

# Complexity of Higher-Degree Orthogonal Graph Embedding in the Kandinsky Model<sup>\*</sup>

Thomas Bläsius<sup>1</sup>, Guido Brückner<sup>1</sup>, and Ignaz Rutter<sup>1,2</sup>

<sup>1</sup> Faculty of Informatics, Karlsruhe Institute of Technology, Karlsruhe

<sup>2</sup> Department of Applied Mathematics, Charles University, Prague

**Abstract.** We show that finding orthogonal grid embeddings of *plane graphs* (planar with fixed combinatorial embedding) with the minimum number of bends in the so-called Kandinsky model (allowing vertices of degree  $> 4$ ) is NP-complete, thus solving a long-standing open problem. On the positive side, we give an efficient algorithm for several restricted variants, such as graphs of bounded branch width and a subexponential exact algorithm for general plane graphs.

## 1 Introduction

Orthogonal grid embeddings are a fundamental topic in computer science and the problem of finding suitable grid embeddings of planar graphs is a subproblem in many applications, such as graph visualization [19] and VLSI design [17,21]. Aside from the area requirement, the typical optimization goal is to minimize the number of bends on the edges (which heuristically minimizes the area). Traditionally, grid embeddings have been studied for *4-planar graph* (max-deg 4), which is natural since it allows to represent vertices by grid points and edges by internally disjoint chains of horizontal and vertical segments on the grid. For plane graphs, Tamassia showed that the number of bends can be efficiently minimized [14]; the running time was recently reduced to  $O(n^{1.5})$  [7]. In contrast, if the combinatorial embedding is not fixed, it is NP-complete to decide whether a 0-embedding (a *k-embedding* is a planar grid embedding with at most  $k$  bends per edge) exists [14], thus also showing that bend minimization is NP-complete and hard to approximate. In contrast, a 2-embedding exists for every graph except the octahedron [2]. Recently it was shown that the existence of a 1-embedding can be tested efficiently [4]. The problem is FPT if some subset of  $k$  edges has to have 0 bends [5]. If there are no 0-bend edges, it is even possible to minimize the number of bends in the embedding, not counting the first bend on each edge [6].

These results only apply to graphs of maximum degree 4. There have been several suggestions for possible generalizations to allow vertices of higher degree [16,20]. For example, it is possible to represent higher-degree vertices by rectangles. The disadvantage is that the vertices may be stretched arbitrarily in order to avoid bends. In particular, a visibility representation (existing for every planar graph) can be interpreted as a 0-embedding in this model. It is thus natural to forbid stretching of vertices.

---

<sup>\*</sup> Partially supported by grant WA 654/21-1 of the German Research Foundation (DFG). Partly done within GRADR – EUROGIGA project no. 10-EuroGIGA-OP-003. Ignaz Rutter was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

Fößmeier and Kaufmann [13] proposed a generalization of planar orthogonal grid embeddings, the so-called Kandinsky model (originally called *podevsnef*), that overcomes this problem and guarantees that vertices are represented by boxes of uniform size. Essentially their model allows to map vertices to grid points on a coarse grid, while routing the edges on a much finer grid. The vertices are then interpreted as boxes on the finer grid, thus allowing several edges to emanate from the same side of a vertex; see Sect. 2. Fößmeier and Kaufmann model the bend minimization in the fixed combinatorial embedding setting by a flow network similar to the work of Tamassia [18] but with additional constraints that limit the total amount of flow on some pairs of edges. Fößmeier et al. [12] show that every planar graph admits a 1-embedding in this model. Concerning bend minimization, reductions of the mentioned flow networks to ordinary minimum cost flows have been claimed both for general bend minimization [13] and for bend minimization when every edge may have at most one bend [12].

Eiglsperger [10] pointed out that the reductions to minimum cost flow are flawed and gave an efficient 2-approximation. Bertolazzi et al. [1] introduced a restricted variant of the Kandinsky model (requiring more bends), for which bend minimization can be done in polynomial time. Although the Kandinsky model has been later vastly generalized, e.g., to apply to the layout of UML class diagrams [11], the fundamental question about the complexity of bend minimization in the Kandinsky model has remained open for almost two decades.

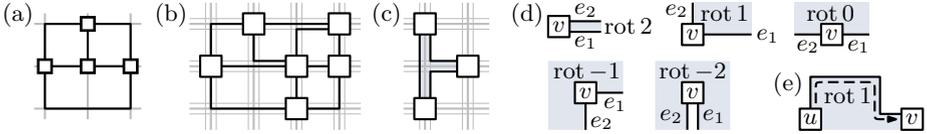
*Contribution and Outline.* We show that the bend minimization problem in the Kandinsky model is NP-complete (no matter if we allow or forbid so-called empty faces). This also holds if each edge may have at most one bend; see Sect. 3. As an intermediate step, we show NP-hardness of the problem ORTHOGONAL 01-EMBEDDABILITY, which asks whether a plane graph (with maximum degree 4) admits a grid embedding when requiring some edges to have exactly one and the remaining edges to have zero bends. This result is interesting on its own, as it can serve as tool to show hardness of other grid embedding problems. In particular, it gives a simpler proof for the hardness of deciding 0-embeddability (maximum degree 4) for graphs with a variable embedding.

We then study the complexity of the problem subject to structural graph parameters in Sect. 4. For graphs with branch width  $k$ , we obtain an algorithm with running time  $2^{O(k \log n)}$ . For fixed branch width this yields a polynomial-time algorithm ( $O(n^3)$  for series-parallel graphs), for general plane graphs the result is an exact algorithm with subexponential running time  $2^{O(\sqrt{n} \log n)}$ .

