

Simple Realizability of Complete Abstract Topological Graphs in \mathbb{P}

Jan Kynčl

Received: 22 October 2007 / Revised: 20 October 2010 / Accepted: 30 November 2010 /

Published online: 7 January 2011

© Springer Science+Business Media, LLC 2011

Abstract An *abstract topological graph* (briefly an *AT-graph*) is a pair $A = (G, \mathcal{R})$ where $G = (V, E)$ is a graph and $\mathcal{R} \subseteq \binom{E}{2}$ is a set of pairs of its edges. An AT-graph A is *simply realizable* if G can be drawn in the plane in such a way that each pair of edges from \mathcal{R} crosses exactly once and no other pair crosses. We present a polynomial algorithm which decides whether a given complete AT-graph is simply realizable. On the other hand, we show that other similar realizability problems for (complete) AT-graphs are NP-hard.

Keywords Abstract topological graph · Simple complete topological graph · Simple realizability

1 Introduction

A *topological graph* $T = (V(T), E(T))$ is a drawing of an (abstract) graph G in the plane with the following properties. The vertices of G are represented by a set $V(T)$ of distinct points in the plane and the edges of G are represented by a set $E(T)$ of simple curves connecting the corresponding pairs of points. We call the elements of $V(T)$ and $E(T)$ *vertices* and *edges* of T . The edges cannot pass through any vertices except their end-points. Any intersection point of two edges is either a common end-point or a *crossing*, a point where the two edges properly cross (“touching” of the edges is not allowed). We also require that any two edges have only finitely many intersection points and that no three edges pass through a single crossing. A topological

A preliminary version appeared in proceedings of Graph Drawing 2007 [19].

ITI is supported by the project 1M0545 of the Ministry of Education of the Czech Republic.

J. Kynčl (✉)

Department of Applied Mathematics and Institute for Theoretical Computer Science, Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic
e-mail: kyncl@kam.mff.cuni.cz

graph is *simple* if every two edges have at most one common point (which is either a common end-point or a crossing). A topological graph is *complete* if it is a drawing of a complete graph.

An *abstract topological graph* (briefly an *AT-graph*), a notion established in [15], is a pair (G, \mathcal{R}) where G is a graph and $\mathcal{R} \subseteq \binom{E(G)}{2}$ is a set of pairs of its edges. For a topological graph T which is a drawing of G we define \mathcal{R}_T as a set of pairs of edges having at least one common crossing and we say that (G, \mathcal{R}_T) is an AT-graph of T . A topological graph T is called a *realization* of (G, \mathcal{R}) if $\mathcal{R}_T = \mathcal{R}$. If $\mathcal{R}_T \subseteq \mathcal{R}$, then T is called a *weak realization* (or also a *feasible drawing*) of (G, \mathcal{R}) . If (G, \mathcal{R}) has a (weak) realization, we say that (G, \mathcal{R}) is (*weakly*) *realizable*. We say that (G, \mathcal{R}) is *simply (weakly) realizable* if (G, \mathcal{R}) has a simple (weak) realization, that is, a drawing which is a simple topological graph. We say that (G, \mathcal{R}) is (*weakly*) *rectilinearly realizable* if it has a (weak) realization T with edges drawn as straight-line segments (such drawing T is called a (*weak*) *rectilinear realization* of (G, \mathcal{R})).

Complete topological graphs are one of the most studied classes of topological graphs [11, 20, 22, 25, 27], especially regarding the crossing number problems [1, 9, 29, 33, 35].

We study the complexity of various realizability problems for AT-graphs and complete AT-graphs. For example, the *realizability* problem is defined as follows: the instance is an AT-graph A and the question is whether A is realizable. *Weak realizability*, *simple realizability*, *simple weak realizability*, *rectilinear realizability* and *weak rectilinear realizability* are defined similarly.

