

Algorithms and Area Bounds for Nonplanar Orthogonal Drawings

Ulrich Fößmeier Michael Kaufmann

Universität Tübingen, Wilhelm-Schickard-Institut, Sand 13, 72076 Tübingen,
Germany,
email: { foessmei / mk } @informatik.uni-tuebingen.de

Abstract. We report on some extensions of the **Kandinsky** model: A new and highly nontrivial technique to incorporate nonplanar drawings into the **Kandinsky** model in the same way as in the **GIOTTO** approach is presented. This means a major step towards the practical usability of our approach. The used technique even gives new insights for the solvability of network flow problems. Another variant of **Kandinsky** ensures a minimal size of the vertices removing the requirement of uniform size of each vertex. We present a new technique to evaluate our approach with respect to the area and the number of bends, and to perform a reasonable comparison with the **GIOTTO** approach.

1 Introduction

1.1 Previous Approaches: **GIOTTO** and **Kandinsky**

In the last decade many different ways were examined how to draw graphs in the plane in order to get a clear and understandable visualization of the data. Further many algorithms were developed which compute drawings in these models (see [4] for an overview). One of the most important ways, maybe even the most appealing one to draw a graph is to produce *orthogonal drawings*. Here the vertices of the graph are placed on grid points of a rectilinear grid and the edges run along gridlines. This approach leads to very good results, especially if the graph is planar. In this case the famous algorithm of Tamassia [14] computes a planar drawing with the minimum number of bends while preserving a given topological embedding of the graph. The basic idea is to compute a related network from the underlying topological embedding, to solve a min-cost flow problem on this network, and to obtain a bend-minimum orthogonal representation of the embedded graph that just includes topological information like sequences of bends, angles, etc. In a final compaction step, the missing geometric data are filled in and exact coordinates for all objects are computed.

This algorithm is restricted to work on graphs with a maximum vertex degree of at most four, so other methods had to be found in order to have drawings for high-degree graphs. **GIOTTO** [15], a further extension of Tamassia's algorithm solves this problem by allowing the vertices to grow in order to have enough place for the incident edges. **GIOTTO** replaces a vertex v of degree d ($d \geq 5$) by

a rectangle R . The horizontal sides of R consist of $\lfloor \frac{d}{4} \rfloor$ vertices and the vertical sides of $\lceil \frac{d}{4} \rceil$ vertices. The d edges incident to v are connected to these new vertices such that the new vertices have degree four or three depending of whether the new vertex is a corner of R or not. Using this strategy the edges are distributed uniformly around v . In the final drawing R is represented by a large rectangle. Unfortunately the size of the vertices cannot be bounded even for graphs having only one vertex of degree five and all others have degree one or two.

An experimental comparison of three different approaches [5] clearly proves the great practical performance of the bend minimizing approach. The algorithm presented in [8] – now called **Kandinsky** – proposes a more flexible way to draw high-degree planar graphs guaranteeing that each vertex has the same size (and this size is not (much) larger than necessary). The model requires the uniform-sized vertices to lie on a (sparse) grid whereas the edges may run between them along the gridlines of a dense grid. One of the most important consequences of these rules is that every vertex v may have at most one straight edge (without bends) leaving v at each side, and among all the edge segments leaving v at the same side, the segment of the straight edge is the longest. Fig. 1 shows an example for a drawing in this model. Again the number of bends is minimized.

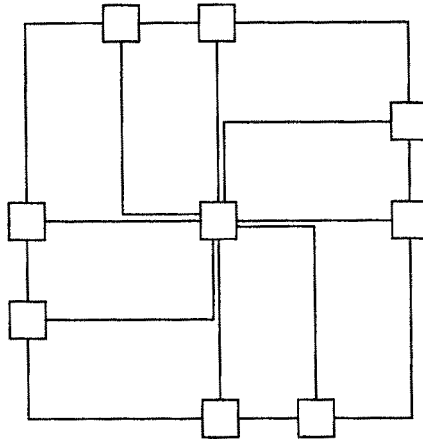


Fig. 1. An example for a Kandinsky-drawing.

All these algorithms are restricted to work on *planar* graphs. This restriction is very unfavourable for most of the practical applications. **GIOTTO** provides a simple but powerful way to handle nonplanar graphs: **GIOTTO** computes for a given graph a corresponding topological embedding and planarizes it by replacing each crossing by an artificial vertex (*crossing vertices*). After the computation of a drawing for this graph these artificial vertices are re-replaced by crossings.