For detailed proofs, we refer to the full version of this paper [3].

## 2 Preliminaries

**Kandinsky Embedding.** Let  $G$  be a plane graph. An *orthogonal embedding* of  $G$  maps vertices to grid points and edges to paths in the grid such that the resulting drawing is planar and respects the combinatorial embedding of  $G$ ; see Fig. 1a. Clearly,  $G$  admits an orthogonal embedding if and only if it is 4-planar. The Kandinsky model [13] overcomes this limitation. A *Kandinsky embedding* of  $G$  (Fig. 1b) maps each vertex to a box of constant size centered at a grid point and each edge to a path in a finer grid such that



**Fig. 1.** (a) An orthogonal embedding of the  $K_4$ . (b) A Kandinsky embedding of the wheel of size 5. (c) A Kandinsky embedding with an empty face. (d–e) The rotation of a vertex (d) and an edge (e) in a face  $f$  (shaded blue).

the resulting drawing is planar, respects the combinatorial embedding of  $G$ , and has no empty faces. A face is empty if it does not include a grid cell of the coarser grid; see Fig. 1c.

One can declare a bend on an edge  $uv$  to be *close* to  $v$  if it is the last bend on  $uv$  (traversing  $uv$  from  $u$  to  $v$ ). A bend cannot simultaneously be close to  $u$  and to  $v$ . Kandinsky embeddings have the *bend-or-end property* [13], requiring that a  $0^\circ$  angle between edges  $uv$  and  $vw$  in the face  $f$  implies that at least one of the edges  $uv$  and  $vw$  has a bend close to  $v$  forming a  $270^\circ$  angle in  $f$ .

**Kandinsky Representation.** A Kandinsky embedding of a planar graph can be specified in three stages. First, its topology is fixed by choosing a combinatorial embedding. Second, its shape in terms of angles between edges and sequences of bends on edges is fixed. Third, the geometry is fixed by specifying coordinates for vertices and bend points. In analogy to combinatorial embeddings as equivalence classes of planar drawings with the same topology, one can define *Kandinsky representations* as equivalence classes of Kandinsky embeddings with the same topology and the same shape. This approach was first introduced for orthogonal embeddings [18] and extended to Kandinsky embeddings [13].

Let  $\Gamma$  be a Kandinsky embedding. Let  $f$  be a face with an edge  $e_1$  in its boundary and let  $e_2$  be the successor of  $e_1$  in clockwise direction (counter-clockwise if  $f$  is the outer face). Let further  $v$  be the vertex between  $e_1$  and  $e_2$  and let  $\alpha$  be the angle at  $v$  in  $f$ . We define the *rotation*  $\text{rot}_f(e_1, e_2)$  between  $e_1$  and  $e_2$  to be  $\text{rot}_f(e_1, e_2) = 2 - \alpha/90^\circ$ ; see Fig. 1d. The rotation  $\text{rot}_f(e_1, e_2)$  can be interpreted as the number of right turns between the edges  $e_1$  and  $e_2$  at the vertex  $v$  in the face  $f$ . We also write  $\text{rot}_f(v)$  instead of  $\text{rot}_f(e_1, e_2)$  if the edges are clear from the context and call it the *rotation of  $v$  in  $f$* .

The shape of every edge can also be described in terms of its rotation. Let  $e = uv$  be an edge incident to a face  $f$  such that  $v$  is the clockwise successor of  $u$  along the boundary of  $f$  (counter-clockwise if  $f$  is the outer face). The *rotation*  $\text{rot}_f(e)$  of  $e$  in  $f$  is the number of right bends minus the number of left bends one encounters, when traversing  $e$  from  $u$  to  $v$ ; see Fig. 1e.

Let  $uv, vw$  be a path of length 2 in the face  $f$ . If  $uv$  and  $vw$  form an angle of  $0^\circ$  ( $\text{rot}_f(v) = 2$ ), at least one of the edges  $uv$  or  $vw$  has a bend close to  $v$  with rotation  $-1$  in  $f$  (bend-or-end property). We represent the information of which bends are close to vertices as follows. If  $uv$  has a bend close to  $v$ , we define the *rotation*  $\text{rot}_f(uv[v])$  at the end  $v$  of  $uv$  to be 1 ( $-1$ ) if it has rotation 1 ( $-1$ ) in  $f$ . If  $uv$  has no bend close to  $v$ , we set  $\text{rot}_f(uv[v]) = 0$ .

A set of values for the rotations is a Kandinsky representation (i.e., there is a corresponding embedding) if and only if it satisfies the following properties [13].

- (1) The sum over all rotations in a face is 4 (−4 for the outer face).
- (2) For every edge  $uv$  with incident faces  $f_\ell$  and  $f_r$ , we have  $\text{rot}_{f_\ell}(uv) + \text{rot}_{f_r}(uv) = 0$ ,  $\text{rot}_{f_\ell}(uv[u]) + \text{rot}_{f_r}(uv[u]) = 0$ , and  $\text{rot}_{f_\ell}(uv[v]) + \text{rot}_{f_r}(uv[v]) = 0$ .
- (3) The sum of rotations around a vertex  $v$  is  $2 \cdot \text{deg}(v) - 4$ .
- (4) The rotations at vertices lie in the range  $[-2, 2]$ .
- (5) If  $\text{rot}_f(uv, vw) = 2$  then  $\text{rot}_f(uv[v]) = -1$  or  $\text{rot}_f(vw[v]) = -1$ .