Kratochvíl [13] proved that realizability and weak realizability are NP-hard (for the class of all AT-graphs). For a long time, the decidability of these problems was an open question. Schaefer and Štefankovič [32] gave a positive answer by showing that weak realizability is in NEXP. Pach and Tóth [26] suggested an alternative approach to show decidability, but their proof has been found incorrect [21]. Later Schaefer, Sedgwick and Štefankovič [31] improved the upper bound to NP, thus finishing the proof of the NP-completeness of weak realizability. Since the related problem of recognizing string graphs is NP-hard [13], polynomially reducible to weak realizability [17] and polynomially equivalent to realizability [13], the string graph problem and the realizability problem are NP-complete as well.

It is also known that simple realizability is NP-complete [16] and rectilinear realizability is NP-hard [16], even when the edges are restricted to have only 2 directions [14]; these results follow from the corresponding results for intersection graphs of pseudosegments and segments, respectively.

Some of these realizability problems can be shown to be NP-hard even for complete graphs. For realizability and weak realizability we can use a simple reduction from the version for all graphs: to a given AT-graph, we add all the missing edges and extend the corresponding set \mathcal{R} by all pairs of edges containing at least one of the added edges. In the realizability case, the added edges can be drawn so that they cross all the other edges at least once. In the weak realizability case, the new edges can be drawn arbitrarily.

As noted by the anonymous referee, there is also a common polynomial reduction from weak realizability (of AT-graphs) to simple weak realizability of complete AT-graphs and weak rectilinear realizability of complete AT-graphs. Kratochvíl's reduction [13] from planar 3-connected 3-SAT produces instances of weak realizability

that can be drawn with only polynomial number of crossings (namely $O(n^2)$, where n is the number of vertices). Take such an AT-graph A . We subdivide each edge of A into $O(n^2)$ shorter edges and allow a pair of these new edges to cross if and only if the original edges they belonged to were allowed to cross. We obtain an AT-graph A' which is weakly realizable if and only if A is. Moreover, in this case A' has a weak rectilinear realization. Indeed, we can use the new vertices of A' to cover the crossings of the drawing of A and to subdivide each portion of every edge between two consecutive crossings into two parts. We obtain a planar graph, which has a rectilinear drawing by Fáry's theorem. Then we perturb the vertices a little bit to split the pairs of the new vertices covering the same crossing. Now we add all the missing edges to A' and allow them to cross all the other edges, as in the case of weak realizability. We obtain a complete AT-graph A'' which is weakly realizable if and only if A is, and in this case A'' also has a weak rectilinear (and thus a simple weak) realization.

Kratochvíl and Matoušek [18] showed that the problem of recognizing intersection graphs of segments is polynomially equivalent to deciding truth in the existential theory of the reals, that is, deciding solvability of systems of strict polynomial inequalities in real numbers. This problem is known to be in PSPACE [2] and NP-hard, but is not known to be in NP. Recently Schaefer [30] introduced a complexity class $\exists\mathbb{R}$ that captures the complexity of the existential theory of the reals, and found several other problems of geometric nature that are $\exists\mathbb{R}$ -complete, that is, polynomially equivalent to the existential theory of the reals. In this terminology, recognizing intersection graphs of segments and consequently rectilinear realizability are $\exists\mathbb{R}$ -complete. Note that both realizability and weak realizability trivially belong to $\exists\mathbb{R}$, since the fact that two segments determined by the coordinates of their endpoints cross (or do not cross) can be expressed by a system of quadratic polynomial inequalities; see [18] for details.

Weak rectilinear realizability is also $\exists\mathbb{R}$ -hard. This follows by a polynomial reduction from stretchability of simple pseudoline arrangements, which was shown to be NP-hard by Shor [34] and $\exists\mathbb{R}$ -complete by Mnev [24]. For the reduction, one can use a simpler variant of the Order Forcing Lemma [18]. Given a simple arrangement X of n pseudolines p_1, p_2, \dots, p_n (such that they meet a sufficiently large circle in the cyclic order $p_1, p_2, \dots, p_n, p_1, p_2, \dots, p_n$), we construct an AT-graph A_X consisting of a cycle $v_0v_1 \dots v_{6n-1}$ of length $6n$, n pairwise crossing chords $v_{3i}v_{3i+3n}$ corresponding to the pseudolines and $2n$ paths $P_1, \dots, P_n, P'_1, \dots, P'_n$ of length $n-1$, where P_i connects v_{3i-1} with v_{3i+n+1} and P'_i connects v_{3i+1} with v_{3i+n-1} . The only pairs of edges allowed to cross are all pairs of the chords of the cycle and all pairs $e_{i,j}, c_{i,j}$ and $e'_{i,j}, c'_{i,j}$, where $e_{i,j}$ ($e'_{i,j}$) is the j th edge of a path P_i (P'_i , respectively) and $c_{i,j}$ is the j th pseudoline crossing p_i in X . It is easy to see that A_X has a weak rectilinear realization if and only if X is stretchable. Weak rectilinear realizability remains $\exists\mathbb{R}$ -hard for complete AT-graphs: as before, add all the missing edges and allow them intersect all other edges.