Applying this strategy to **Kandinsky** would lead to drawings like shown in the left drawing of Fig. 2 because each but one of the edges (2, 5), (2, 7), (2, 9) must have a bend before reaching the crossing. The right drawing of Fig. 2 is a better alternative because it produces less bends and avoids the additional crossing

vertices of large size. More precisely: We allow parallel edges to have no bend if we can assure that these edges lie close to each other thus not increasing the size of the vertex. This can only be assured if the face between them has zero area with respect to the sparse grid. The faces defined by the vertex 2 and the crossing vertices in Fig. 2 show an example for this situation.

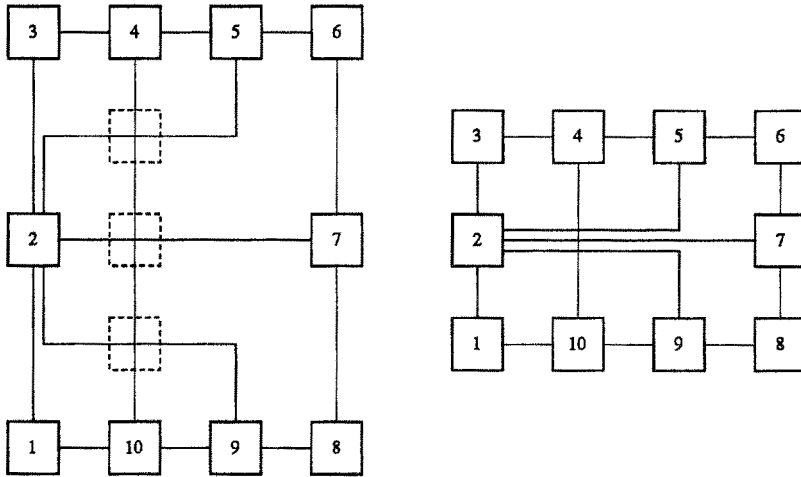


Fig. 2. Kandinsky-drawings without (left) and with (right) modification of the alg.

1.2 This Paper

The new algorithm must distinguish between original vertices and crossing vertices. **Kandinsky** (like Tamassia's algorithm and **GIOTTO**) transforms the graph into a directed network, solves a corresponding min-cost flow problem and transforms the solution back to an orthogonal drawing. We will neglect the compaction step in the following. The network represents relations between every vertex v of the graph and the faces occurring around v (for details see [14] and [8]). This 'local exchange' of informations is sufficient to compute a drawing.

Using **Kandinsky** this situation changes: Consider two edges incident at the same side of some vertex. At least one of them must bend, but not necessarily before the first crossing vertex. Thus the bends in the right drawing of Fig. 2 are related to the vertex 2 without being part of one of the faces occurring around of 2. In Section 2 we will present a new network approach and an alternative technique to solve these problems of 'long-distance'-exchange of informations.

The rest of the paper is devoted to the aspect of the used area. In Section 3 we present a variant of the **Kandinsky** model, where each of the vertices has minimal size. When enhanced by a reasonable compaction, the new algorithm produces drawings that are not only bend-minimal but also of good quality in terms of area. In Section 4 we will derive area bounds for the drawings of the new variant of **Kandinsky** and compare them with corresponding bounds for **GIOTTO**-drawings. Finally, we compare the two approaches using some examples.

2 Nonplanar Orthogonal Drawings

2.1 The New Network

We follow the idea to reduce nonplanar graphs to the planar case by replacing the crossings by crossing vertices. An intuitive view to the networks in all algorithms mentioned so far shows that the faces of the planar graph are nodes in the network and a bend corresponds to a flow unit between two faces having a common edge. The crucial point in the model suggested above is that some arcs in the network are not allowed to have a positive flow *simultaneously*. If e.g. one of the bends in the right drawing of Fig. 2 is related to vertex 2, the edge segments between this bend and vertex 2 are not allowed to have other bends, i.e. the flow related to these segments must be zero if there is a positive flow representing the bend. We call such a set of arcs *forbidden combinations*.