If the face is clear from the context, we often omit the subscript in  $\text{rot}_f$ . One can assume that all bends on an edge (except for bends close to vertices) have the same direction. It follows that the actual number of bends of  $uv$  can be computed from  $\text{rot}(uv)$ ,  $\text{rot}(uv[u])$ , and  $\text{rot}(uv[v])$ .

Let  $f$  be a face of  $G$  and let  $u$  and  $v$  be two vertices on the boundary of  $f$ . By  $\pi_f(u, v)$  we denote the path from  $u$  to  $v$  on the boundary of  $f$  in clockwise direction (counter-clockwise for the outer face). The rotation  $\text{rot}_f(\pi)$  of a path  $\pi$  in the face  $f$  is the sum of all rotations of edges and inner vertices of  $\pi$  in  $f$ .

An orthogonal embedding is basically a Kandinsky embedding without  $0^\circ$  angles at vertices. Thus, we can define *orthogonal representations* [18] (equivalence class of orthogonal embeddings), by forbidding rotation 2 at vertices.

### 3 Complexity

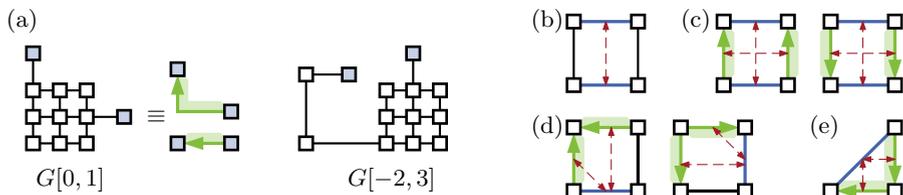
Let  $\mathcal{S}$  be an instance of 3-SAT. In its *variable-clause graph*, the variables and clauses are represented by vertices and there is an edge  $xc$  connecting a variable  $x$  with a clause  $c$  if and only if  $x \in c$  or  $\neg x \in c$ . The NP-hard problem PLANAR MONOTONE 3-SAT [8] restricts the instances of 3-SAT as follows. Every clause contains only positive or only negative literals. Moreover, the variable-clause graph admits a planar embedding such that the edges connecting a variable  $x$  to its positive clauses appear consecutively around  $x$ .

The problem ORTHOGONAL 01-EMBEDDABILITY is defined as follows. Let  $G = (V, E)$  be a 4-plane graph having its edges  $E = E_0 \cup E_1$  partitioned into 0-edges ( $E_0$ ) and 1-edges ( $E_1$ ). Decide whether  $G$  admits an *orthogonal 01-representation* such that every edge in  $E_i$  has exactly  $i$  bends. In the following, we always consider the variant of ORTHOGONAL 01-EMBEDDABILITY where we allow to fix angles at vertices. Fixing the angles at vertices does not make the problem harder since augmenting a vertex  $v$  to have degree 4 by adding degree-1 vertices incident to  $v$  has the same effect as fixing the angles at  $v$ .

We first reduce PLANAR MONOTONE 3-SAT to ORTHOGONAL 01-EMBEDDABILITY, which is further reduced to KANDINSKY BEND MINIMIZATION.

#### 3.1 Orthogonal 01-Embeddability

In the reduction from PLANAR MONOTONE 3-SAT, the decision of setting a variable to `true` or `false` is encoded in the bend-direction of a 1-edge. We show how to build gadgets for variables (outputting a positive and negative literal) and for clauses



**Fig. 2.** (a) The interval gadgets  $G[0, 1]$  ( $\equiv$  01-edge) and  $G[-2, 3]$  ( $s$  and  $t$  are blue). (b–e) Edges are color-coded; 0-edges are black; 1-edges are blue; 01-edges are green and directed such that they may bend right but not left. The building blocks are (b) the box; (c) the bendable box; (d) the merger; (e) the splitter.

(admitting drawings if and only if at least one input edge encodes the value `true`). To carry the decision of one variable to several clauses we need gadgets that impose the bend direction from one edge on multiple edges (literal duplicator). Finally, we build bendable pipes to carry the information (in a flexible way) to the clause gadgets. We first present some basic building blocks.