The $\exists\mathbb{R}$ -hardness of rectilinear realizability of complete AT-graphs can be shown by a different argument; see Sect. 3.

The above methods for showing hardness, however, fail in the case of simple realizability of complete AT-graphs. The main result of this paper is the following theorem.

Theorem 1 *There is a polynomial-time algorithm which decides whether a given complete AT-graph A is simply realizable. In the affirmative case, it also gives a combinatorial description of some simple realization of A .*

We prove Theorem 1 in Sect. 4.

The results for all the considered problems are summarized in the following table.

Theorem 2

	AT-graphs	complete AT-graphs
realizability	NP-complete [13, 31]	NP-complete
weak realizability	NP-complete [13, 31]	NP-complete
simple realizability	NP-complete [16]	P
simple weak realizability	NP-complete	NP-complete
rectilinear realizability	$\exists\mathbb{R}$ -complete [18]	$\exists\mathbb{R}$ -complete
weak rectilin. realizability	$\exists\mathbb{R}$ -complete	$\exists\mathbb{R}$ -complete

Weak realizability of AT-graphs is polynomially equivalent to the *simultaneous drawing* problem [7]. The instance of this problem is a graph G given as a union of planar graphs G_1, G_2, \dots, G_k sharing some common edges. The question is whether G can be drawn in the plane so that each of the subgraphs G_i is drawn without crossings. Simultaneous drawing of three planar graphs is known to be an NP-complete problem [7]; this gives an alternative proof of the NP-completeness of weak realizability. The complexity of simultaneous drawing of two planar graphs remains open.

2 Additional Definitions

A *face* of a topological graph T is a connected component of the set $\mathbb{R}^2 \setminus \bigcup E(T)$. A *rotation* of a vertex $v \in V(T)$ is the clockwise cyclic order in which the edges incident with v leave the vertex v . A *rotation system* of the topological graph T is the set of rotations of all its vertices. Similarly, we define a *rotation* of a crossing c of edges uv and xy as the clockwise order in which the four portions cu, cv, cx and cy of the edges uv and xy leave the point c (note that each crossing has exactly two possible rotations). An *extended rotation system* of a topological graph is the set of rotations of all its vertices and crossings.

Assuming that T and T' are drawings of the same abstract graph, we say that their (extended) rotation systems are *inverse* if for each vertex $v \in V(T)$ (and each crossing c in T) the rotation of v and the rotation of the corresponding vertex $v' \in V(T')$ are inverse cyclic permutations (and so are the rotations of c and the corresponding crossing c' in T'). For example, if T' is a mirror image of T , then T and T' have inverse (extended) rotation systems.

Topological graphs G and H are *weakly isomorphic* if there exists an incidence preserving one-to-one correspondence between $V(G), E(G)$ and $V(H), E(H)$ such that two edges of G cross if and only if the corresponding two edges of H do. In

other words, two topological graphs are weakly isomorphic if and only if they are realizations of the same abstract topological graph.

Topological graphs G and H are *isomorphic* if (1) G and H are weakly isomorphic, (2) for each edge e of G the order of crossings with the other edges of G is the same as the order of crossings on the corresponding edge e' in H , and (3) the extended rotation systems of G and H are the same or inverse. This induces a one-to-one correspondence between the faces of G and H such that the crossings and the vertices incident with a face f of G appear along the boundary of f in the same (or inverse) cyclic order as the corresponding crossings and vertices in H appear along the boundary of the face f' corresponding to f .