If generally there are k arcs in some network N having the restriction that only one of them may have a positive flow the following construction models this situation: Let $a_1 = (v_1, v'_1), \dots, a_k = (v_k, v'_k)$ be these arcs (each having capacity 1); construct a new network N^* by establishing k additional nodes $\bar{v}_1, \dots, \bar{v}_k$ in the network N and connecting them by a simple cycle $(\bar{v}_1, \bar{v}_2), \dots, (\bar{v}_{k-1}, \bar{v}_k), (\bar{v}_k, \bar{v}_1)$ of directed edges of capacity 1 and cost $-c$ for some constant c . Replace a_i by two arcs (v_i, \bar{v}_i) and (\bar{v}_i, v'_i) both having capacity 1. The cost of (v_i, \bar{v}_i) is set to $c \cdot k + \text{cost}(a_i)$ and the cost of (\bar{v}_i, v'_i) is 0. A flow on an arc $a_i = (v_i, v'_i)$ in N corresponds to a flow on the path $(v_i, \bar{v}_i), (\bar{v}_i, \bar{v}_{i+1}), \dots, (\bar{v}_{i-1}, \bar{v}_i), (\bar{v}_i, v'_i)$ (mod k) in N^* having the same cost (see Fig. 3 for an example). But the realization of a flow on this path uses all edges along the cycle, thus another flow unit on an arc $a_j = (v_j, v'_j)$ in N cannot be represented in N^* in this way because the arcs of the cycle are saturated. Thus the flow on a_j would cause $c \cdot k + \text{cost}(a_j)$ cost in N^* by just using the arcs (v_j, \bar{v}_j) and (\bar{v}_j, v'_j) . If c is chosen large enough, e.g. $c = \sum_{a \in N} \text{cost}(a)$, we can easily prove the following

Lemma 1. *There is a flow x of size r in N and without any forbidden combinations if and only if there is a flow x^* in N^* with $|x^*| = r$ and $\text{cost}(x^*) < c$.*

The size of the new network depends on the number of arcs defining forbidden combinations and can be quadratic in the size of N .

2.2 Solving the New Min-Cost Flow Problem

The common way to solve min-cost flow problems is to use an augmenting algorithm [12] that iteratively augments the flow along shortest paths from s to t . But the existence of negative cycles in N^* makes this strategy unprofitable because solving shortest-path problems in graphs with negative cycles is \mathcal{NP} -hard [10]. A better way is to transform the min-cost flow problem into a linear program (LP) and to solve this LP. This can be done using any polynomial time algorithm for solving such problems. Since all coefficients of the LP (= capacities in N^*) have integer values, the convex polyhedron representing the feasible

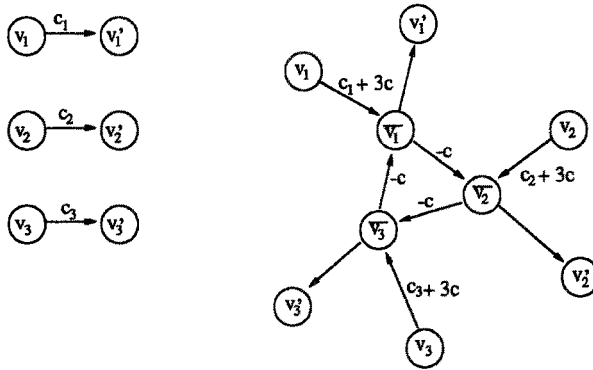


Fig. 3. The construction of N^* if at most one of the arcs (v_1, v_1') , (v_2, v_2') and (v_3, v_3') is allowed to have positive flow.

solutions of the LP only has extreme points with integer coordinates [11]. Thus an integer solution of the LP will be found and we get an integer solution for the flow in N^* which corresponds to a drawing of the graph. Since the objective function of the LP minimizes the cost of flows in N^* , the total number of bends of this drawing is minimized.

For practical implementations we can use the LP solving package CPLEX [6] which leads to an very short running time for this part of the algorithm.

If the way to solve the problem of computing flows with additional constraints of the form $\sum_{i \in I} x_i \leq 1$ leads to a complicated large network which is then transformed into a linear program, an obvious idea is to solve an LP representing the simpler network N to which we add these constraints explicitly. This idea leads to excellent results because N can be *constructed* easier.

2.3 The Drawing Algorithm

We describe the 'simple' network N together with the additional constraints. The transformation either to a network N^* really modeling the drawing problem or to a linear program can easily be done using the techniques presented in the last sections.