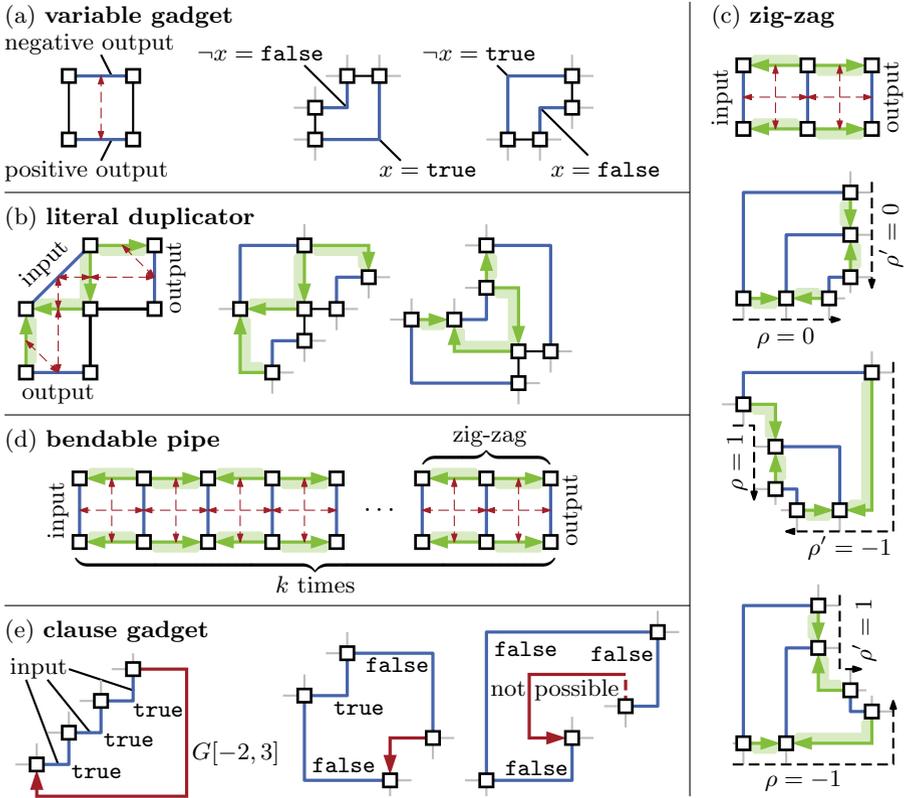
**Building Blocks.** An interval gadget  $G[\rho_1, \rho_2]$  is a graph with two designated degree-1 vertices (its *endpoints*)  $s$  and  $t$  on the outer face. It has the property that  $\text{rot}(\pi(s, t)) \in [\rho_1, \rho_2]$  for any orthogonal embedding. The construction is similar to the tendrils used by Garg and Tamassia [14]; Fig. 2a shows  $G[0, 1]$  and  $G[-2, 3]$ . Note that  $G[0, 1]$  behaves like an edge that may have one bend, but only into a fixed direction. In the following, we draw copies of  $G[0, 1]$  as directed green edges and refer to them as 01-edges.

All our gadgets are based on the building blocks shown in Fig. 2b–e. We require that the angles at the vertices in the internal face  $f$  are  $90^\circ$  (rotation 1). Note that, apart from the 0-edges, all edges of the building blocks admit precisely two possible rotation values in each face. Thus, each edge attains its maximum rotation value in one of its faces and the minimum rotation in the other. It can be shown that in any orthogonal 01-representation the rotation values of some edge pairs are not independent but are linked in the sense that exactly one of them must attain its minimum (maximum) rotation value in  $f$ . In Fig. 2b–e such dependencies are displayed as red dashed arrows.

**Gadget Constructions.** Our gadgets will always have 1-edges on the outer face, whose bend directions represent truth values (as output or as input). We again use red dashed arrows to indicate which edges have to bend consistently. It follows that when there is a path of such red arrows from one edge to another edge, then they are synchronized.

The *variable gadget* for a variable  $x$  consists of a single box. The two 1-bend edges are called *positive* and *negative output*. The variable gadget has exactly two different representations; see Fig. 3a. We interpret a rotation of  $-1$  and  $1$  of the positive output in the outer face as  $x = \text{true}$  and  $x = \text{false}$ , respectively.

The *literal duplicator* is formed by a splitter, which is glued to two mergers via its 01-edges; see Fig. 3b. It has one *input edge* and two *output edges* and transfers the state of the input to both outputs in every orthogonal 01-representation (red dashed paths),



**Fig. 3.** The different gadgets we use in our construction

i.e., the output edges have rotation  $-1$  ( $1$ ) in the outer face if and only if the input edge has rotation  $1$  ( $-1$ ). The literal duplicator admits orthogonal 01-representations for both inputs *true* and *false*; see Fig. 3b.

A *zig-zag* consists of the two bendable boxes glued along a pair of 1-edges; see Fig. 3c. It has an input and an output edge, and in any valid drawing the information encoded in the input is transmitted to the output. Moreover, the decision which of the bendable boxes bend their 01-edges can be taken independently. Thus, the zig-zag allows to choose the rotations  $\rho, \rho'$  of the paths between the input and the output edge with  $\rho = -\rho'$  for each  $\rho \in \{-1, 0, 1\}$ ; see the drawings in Fig. 3c. A *k-bendable pipe* is obtained by concatenating  $k$  zig-zags; see Fig. 3d. It has the same properties as a zig-zag, except that the rotation  $\rho$  can be in the interval  $[-k, k]$ . In a high-level view, a bendable pipe looks like a flexible edge that transfers information between its endpoints.

The *clause gadget* is a cycle of length 4, consisting of three 1-edges, the *input edges*, and the interval gadget  $G[-2, 3]$ ; see Fig. 3e. The inner face lies to the right of  $G[-2, 3]$  (i.e., the rotation of  $G[-2, 3]$  in the inner face is in  $[-2, 3]$ ) and the angles at vertices in

inner faces are fixed to  $90^\circ$ . Interpreting a rotation of  $-1$  (of  $1$ ) of an input edge in the inner face as `true` (as `false`), we get a valid embedding if and only if not all inputs are `false`; see Fig. 3e.

**Putting Things Together.** Let  $S$  be an instance of PLANAR MONOTONE 3-SAT. To obtain the graph  $G(S)$ , we create a variable gadget for every variable and a clause gadget for every clause, duplicate the literals (using the literal duplicator) outputted by the variable gadget as often as they occur in clauses, and connect the resulting output edges with the corresponding input edges of the clauses using bendable pipes of sufficient length. Note that  $G(S)$  is planar if we adhere to the planar embedding of the variable-clause graph of  $S$ .

If  $G(S)$  admits an orthogonal 01-representation, the drawings of the variable gadgets imply a truth assignment for the variables in  $S$ . Moreover, it satisfies  $S$ , since a non-satisfied clause would imply an orthogonal 01-representation of a clause gadget with value `false` on every input edge. Conversely, a satisfying truth assignment of  $S$ , completely fixes the orthogonal 01-representation of each gadget, except for the rotations along the bendable pipes. One needs to show that these representations can be plugged together to a representation of the whole graph  $G(S)$ , which is the case if the bendable pipes are sufficiently long.