Assuming that the topological graphs G and H are drawn on the sphere, it follows from the Jordan–Schönflies theorem that G and H are isomorphic if and only if there exists a homeomorphism of the sphere which transforms G into H .

Unlike the isomorphism, the weak isomorphism can change the faces of the involved topological graphs, as well as the order in which one edge crosses other edges.

3 Rectilinear Realizability of Complete AT-Graphs

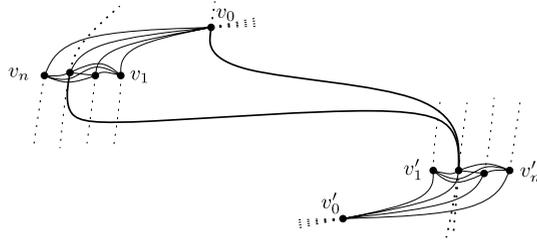
Here we show that rectilinear realizability of complete AT-graphs is $\exists\mathbb{R}$ -complete by a polynomial reduction from stretchability of simple pseudoline arrangements. By duality [8], stretchability of simple pseudoline arrangements is equivalent to stretchability of pseudoarrangements of points (a special type of arrangements with n points and $\binom{n}{2}$ pseudolines, one pseudoline passing through each pair of points). Each such pseudoarrangement can be combinatorially characterized by its order type, which is the set of orientations of all $\binom{n}{3}$ pseudotriangles. Each pseudoarrangement P of n points determines a simple complete topological graph T_P with n vertices. Moreover, by an easy investigation of 4-point pseudoarrangements, the order type of P determines which pairs of edges of T_P cross (i.e., it determines the AT-graph of T_P) and also the (oriented) cycle bounding the outer face of T_P . By Proposition 3(i), this correspondence is one-to-one: the AT-graph of T_P determines the rotation of each vertex of T_P , which together with the outer cycle and its orientation determines the orientation of each triangle in T_P .

Now we construct a pseudoarrangement Q consisting of P and its copy P' , such that the AT-graph of the corresponding topological graph T_Q is rectilinearly realizable if and only if P is stretchable.

Choose a vertex v_0 from the outer face of T_P . The clockwise order of the edges incident with v_0 determines a linear order v_1, v_2, \dots, v_{n-1} of the other vertices of T_P (we start the labeling from the outer face). Let $v'_0, v'_1, v'_2, \dots, v'_{n-1}$ be the corresponding points in a copy P' of P . Now we add all the missing pseudolines joining points of P and P' and describe the order type of the new pseudoarrangement Q .

The orientation of each triangle $v'_i v'_j v'_k$ is the same as the orientation of the triangle $v_i v_j v_k$. For each $i < j$, the counter-clockwise orientation of triangles with vertices from both P and P' is v_i, v_j, v'_k (resp., v'_i, v'_j, v_k). This order type can be realized by taking P' as a central image of P and drawing the missing pseudolines as in Fig. 1.

Fig. 1 Pseudolines $v_3v'_2$ and $v_0v'_2$ added to the pseudoarrangements P and P' . Some pseudolines are drawn only between the two determining points



If T_Q has a rectilinear realization, then at least one of the subgraphs $T_P, T_{P'}$ has the same outer face as the original drawing of T_P , hence the pseudoarrangement P is stretchable.

If P is stretchable, then T_P can be stretched while preserving its outer face. Since the graph T_P is complete, its outer cycle is a convex piecewise linear curve. We rotate the plane such that v_0 becomes the topmost vertex (the vertex with the largest y -coordinate). We also assume that the x -coordinate of v_0 is 0. Then we apply the following projective transformation sending v_0 to infinity: $(x, y) \rightarrow (x, y) \cdot \frac{y_0}{y_0 - y}$ (y_0 is the y -coordinate of v_0). At this point the vertices v_1, v_2, \dots, v_n are ordered by their x -coordinate from left to right as v_n, v_{n-1}, \dots, v_1 . Now we apply an affine transformation $(x, y) \rightarrow (x, y/N)$ with a sufficiently large N . This flattens the graph induced by the vertices v_1, \dots, v_n . Then we apply another projective transformation $(x, y) \rightarrow (x, y) \cdot \frac{M}{M+y}$ with large enough M that moves v_0 back from the infinity but keeps it sufficiently high above the rest of the vertices, and also does not change the horizontal order of the vertices v_1, \dots, v_n . Then we mirror this final drawing of T_P by a suitable center (lying sufficiently below and to the right of v_1) to obtain its copy $T_{P'}$. Finally, we add all the missing lines between the vertices of T_P and $T_{P'}$ and get a rectilinear realization of the AT-graph of T_Q .