Let $G = (V, E)$ be the (nonplanar) embedded graph to be drawn and $G^* = (V \cup V^*, E^*)$ the corresponding planarized version of G , where V^* are the crossing vertices and E^* is the set of edges of G^* as defined above. Let F be the faces of G^* (due to the given drawing); $F'_v \subseteq F$ is the set of faces consisting of three edges e_i, e_j and e_l , two of them (say e_i and e_j) being incident to the same vertex $v \in V$ and the third one (e_l) being incident to two crossing vertices. These faces are called *critical triangles with respect to v* . Define $F' = \bigcup_{v \in V} F'_v$. $F^*_v \subseteq F$ is the set of faces consisting of exactly four edges, all of them being incident to two crossing vertices, *and* having the following property: One of the faces being

neighbourhood to a face $f \in F_v^*$ lies in F'_v or F_v^* . The faces in F_v^* are called *critical quadrangles with respect to v* . Define $F^* = \bigcup_{v \in V} F_v^*$. The set $F_n = F \setminus (F' \cup F^*)$ contains the *normal faces*. Fig. 4 shows an example: f_1 and f_{10} are critical triangles with respect to v ; f_6 is a critical triangle with respect to w ; f_{11} is a critical triangle with respect to u ; f_2 and f_9 are critical quadrangles with respect to v ; f_{12} is a critical quadrangle with respect to u ; f_8 is a critical quadrangle with respect both to v and to u . It is easy to see that only critical faces have the property that a pair of parallel edges uses two gridlines (of the dense grid) lying close to each other (like described in Section 1.1). If we would define e.g. face f_5 in Fig. 4 as critical, too, the edges incident to w could not run parallel without having bends but using neighboured gridlines: The crossing vertex lying opposite to w would need an additional gridline to be placed on between these edges. This gives an intuition for the definition of the critical triangles as the only faces where parallel edges without bends are allowed.

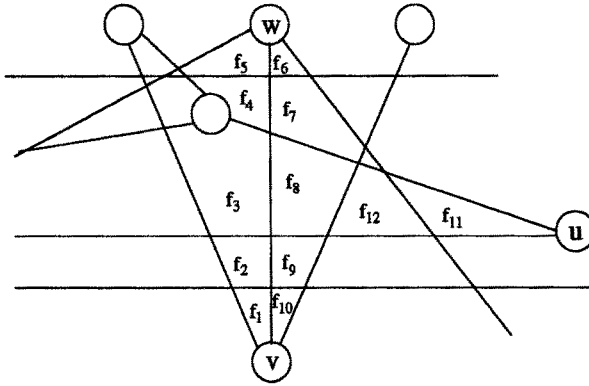


Fig. 4. Critical faces.

The network $N = (U, A, s, t, b, c)$ is defined as follows:

Nodes of N : $U = U_V \cup U_F \cup \{s, t\}$ where $U_V = \{u_v : v \in V\}$ and $U_F = \{u_f : f \in F\}$.

Arcs of N : $A = A_{SV} \cup A_{SF} \cup A_{VT} \cup A_{FT} \cup A_{VF} \cup A_{FFV} \cup A_{FF}$ where

A_{SV} contains arcs from s to $u_v \in U_V$ if $\text{degree}(v) \leq 3$; the arcs have cost 0 and capacity $4 - \text{degree}(v)$.

A_{SF} contains arcs from s to $u_f \in U_F$ if f is not the external face of G^* and if f has at most three edges; the arcs have cost 0 and capacity $4 - (\# \text{ edges of } f)$.

A_{VT} contains arcs from $u_v \in U_V$ to t if $\text{degree}(v) \geq 5$; the arcs have cost 0 and capacity $\text{degree}(v) - 4$.

A_{FT} contains arcs from $u_f \in U_F$ to t if f is not the external face of G^* and if f consists of at least five edges; the arcs have cost 0 and capacity $(\# \text{ edges of } f) - 4$. Moreover there is an arc from u_f to t in A_{FT} if f is the external face of G^* ; this arc has cost 0 and capacity $(\# \text{ edges of } f) + 4$.

A_{VF} contains arcs from $u_v \in U_V$ to $u_f \in U_F$ if v lies at the border of f ; the arcs have cost 0 and capacity 3.