**Theorem 1.** ORTHOGONAL 01-EMBEDDABILITY is NP-complete.

In fact, we even showed NP-hardness for the case where all angles at vertices incident to 1-edges are fixed. Moreover, it can be seen that both variants remain hard if the combinatorial embedding is fixed up to the choice of an outer face.

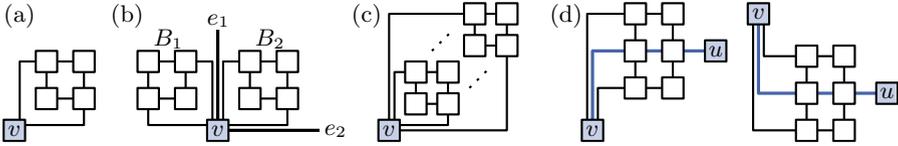
It can be shown that ORTHOGONAL 01-EMBEDDABILITY remains NP-hard for subdivisions of triconnected graphs [3], which have a unique combinatorial embedding. Replacing in such an instance every 1-edge with a copy of the interval gadget  $G[1, 1]$  and releasing the combinatorial embedding gives an equivalent instance of 0-EMBEDDABILITY (variable embedding) where mirroring the embedding of  $G[1, 1]$  corresponds to bending a 1-edge in different directions. This simplifies the hardness proof by Garg and Tamassia [14].

### 3.2 Kandinsky Bend Minimization

The reduction from ORTHOGONAL 01-EMBEDDABILITY to KANDINSKY BEND MINIMIZATION consists of two basic building blocks. In an orthogonal embedding,  $0^\circ$  angles between edges are forbidden. We show how to enforce this for Kandinsky embeddings. Moreover, we construct a subgraph whose Kandinsky embeddings behave like the embeddings of an edge with exactly one bend.

The graph  $B$  in Fig. 4a is called *corner blocker*. The vertex  $v$  is its *attachment vertex*. Clearly,  $B$  admits a Kandinsky representation with two bends. It can be shown that there is no representation with fewer bends and that three bends are necessary if the angle at  $v$  is  $0^\circ$ .

Let  $v$  be a vertex with incident edges  $e_1$  and  $e_2$ . Assume we attach two corner blockers  $B_1$  and  $B_2$  and embed them as in Fig. 4b. Then the angle between  $e_1$  and  $e_2$  cannot be  $0^\circ$  without causing  $B_1$  or  $B_2$  to have three bends. By *nesting* corner blockers



**Fig. 4.** (a) The corner blocker. (b) Two corner blockers enforcing at least  $90^\circ$  angles between  $e_1$  and  $e_2$  (c) Nesting corner blockers. (d) The one-bend gadget.

(Fig. 4c), one can increase this cost arbitrarily. Hence, we can force angles between edges to be at least  $90^\circ$  by adding (nested) corner blockers.

The graph  $\Gamma$  in Fig. 4d with the two *endvertices*  $u$  and  $v$  is called *one-bend gadget*. The path  $\pi$  from  $u$  to  $v$  (blue in Fig. 4d) is the *bending path* of  $\Gamma$ . A Kandinsky representation of  $\Gamma$  *blocks no corner* if all three edges incident to  $v$  leave  $v$  on the same side. Clearly,  $\Gamma$  admits Kandinsky representations blocking no corner with three bends and rotation 1 and  $-1$  on  $\pi$ . We can show that an optimal representation of  $\Gamma$  blocking no corner requires three bends and rotation either  $-1$  or  $1$  on  $\pi$ . Thus,  $\Gamma$  behaves like a 1-edge.

Let  $G = (V, E = E_0 \cup E_1)$  (with combinatorial embedding) be an instance of ORTHOGONAL 01-EMBEDDABILITY. We assume that all angles at vertices incident to a 1-edge in  $G$  are fixed. Starting with  $G$ , we construct a graph  $G'$  that serves as instance of KANDINSKY BEND MINIMIZATION. Let  $v$  be a vertex of  $G$ . If the angles at  $v$  are not fixed, we add a nested corner blocker for every face incident to  $v$ , which forbids  $0^\circ$  angles between edges of  $G$  incident to  $v$ . If the angles are fixed, we add  $\alpha/90^\circ$  nested corner blockers into a face with angle  $\alpha$ , which enforces the correct angles. Then we replace every 1-edge  $uv$  in  $G$  with a one-bend gadget  $\Gamma$ . As the angles around  $v$  were fixed ( $v$  is incident to a 1-edge),  $\Gamma$  is forced to block no corner. Hence,  $\Gamma$  has at least three bends in every Kandinsky representation of  $G'$  and its bending path has rotation 1 or  $-1$ .

We show that  $G$  admits an orthogonal 01-representation if and only if  $G'$  has a Kandinsky representation with  $2b + 3|E_1|$  bends ( $b$  is the number of corner blockers). Given an orthogonal 01-representation of  $G$ , one can add the corner blockers (two bends each) and replace 1-edges by one-bend gadgets (three bends each). Conversely, given a Kandinsky representation of  $G'$  with  $2b + 3|E_1|$  bends, removing the corner blockers and replacing the one-bend gadgets by edges with one bend gives an orthogonal 01-representation. The construction still works when allowing empty faces or restricting edges to have at most one bend (or both).

**Theorem 2.** KANDINSKY BEND MINIMIZATION is NP-complete.

## 4 A Subexponential Algorithm

In this section, we give a subexponential algorithm for computing optimal Kandinsky representations of plane graphs. We use dynamic programming on sphere cut decompositions, which are special types of branch decompositions [9]. Assume graph  $G$  is

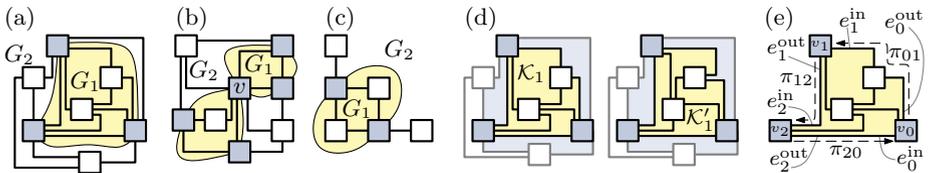
decomposed into subgraphs  $G_1$  and  $G_2$ . It may be possible to merge Kandinsky representations  $\mathcal{K}_1$  and  $\mathcal{K}_2$  of  $G_1$  and  $G_2$  into a representation of  $G$ . We show (Sect. 4.1) which properties of  $\mathcal{K}_1$  are important when trying to merge it with  $\mathcal{K}_2$  and derive classes of Kandinsky representations whose members behave equivalently. If we know optimal Kandinsky representations of  $G_1$  and  $G_2$  for each of these equivalence classes, we find an optimal representation of  $G$  by trying to merge every pair of representations of  $G_1$  and  $G_2$ . We bound the number of combinations one has to consider in Sect. 4.2. Iteratively applying this merging step in a sphere cut decomposition results in our Algorithm (Sect. 4.3).

### 4.1 Interfaces of Kandinsky Representations

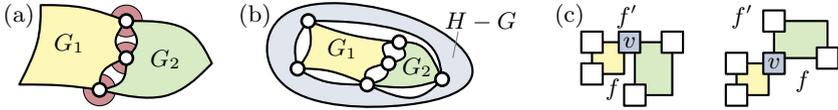
Consider two edge-disjoint graphs  $G_1$  and  $G_2$  sharing a set of *attachment vertices*. Let the union  $G$  of  $G_1$  and  $G_2$  be plane. We say that  $G_1$  and  $G_2$  are *glueable* if both graphs are connected and there is a simple closed curve that separates  $G_1$  from  $G_2$ ; see Fig. 5a–c. We also say that  $G_1$  ( $G_2$ ) is a *glueable subgraph* of  $G$ . A sphere cut decomposition of width  $k$  basically recursively decomposes a plane graph into glueable subgraphs with at most  $k$  attachment vertices.

Let  $\mathcal{K}$  be a Kandinsky representation of  $G$  with restriction  $\mathcal{K}_1$  to  $G_1$ . Let  $\mathcal{K}'_1$  be another representation of  $G_1$ . *Replacing  $\mathcal{K}_1$  with  $\mathcal{K}'_1$  in  $\mathcal{K}$*  means to set every rotation in  $\mathcal{K}$  involving only edges in  $G_1$  to its value in  $\mathcal{K}'_1$  (other values remain unchanged). The result is not necessarily a Kandinsky representation. We say that  $\mathcal{K}_1$  and  $\mathcal{K}'_1$  have *the same interface* if replacing  $\mathcal{K}_1$  with  $\mathcal{K}'_1$  (and vice versa) in any Kandinsky representation of  $G$  yields a Kandinsky representation of  $G$ ; see Fig. 5d. In the following we derive a combinatorial description of an interface.

Consider two glueable subgraphs  $G_1$  and  $G_2$  of a plane graph  $G$ . Let  $v_0, \dots, v_\ell$  be the attachment vertices in the order they appear on the simple closed curve separating  $G_1$  from  $G_2$ . Let  $f$  be the face of  $G_1$  containing  $G_2$  and let  $C_f$  be its facial cycle ( $C_f$  contains  $v_0, \dots, v_\ell$  in that order). The  $v_i$  decompose  $C_f$  into the *interface paths*  $\pi_{01}, \pi_{12}, \dots, \pi_{\ell 0}$  with  $\pi_{ij} = \pi(v_i, v_j)$ . For an attachment vertex  $v_i$ , denote the last edge of the path  $\pi_{i-1 i}$  by  $e_i^{\text{in}}$  and the first edge of the path  $\pi_{i i+1}$  by  $e_i^{\text{out}}$  (indices are considered modulo  $\ell + 1$ ); see Fig. 5e.



**Fig. 5.** (a) Decomposition of a graph into glueable subgraphs  $G_1$  and  $G_2$  (attachment vertices are shaded blue). (b) A non-glueable decomposition (a closed curve separating  $G_1$  from  $G_2$  cannot be simple as  $v$  must be visited twice). (c) Non-glueable decomposition ( $G_2$  is disconnected). (d) Graph  $G$  with glueable subgraph  $G_1$  (yellow). Faces shared by  $G_1$  and  $G_2$  are blue. The Kandinsky representations  $\mathcal{K}_1$  and  $\mathcal{K}'_1$  are interchangeable. (e) Some notation.



**Fig. 6.** (a) Merging  $G_1$  and  $G_2$ . Shared rotations are marked red. (b) A merging step of width 5. (c) Two ways to choose the shared rotations.

The representations  $\mathcal{K}_1$  and  $\mathcal{K}'_1$  of  $G_1$  have *compatible interface paths* if each  $\pi_{i\ i+1}$  has the same rotation in  $\mathcal{K}_1$  and  $\mathcal{K}'_1$ . They have *the same attachment rotations* if for every attachment vertex  $v_i$ , the rotation  $\text{rot}(e_i^{\text{in}}, e_i^{\text{out}})$  is the same. In Fig. 5e, interface paths  $\pi_{01}$ ,  $\pi_{12}$ , and  $\pi_{20}$  have rotations  $-1$ ,  $1$ , and  $0$ , and the attachment rotations at  $v_0$ ,  $v_1$ , and  $v_2$  are  $-1$ ,  $-1$ , and  $-2$ , respectively.

For an attachment vertex  $v_i$ , the rotations at the end  $v_i$  of the edges  $e_i^{\text{in}}$  and  $e_i^{\text{out}}$  ( $\text{rot}(e_i^{\text{in}}[v_i])$  and  $\text{rot}(e_i^{\text{out}}[v_i])$ ) indicate whether  $0^\circ$  angles at  $v_i$  are allowed. For both rotations, we define the  $0^\circ$  *flag* to be `true` if a  $0^\circ$  angle is allowed (rotation  $-1$ ) and `false` otherwise (rotations  $0, 1$ ). Possible values for the  $0^\circ$  flags in Fig. 5e are `true` for  $e_0^{\text{out}}[v_0]$  and for  $e_1^{\text{in}}[v_1]$  and `false` for all other flags.

**Lemma 1.** *Two Kandinsky representations have the same interface iff they have compatible interface paths, the same attachment rotations, and the same  $0^\circ$  flags.*

It follows that each interface class is uniquely described by this information. We simply call it the *interface* of  $G_1$  ( $G_2$ ) in  $G$ .

## 4.2 Merging Two Kandinsky Representations

Let  $\mathcal{K}_1$  and  $\mathcal{K}_2$  be Kandinsky representations of  $G_1$  and  $G_2$ , respectively, and let  $G = G_1 \cup G_2$ . We say that  $\mathcal{K}_1$  and  $\mathcal{K}_2$  can be *merged* if there exists a Kandinsky representation  $\mathcal{K}$  of  $G$  whose restriction to  $G_1$  and  $G_2$  is  $\mathcal{K}_1$  and  $\mathcal{K}_2$ , respectively. Note that the only rotations in  $\mathcal{K}$  that occur neither in  $\mathcal{K}_1$  nor in  $\mathcal{K}_2$  are rotations at attachment vertices between an edge of  $G_1$  and an edge of  $G_2$ . We call these rotations the *shared rotations*; see Fig. 6a. Thus, merging  $\mathcal{K}_1$  and  $\mathcal{K}_2$  is the process of choosing values for the shared rotation such that the resulting set of rotations is a Kandinsky representation of  $G$ .

In the following, we consider the case where  $G$  itself is a glueable subgraph of a larger graph  $H$ . We call this the *merging step*  $G = G_1 \sqcup G_2$ . Note that  $G_1$  and  $G_2$  are also glueable subgraphs of  $H$ . Note further that the interface of  $G_1$  ( $G_2$ ) in  $G$  can be deduced from the interface of  $G_1$  ( $G_2$ ) in  $H$ . When dealing with a merging step, we always consider the interfaces of  $G_1$  and  $G_2$  in  $H$ . The *width* of a merging step is the maximum number of attachment vertices of  $G_1$ ,  $G_2$ , and  $G$  in  $H$ ; see Fig. 6b for an example.

If the Kandinsky representations  $\mathcal{K}_1$  and  $\mathcal{K}_2$  can be merged, then every Kandinsky representation  $\mathcal{K}'_1$  with the same interface as  $\mathcal{K}_1$  can be merged in the same way (i.e., with the same shared rotations) with  $\mathcal{K}_2$ . Moreover, the resulting Kandinsky representations  $\mathcal{K}$  and  $\mathcal{K}'$  of  $G$  have the same interface. Thus, the only choices that matter when

merging two representations are to choose shared rotations and interfaces for  $G_1$  and  $G_2$ . A choice of shared rotations and interfaces is *compatible* if these interfaces can be merged using the chosen rotations.

We bound the number of compatible combinations, depending on the width  $k$  of the merging step and the *maximum rotation*  $\rho$ . The maximum rotation of a graph  $H$  is  $\rho$  if  $H$  admits an optimal Kandinsky representation such that the absolute rotations of the interface paths in every glueable subgraph of  $H$  are at most  $\rho$ ; the maximum rotation  $\rho$  of a merging step refers to the maximum rotation of the whole graph  $H$ .

A simple bound can be obtained as follows. There are  $2k$  interface paths, each admitting up to  $(2\rho+1)$  possible rotations, giving  $(2\rho+1)^{2k}$  combinations. Every attachment vertex has its attachment rotation in  $[-2, 2]$  and two binary  $0^\circ$ -flags, yielding another  $20^{2k}$  combinations. Finally, each shared rotation (two per attachment) may be chosen from  $[-2, 2]$ , yielding again  $5^{2k}$  combinations. That are  $(2\rho+1)^{2k}10000^k$  combinations in total. By a careful consideration which combinations are actually meaningful this number can be reduced greatly.

**Lemma 2.** *In a merging step  $G = G_1 \sqcup G_2$  of width  $k$  with maximum rotation  $\rho$ , there are at most  $(2\rho+1)^{\lfloor 1.5k \rfloor - 1} \cdot 330^k$  compatible choices for the shared rotations and the interfaces of  $G_1$  and  $G_2$ .*

Let  $G$  be a glueable subgraph of  $H$ . The *cost* of an interface class is the minimum cost (e.g., number of bends) of the Kandinsky representations it contains. The *cost table* of  $G$  is a table containing the cost of each interface class of  $G$ .