4 Recognition of Simply Realizable Complete AT-Graphs

In this section, we present a polynomial-time algorithm for deciding simple realizability of a given complete AT-graph A . In the affirmative case, the algorithm also gives a combinatorial description of the isomorphism class of some simple realization of A . For simplicity, we do not try to optimize the order of the polynomial bounding the computing time of the algorithm.

We begin with the following key observation, which has been independently proved by Gioan [6].

Proposition 3

- (i) *If two simple complete topological graphs are weakly isomorphic, then their extended rotation systems are either the same or inverse.*
- (ii) *For each edge e of a simple complete topological graph G and for each pair of edges $f, f' \in E(G)$ which have a common end-point and cross e , the AT-graph of G determines the order of crossings of e with the edges f, f' .*

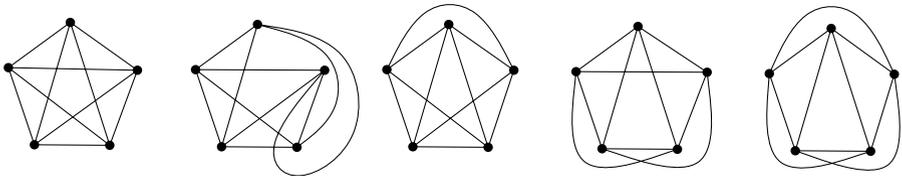


Fig. 2 All five non-isomorphic simple drawings of K_5 [11]

It is also not hard to prove that the rotation system of a simple complete topological graph determines its AT-graph, but we will not need it for our purposes.

We will denote the rotation system of a topological graph G as $\mathcal{R}(G)$ and we will represent it as a sequence of rotations of its vertices. The rotation $\mathcal{R}(v)$ of a vertex v will be represented by a cyclic sequence of the labels of the remaining vertices.

Proof

(i) Let G and G' be two weakly isomorphic simple complete topological graphs on n vertices. First, we prove that the rotation systems $\mathcal{R}(G)$ and $\mathcal{R}(G')$ are either the same or inverse.

For $n \leq 3$, it is trivial; for $n = 4$ and $n = 5$, this follows from the fact that for every simple complete topological graph with 4 or 5 vertices the isomorphism classes coincide with the weak isomorphism classes: there are two non-isomorphic simple drawings of K_4 and five non-isomorphic simple drawings of K_5 (see [11] or Fig. 2) and each of them is a realization of a different AT-graph.

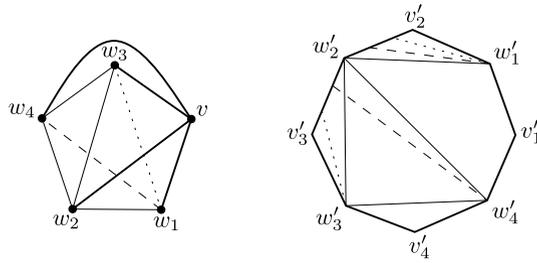
Now we use the case $n = 5$ to extend the statement to graphs with more than five vertices. Let A be a simply realizable complete AT-graph with the vertex set $\{1, 2, \dots, n\}$, where $n \geq 6$. We know that each complete 5-vertex subgraph of A has only two possible rotation systems. Suppose that the rotation system of $A[\{1, 2, 3, 4, 5\}]$, the induced subgraph of A with the vertices 1, 2, 3, 4, 5, is fixed (in some simple realization of A). We show that then the rotation system of every other 5-vertex complete subgraph of A is uniquely determined.