A_{FFV} contains arcs $(u_f, u_v)_e$ from $u_f \in U_F$ to $u_v \in U_V$, namely one for every edge segment e if either e lies at the border of f and is incident to v , or g is a critical quadrangle with respect to v , and f and g have e as a common edge; these arcs have cost 1 and capacity 1.

A_{FF} contains arcs from $u_f \in U_F$ to $u_g \in U_F$ if f and g have at least one common edge; these arcs have cost 1 and capacity ∞ .

Let $x : A \rightarrow \mathbb{R}^+$ be a feasible integer flow in N with maximum $|x|$; a drawing can be constructed as follows: the angle at a vertex v between two incident edges having the face f between them is set to $(x((u_v, u_f)) - x((u_f, u_v)) + 1) \cdot 90^\circ$. Each edge e has $|x((u_f, u_g))|$ bends if f and g have e as common edge.

Before these angles and bends are computed the arcs in A_{FFV} have to be transformed: An arc $(u_f, u_v)_e \in A_{FFV}$ stands for a combination of an arc $(u_f, u_g) \in A_{FF}$ and an arc $(u_g, u_v) \in A_{FV}$ if f and g have e as common edge. In this way it is guaranteed that a flow from a face into a vertex (which corresponds to an angle of 0°) can only occur together with a bend. This situation is described more detailed in [8]. The crossing vertices are not represented in the network, so it is guaranteed that all angles around these vertices are equal to 90° and thus these vertices can be replaced by crossings in the final drawing step.

The cost of x is equal to the number of bends in the drawing, thus a flow with minimum cost produces a drawing with the minimum number of bends.

The network described so far cannot simulate the drawing problem: Obviously several combinations are not allowed to occur in the drawing. The first two conditions appeared similarly in previous papers on Kandinsky [8,9], while the third condition reflexed the difficulties concerning nonplanarity.

- It is not possible to have negative angles; so the sum of the flows from a face into a vertex (after replacing the arcs in A_{FFV}) has to be at most one.
- Each bend belonging to a 0° -angle at a vertex v must lie close to v (i.e. between the bend and v there are no other bends); thus an edge cannot bend in different directions corresponding to 0° -angles at the same vertex.
- Further if b is a bend on an edge belonging to a critical quadrangle with respect to v and b corresponds to a 0° -angle at v then all critical faces between b and v have to be drawn without bends (see the right drawing in Fig. 2 for an example: the faces defined by vertex 2 and the crossing vertices do not have any bends at their edge segments).

These conditions can be guaranteed by applying one of the techniques described in the previous section. There is one weak point in this concept that does not allow to have an unbounded number of normal bends on a single edge. Details can be found in the full paper. We summarize the results of this section in the following

Theorem 1. *For a given drawing of a graph $G = (V, E)$ an embedding preserving drawing in the Kandinsky-model with the minimum number of bends can be computed in polynomial time under the restriction that each edge of a critical triangle or of a critical quadrangle has at most k bends (for a constant k).*

3 Vertices of Minimal Size

The requirement of vertices of uniform size sometimes seems artificial. Often it is more important to represent the vertices just as big that the incident edges can be drawn separately. In the full version, we show how to modify the **Kandinsky** model such that the vertices may have different size and a detailed analysis of the area can be done.

In original **Kandinsky** drawings, the vertices of uniform squarish size sit on the intersections of a sparse uniform grid. In between the sparse grid lines there is a finer grid supporting the routing of the edges [8]. Now we resolve the two grids such that only one grid remains. The spacing between neighbored parallel grid lines is set to be the minimal distance of two parallel edges.

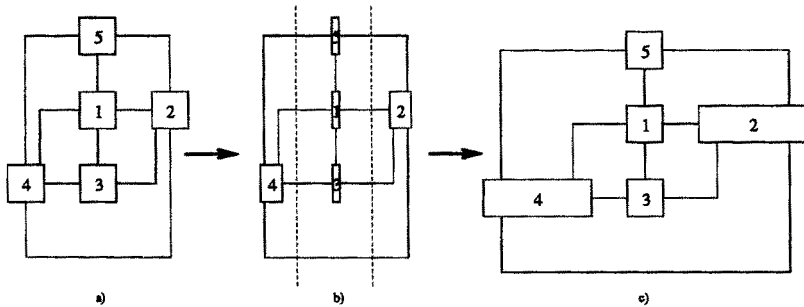


Fig. 5. Minimizing the size of the vertices: 3 snapshots of the algorithm.

4 Bounds for the Area of Orthogonal Drawings

In this section we give upper bounds for the area used by drawings produced by **GIOTTO** and **Kandinsky**. We consider the **Kandinsky** only with shrunken vertices, since otherwise a fair comparison is not possible because of different underlying grids. Recently two area bound have been given for general graphs [13] and [3] ($\frac{3}{2}m$ and $m + n$ gridlines respectively) but the approaches are completely different, and the algorithms produce highly nonplanar drawings with relatively many bends. Note that our bounds are not restricted to planar drawings.

4.1 Graphs of Small Degree

For the **GIOTTO** model, the following easy observation [2] helps us to develop an upper bound for the area. Later, we will extend it to the **Kandinsky** model.

Assume that each vertex has degree at most 4. If each edge has exactly one bend, each gridline in horizontal and vertical direction carries at most one vertex in the worst case. The number of gridlines is $2n$ in this case. If one edge has no bends, the corresponding vertices share a gridline, if an edge has b bends, there are $b - 1$ segments not incident to any vertex. Each of the segments might need a separate gridline. The number of gridlines is linear in the number of bends.

Hence the total number of gridlines is at most $2n - m + k_G$, where k_G is the number of bends produced by GIOTTO. Note that the argument does not use planarity but only that the degree is at most 4 and the edges extend to all sides.

In **Kandinsky** drawings (even for graphs with a vertex degree of at most four) there may arise larger vertices. The vertices use at most $2 + \#(0^\circ\text{-angles})$ gridlines because every 0° -angle may force the vertex to grow by one unit. This gives a bound of $2n - m + k_K + \#(0^\circ\text{-angles})$ where k_K denotes the number of bends.

In order to compare this formula with the bound for GIOTTO we analyse the number of 0° -angles. A 0° -angle always arises together with a bend; thus $\#(0^\circ\text{-angles}) \leq k_K$. Clearly, $k_K \leq k_G$, and if there are 0° -angles then $k_K < k_G$. We can show

$$k_G \leq k_K + \#(0^\circ\text{-angles}) \leq 2 \cdot k_G$$

, since GIOTTO can simulate a 0° -angle by two 'normal' bends and the second inequality follows from the observations above. Usually $k_G \approx k_K + \#(0^\circ\text{-angles})$ and both algorithms produce drawings of the same size, but there are lower bound examples where the two terms are a factor of two off.

As a consequence the upper bound for the area of GIOTTO drawings is never larger than the upper bound for **Kandinsky** drawings.

4.2 Graphs of Higher Degree

To apply the technique of the previous section to graphs with an arbitrary vertex degree we have to formulate it in a more general way. A vertex v is said to have a *free pin* at every position where an additional edge could be connected to v . 'freepins(v)' denotes the number of free pins of vertex v and 'freepins' = \sum_v freepins(v). Thus a vertex v uses $\frac{\deg(v) + \text{freepins}(v)}{2}$ gridlines. Therefore the total area can be bounded by $m + \frac{\text{freepins}}{2}$ if every edge has exactly one bend. With the argument of the previous section an upper bound for the area in the general case is $m + \frac{\text{freepins}}{2} + k - m = k + \frac{\text{freepins}}{2}$ where k is the number of bends.

In **Kandinsky** drawings we have to take into account that the edges being incident to some vertex v might extend all towards the same direction. In this case $\text{freepins}(v) = \deg(v) + 2$ and the total area can only be bounded by $k_K + \frac{1}{2} \sum_v (\deg(v) + 2) = k_K + m + n$. But we can do better: Using some properties for bend minimum drawings excludes most of the vertices from the sum:

Theorem 2. *A drawing in the Kandinsky model with different vertex sizes uses at most $m + k_K + |\{v : \deg(v) = 1\}|$ gridlines, k_K being the number of bends.*

Note that we did not take into account that there may be many pairs of edges incident to opposite sides of the same vertex and using the same gridline. In all examples in the rest of this paper there is not a single vertex having the bad property that all its incident edges run into the same direction.

Analyzing GIOTTO is easier because the edges are forced to be uniformly distributed to the vertex sides. Similarly as above, we can show:

Theorem 3. *A drawing in the GIOTTO model uses at most $k_G + \frac{1}{2}(|\{v : \deg(v) \text{ is odd}\}| + |\{v : \deg(v) \leq 2\}|)$ gridlines, k_G being the number of bends.*

Orthogonal drawings usually need smaller area than due to the formulas given in this section. The formulas induce that all small vertices of the auxiliary representation for a high degree vertex that lie on the same gridline are connected by straight edges. If a gridline is used by r such connected sequences of vertices the gridline is counted r times. To count this effect we replace the large vertices by the auxiliary construction for GIOTTO (a rectangle of small vertices) and look for gridlines that are used by more than one connected sequence (a connected sequence also can consist of zero vertices if there is an edge segment connecting two bends). We can show

Theorem 4. *An orthogonal drawing uses exactly $(k + 1/2(\text{freepins}) - \text{savedlines})$ gridlines where k is the number of bends and savedlines is the number of multiple use of gridlines as observed above.*

An example for this analysis is given in the next section.

5 Examples

Our first example shows the graph G_3 of the graph drawing contest of GD'94 and we took the embedding found by Petra Mutzel for her winning drawing. The results are shown in Fig. 6. GIOTTO changed the embedding. Observe the marked edge. It has 10 crossings and 4 bends. A simple rerouting of this edge would reduce the size by 2 gridlines, save 6 crossings and 2 bends. In total we have size 23×20 , 32 bends and 27 crossings. Some vertices are much larger than necessary, e.g. 13, 11, 2.

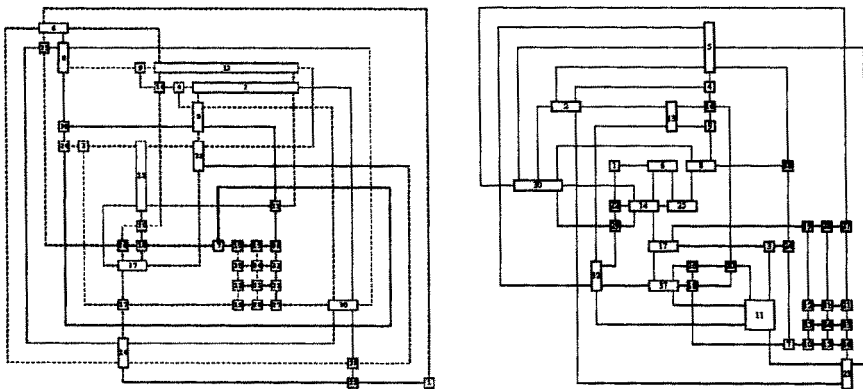


Fig. 6. The resulting drawings by GIOTTO (left) and by Kandinsky (right) for G_3 .

For the **Kandinsky** drawing, the situation is better: We obtain size 21×20 , 29 bends and 20 crossings. Note, that we have maintained the original embedding given in [7].

We derive the used area of the drawings using the formula from Theorem 4: In the **GIOTTO** drawing of Fig. 6 two gridlines are multiple used: The 8th leftmost column and the 7th bollowmmost row are doubly used. Thus the drawing should save two gridlines compared to the formula. Indeed, the drawing uses $23+20 = 43$ gridlines, while the formula bounds the area by 32 (bends) $+ 12$ (24 vertices have odd degree) $+ 1$ (one vertex with degree two) $= 45$.

In the **Kandinsky** drawing of Fig. 6 we can save 10 gridlines compared to the formula by multiple use of gridlines. The number of free pins is 44, so the formula states an area of 29 (bends) $+ 22$ ($1/2 \cdot$ (freepins)) $- 10$ (savedlines) $= 41$. This is conform to the needed area of (21×20) .

We leave the question open for future research if the better re-use of gridlines is a provable quality of the **Kandinsky** model.

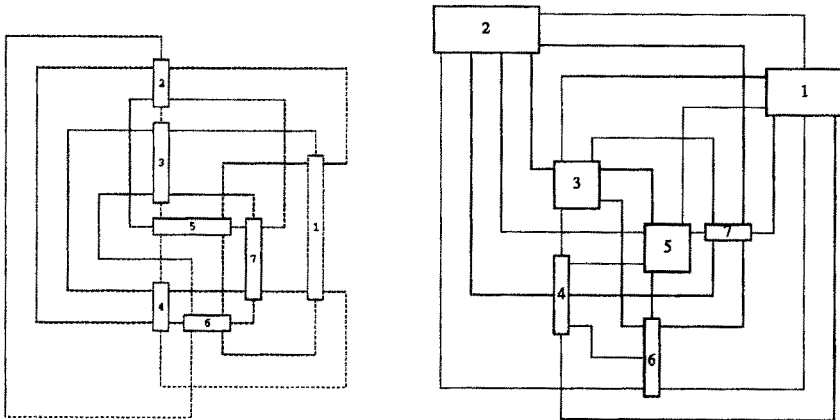


Fig. 7. The K_7 : The resulting drawing by **GIOTTO**, and the **Kandinsky** drawing.

As a second example we compared the behaviour of the algorithms applied to the complete graph K_7 (see Fig. 7). For the **GIOTTO** drawing, we obtain size 12×13 , 26 bends and 9 crossings. The main drawback here is that the vertices are (often) stretched much. The **Kandinsky** drawing has size 14×14 , 21 bends and 9 crossings. We have taken the same embedding as produced by **GIOTTO**. So, the number of crossings is the same. This example is specially suitable for **GIOTTO** since freepins is zero (because every vertex has even degree) and the graph is extremely dense; this fact makes it more difficult to re-use gridlines such that **Kandinsky** can only subtract two gridlines because of multiple use, but it suffers from the large number of 18 free pins.

More examples can be found in the full version of the paper.

6 Conclusion

Summarizing the results of the comparison done in this paper we state that

- **GIOTTO**
 - + gives good theoretical bounds for the used area
 - + performs well in practice w.r.t. the number of bends and to the used area
 - does not define a model of orthogonal drawings for high-degree graphs (it is not clear under which conditions a vertex may be large)
 - area/bends/size of vertices depend on the chosen embedding [1] but also on the distribution of the edges around the vertices
- **Kandinsky**
 - + defines a model for orthogonal drawings
 - + minimizes a cost function (number of bends) in this model
 - + behaves well in practice w.r.t. the area (multiple use of gridlines)
 - needs an embedding as part of the input
 - does not allow an easy analysis of the used area.

References

1. Bertolazzi, P., G. Di Battista, W. Didimo, *Computing Orthogonal Drawings with the Minimum Number of Bends*, Proc. of WADS'97, to appear.
2. Biedl, T.C., *Orthogonal Graph Drawing, Algorithms and Lower Bounds*, Diplomarbeit TU Berlin, 1995.
3. Biedl, T.C., M. Kaufmann, *Area-Efficient Static and Incremental Drawings of High-Degree Graphs*, to appear in ESA 1997.
4. Di Battista, G., P. Eades, R. Tamassia, I.G.Tollis, *Algorithms for Drawing Graphs: An Annotated Bibliography*, Computational Geometry: Theory & Applications, 235-282, 1994.
5. Di Battista, G., A. Garg, G. Liotta, R. Tamassia, E. Tassinari, F. Vargiu, *An Experimental Comparison of Three Graph Drawing Algorithms*, Computational Geometry: Theory & Applications, 1996.
6. CPLEX optimization, Inc., *Using the CPLEX Base System*.
7. Eades, P., J. Marks, *Graph-Drawing Contest Report*, Proceedings on GD'94, Princeton, LNCS 894, 143-146, 1995.
8. Föfmeier U., M. Kaufmann, *Drawing High Degree Graphs with Low Bend Numbers*, Proceedings on GD'95, Passau, LNCS 1027, 254-266, 1995.
9. Föfmeier U., G. Kant, M. Kaufmann, *2-Visibility Drawings of Planar Graphs*, Proceedings on GD'96, Berkeley, LNCS 1190, 155-168, 1996.
10. Garey, M.R., D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.
11. Kleinschmidt, P., *personal communication*.
12. Lawler, E.L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976, Chapter 4.
13. Papakostas A., I.G. Tollis, *High-degree orthogonal drawings with small grid-size and few bends*. Technical report, University of Texas at Dallas, Richardson, TX 75083, December 1996.
14. Tamassia, R. *On Embedding a Graph in the Grid with the Minimum Number of Bends*, SIAM Journal of Computing, vol. 16, No. 3, 421-444, 1987.
15. Tamassia, R., G. Di Battista, C. Batini, *Automatic Graph Drawing and Readability of Diagrams*, IEEE Trans. Systems, Man and Cybernetics, 61 - 79, 1988.