**Lemma 3.** *Let  $G = G_1 \sqcup G_2$  be a merging step of width  $k$  with maximum rotation  $\rho$ . Given the cost tables of  $G_1$  and  $G_2$ , the cost table of  $G$  can be computed in  $O(k \cdot (2\rho+1)^{\lfloor 1.5k \rfloor - 1} \cdot 330^k)$  time.*

### 4.3 The Algorithm

The previous three lemmas together with an optimal sphere cut decomposition (computable in  $O(n^3)$  time [15,9]) can be used to prove the following theorem.

**Theorem 3.** *An optimal Kandinsky representation of a plane graph  $G$  can be computed in  $O(n^3 + n \cdot k \cdot (2\rho+1)^{\lfloor 1.5k \rfloor - 1} \cdot 330^k)$  time, where  $k$  is the branch width and  $\rho$  the maximum rotation of  $G$ .*

To obtain the following corollaries, we bound  $\rho$  in terms of the optimal bend number and the maximum face size and use upper bounds of 2 and  $O(\sqrt{n})$  on the branch width of series-parallel and planar graphs, respectively.

**Corollary 1.** *Let  $G$  be a plane graph with maximum face-degree  $\Delta_F$ , and branch width  $k$ . An optimal Kandinsky representation can be computed in  $O(n^3 + n \cdot k \cdot (2m + 2\Delta_F - 3)^{\lfloor 1.5k \rfloor - 1} \cdot 330^k)$  time. An optimal  $b$ -bend Kandinsky representation can be computed in  $O(n^3 + n \cdot k \cdot ((2b+2) \cdot \Delta_F - 2b - 3)^{\lfloor 1.5k \rfloor - 1} \cdot 330^k)$  time.*

**Corollary 2.** *For series-parallel and general plane graphs an optimal Kandinsky representation can be computed in  $O(n^3)$  and  $2^{O(\sqrt{n} \log n)}$  time, respectively.*

**Acknowledgments.** We thank Therese Biedl for discussions.

## References

1. Bertolazzi, P., Di Battista, G., Didimo, W.: Computing orthogonal drawings with the minimum number of bends. *IEEE Trans. Comput.* 49(8), 826–840 (2000)
2. Biedl, T., Kant, G.: A better heuristic for orthogonal graph drawings. *Comput. Geom. Theory Appl.* 9, 159–180 (1998)
3. Bläsius, T., Brückner, G., Rutter, I.: Complexity of higher-degree orthogonal graph embedding in the kandinsky model. *CoRR abs/1405.2300* (2014)
4. Bläsius, T., Krug, M., Rutter, I., Wagner, D.: Orthogonal graph drawing with flexibility constraints. *Algorithmica* 68(4), 859–885 (2014)
5. Bläsius, T., Lehmann, S., Rutter, I.: Orthogonal graph drawing with inflexible edges. *CoRR abs/1404.2943* (2014)
6. Bläsius, T., Rutter, I., Wagner, D.: Optimal orthogonal graph drawing with convex bend costs. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) *ICALP 2013, Part I. LNCS*, vol. 7965, pp. 184–195. Springer, Heidelberg (2013)
7. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. *J. Graph Algorithms Appl.* 16(3), 635–650 (2012)
8. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geometry Appl.* 22(3), 187–206 (2012)
9. Dorn, F., Penninx, E., Bodlaender, H., Fomin, F.: Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica* 58(3), 790–810 (2010)
10. Eiglsperger, M.: Automatic Layout of UML Class Diagrams: A Topology-Shape-Metrics Approach. Ph.D. thesis, Universität Tübingen (2003)
11. Eiglsperger, M., Gutwenger, C., Kaufmann, M., Kupke, J., Jünger, M., Leipert, S., Klein, K., Mutzel, P., Siebenhaller, M.: Automatic layout of UML class diagrams in orthogonal style. *Information Visualization* 3(3), 189–208 (2004)
12. Fößmeier, U., Kant, G., Kaufmann, M.: 2-visibility drawings of planar graphs. In: North, S. (ed.) *GD 1996. LNCS*, vol. 1190, pp. 155–168. Springer, Heidelberg (1997)
13. Fößmeier, U., Kaufmann, M.: Drawing high degree graphs with low bend numbers. In: Brandenburg, F.J. (ed.) *GD 1995. LNCS*, vol. 1027, pp. 254–266. Springer, Heidelberg (1996)
14. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.* 31(2), 601–625 (2001)
15. Gu, Q.P., Tamaki, H.: Optimal branch-decomposition of planar graphs in  $O(n^3)$  time. *ACM Trans. Alg.* 4(3), 30:1–30:13 (2008)
16. Klau, G.W., Mutzel, P.: Quasi-orthogonal drawing of planar graphs. Research Report MPI-I-98-1-013, Max-Planck-Institut für Informatik (1998)
17. Leiserson, C.E.: Area-efficient graph layouts (for VLSI). In: *FOCS 1980*, pp. 270–281 (1980)
18. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* 16(3), 421–444 (1987)
19. Tamassia, R. (ed.): *Handbook of Graph Drawing and Visualization*. No. 81 in *Discrete Mathematics and Its Applications*. Chapman and Hall/CRC (2013)
20. Tamassia, R., Battista, G.D., Batini, C.: Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst., Man, Cybern., Syst.* 18, 61–79 (1988)
21. Valiant, L.G.: Universality considerations in VLSI circuits. *IEEE Trans. Comput.* 30(2), 135–140 (1981)