Lemma *Let B and C be two 5-vertex complete subgraphs of A with exactly 4 common vertices. Then the rotation system $\mathcal{R}(B)$ uniquely determines the rotation system $\mathcal{R}(C)$.*

Proof of Lemma Without loss of generality, let $V(B) = \{1, 2, 3, 4, 5\}$, $V(C) = \{1, 2, 3, 4, 6\}$ and let the rotation of the vertex 1 in $\mathcal{R}(B)$ be $(2, 3, 4, 5)$. Then the rotation of 1 in $A[\{1, 2, 3, 4\}]$ is $(2, 3, 4)$ and it must be a subsequence of the rotation of 1 in $\mathcal{R}(C)$. But this always happens for exactly one of the pair of inverse cyclic permutations of the set $\{2, 3, 4, 6\}$. It follows that the rotation of 1 in C is uniquely determined and so is the whole rotation system of C . \square

By repeated use of this lemma, we obtain that the rotation system of every complete subgraph of A on 5 (and also 4) vertices is uniquely determined by $\mathcal{R}(A[\{1, 2, 3, 4, 5\}])$. It remains to show that this also uniquely determines the rotation of each vertex in A . But this easily follows from the fact that a cyclic order of

Fig. 3 A simple drawing of K_5 and its star-cut representation



a finite set X is uniquely determined by the cyclic order of all 3-element subsets of X (actually, it suffices to know the orders of the triples containing one fixed vertex). It follows that a simple realization of A can have only two possible rotation systems.

Since G and its mirror image have inverse extended rotation systems, it remains to prove that $\mathcal{R}(G)$ uniquely determines the rotation $\mathcal{R}(c)$ of each crossing c of G . Let uv, wz be the edges that cross at c . Then $\mathcal{R}(c)$ is determined by the drawing of the induced subgraph $H = G[\{u, v, w, z\}]$. Since every two weakly isomorphic simple drawings of K_4 are isomorphic, and an isomorphism preserves or inverts the extended rotation system, it follows that $\mathcal{R}(c)$ is determined by $\mathcal{R}(H)$, which is trivially determined by $\mathcal{R}(G)$.

(ii) The edges e, f, f' are contained in a complete 5-vertex subgraph H of G , so the order of crossings of e with f and f' is determined by the isomorphism class of H , which is determined by the AT-graph of H . \square

Now we introduce a *star-cut representation* of the graph G . Choose an arbitrary vertex v and denote by w_1, w_2, \dots, w_{n-1} the remaining vertices of G so that $\mathcal{R}(v) = (w_1, w_2, \dots, w_{n-1})$. Let $S(v)$ denote the union of all the edges vw_i of G ($S(v)$ is a “topological star” with central vertex v). If we consider G drawn on the sphere S^2 , the set $S^2 \setminus S(v)$ is mapped by a homeomorphism Φ onto an open regular $2(n - 1)$ -gon D in the plane. We can visualize this by cutting the sphere along the edges of the star $S(v)$ and then unpacking the resulting surface in the plane. The map Φ^{-1} can be continuously extended to the closure of D , giving a natural correspondence between the vertices and edges of D and the vertices and edges in $S(v)$: each vertex w_i corresponds to one vertex w'_i of D and the vertex v of G corresponds to $n - 1$ vertices $v'_1, v'_2, \dots, v'_{n-1}$ of D . If Φ preserves the orientation, the counter-clockwise order of the vertices of D is $v'_1, w'_1, v'_2, w'_2, \dots, v'_{n-1}, w'_{n-1}$. Each edge $vw_i \in E(G)$ splits into two adjacent edges $v'_i w'_i$ and $v'_{i+1} w'_i$; see Fig. 3. During the cutting operation every edge e_k of G not incident with v can be cut into several pieces. Since e_k crosses each edge of $S(v)$ at most once, it is cut into at most n pieces $e_{k,j}$. Every crossing of the edge e_k with an edge vw_i corresponds to two end-points of two different pieces $e_{k,j}, e_{k,j'}$ lying on the edges $v'_i w'_i$ and $v'_{i+1} w'_i$.

Outline of the Algorithm

The algorithm consists of the following main steps.

1. