changed, and an edge is drawn thicker, if the sequence of its bends changed.

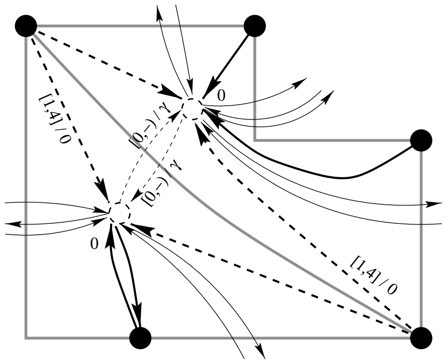


Fig. 3. Insertion of an edge. The modified network has two copies of the face node, two copies of each arc corresponding to a subdivided angle (all with suitably adjusted values), and two new arcs crossing the new edge

It is readily seen that the number of changes in both Fig. 4(c) and 4(d) is much smaller than in Fig. 4(b). On the other hand, the number of bends is not minimum (12 in Fig. 4(c) and 4(d) compared to 10 in Fig. 4(b)). Thus, there is an obvious compromise between the number of bends and the number of changes with respect to the orthogonal representation of Fig. 4(a).

A larger example is given in Fig. 5. Again, a single edge is inserted between two existing vertices, and embeddings obtained from the bend-minimum as well as from the dynamic model are shown. By allowing one additional bend, the dynamic model clearly succeeds in maintaining the overall structure.

Acknowledgment. The authors thank Michael Güdemann for implementing the layout module used to generate the above examples.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows.* Prentice Hall, 1993.

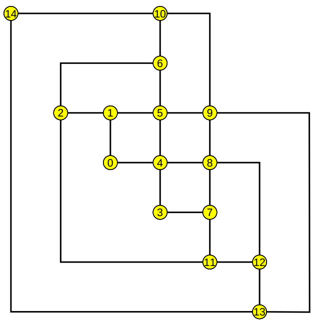
[2] Therese C. Biedl. New lower bounds for orthogonal graph drawings. In *Proc. Graph Drawing '95*, pages 28–39. Springer, LNCS vol. 1027, 1996.

[3] Stina S. Bridgeman, Jody Fanto, Ashim Garg, Roberto Tamassia, and Luca Vismara. InteractiveGiotto: An algorithm for interactive orthogonal graph drawing. In *Proc. Graph Drawing '97*, pages 303–308. Springer, LNCS vol. 1353, 1997.

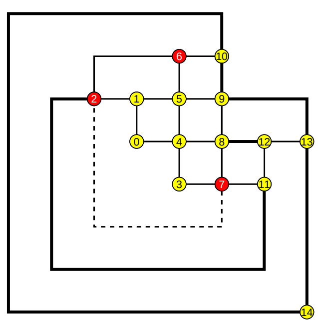
[4] Therese C. Biedl and Goos Kant. A better heuristic for orthogonal graph drawing. In *Proc. 2nd European Symposium on Algorithms*, pages 24–33. Springer, LNCS vol. 855, 1994.

[5] Therese C. Biedl and Michael Kaufmann. Area-efficient static and incremental graph drawings. In *Proc. 5th European Symposium on Algorithms*, pages 37–52. Springer, LNCS vol. 1284, 1997.

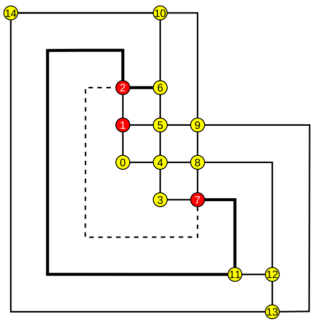
[6] Therese C. Biedl, Brendan P. Madden, and Ioannis G. Tollis. The three-phase method: A unified approach to orthogonal graph drawing. In *Proc. Graph Drawing '97*, pages 391–402. Springer, LNCS vol. 1353, 1997.



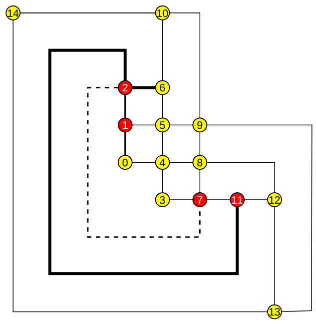
(a) $G^{(1)}$, bend-minimum



(b) $G^{(2)}$, bend-minimum



(c) $G^{(2)}$, dynamic



(d) $G^{(2)}$, dynamic, $\alpha = 0$

Fig. 4. In both cases, the dynamic model produces an embedding with two bends more than necessary, but substantially less changes

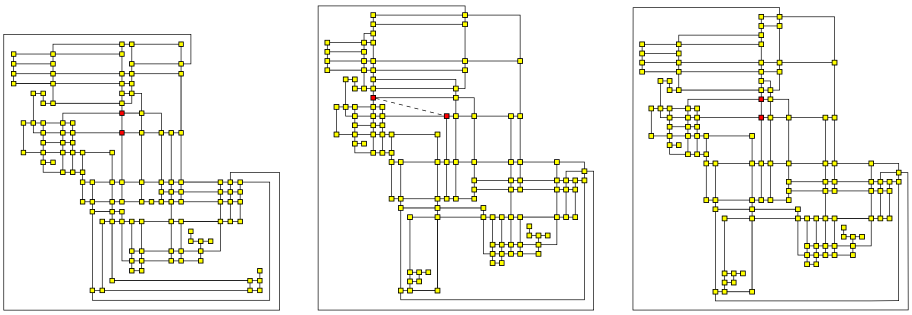


Fig. 5. A larger example with $\alpha = \beta = \gamma = 1$. The layout according to the dynamic model (*right*) displays one more bend than the minimum bend layout (*left*)

[7] Karl-Friedrich Böhringer and Frances Newbery Paulisch. Using constraints to achieve stability in automatic graph layout algorithms. In *CHI'90 Proceedings*, pages 43–51. ACM, The Association for Computing Machinery, 1990.
[8] Ulrik Brandes and Dorothea Wagner. A Bayesian paradigm for dynamic graph layout. In *Proc. Graph Drawing '97*, pages 237–247. Springer, LNCS vol. 1353, 1997.
[9] Ulrik Brandes and Dorothea Wagner. Random field models for graph layout. Konstanzer Schriften in Mathematik und Informatik 33, Universität Konstanz, 1997.
[10] Robert F. Cohen, Giuseppe Di Battista, Roberto Tamassia, Ioannis G. Tollis, and P. Bertolazzi. A framework for dynamic graph drawing. In *Proc. 8th Symposium on Computational Geometry*, pages 261–270. ACM, The Association for Computing Machinery, 1992
[11] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry*, 4:235–282, 1994.
[12] Giuseppe Di Battista and Luca Vismara. Angles of planar triangular graphs. *SIAM J. Discrete Math.*, 9(3):349–359, 1996.
[13] Peter Eades, Wei Lai, Kazuo Misue, and Kozo Sugiyama. Preserving the mental map of a diagram. *Proc. Compugraphics '91*, 9:24–33, 1991.
[14] Ulrich Fößmeier and Michael Kaufmann. Drawing High Degree Graphs with Low Bend Numbers. In *Proc. Graph Drawing '95*, pages 254–266. Springer, LNCS vol. 1027, 1996.
[15] Ashim Garg. On drawing angle graphs. In *Proc. Graph Drawing '94*, pages 84–95. Springer, LNCS vol. 894, 1995.
[16] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. In *Proc. Graph Drawing '94*, pages 286–297. Springer, LNCS vol. 894, 1995.
[17] Ashim Garg and Roberto Tamassia. An new minimum cost flow algorithm with applications to graph drawing. In *Proc. Graph Drawing '96*, pages 201–216. Springer, LNCS vol. 1190, 1997.
[18] Seth Malitz and Achilleas Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Math.*, 7(2):172–183, 1994.
[19] Stephen North. Incremental layout in DynaDAG. In *Proc. Graph Drawing '95*, pages 409–418. Springer, LNCS vol. 1027, 1996.
[20] Achilleas Papakostas and Ioannis G. Tollis. Improved algorithms and bounds for orthogonal drawings. In *Proc. Graph Drawing '94*, pages 40–51. Springer, LNCS vol. 894, 1995.
[21] Achilleas Papakostas and Ioannis G. Tollis. Issues in interactive orthogonal graph drawing. In *Proc. Graph Drawing '95*, pages 419–430. Springer, LNCS vol. 1027, 1996.
[22] Helen Purchase. Which aesthetic has the greatest effect on human understanding? In *Proc. Graph Drawing '97*, pages 248–261. Springer, LNCS vol. 1353, 1997.
[23] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
[24] Roberto Tamassia, Giuseppe DiBattista, and Carlo Batini. Automatic graph drawing and readabilityof diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988.
[25] Gopalakrishnan Vijayan. Geometry of planar graphs with angles. In *Proc. 2nd Symposium on Computational Geometry*, pages 116–124. ACM, The Association of Computing Machinery, 1986.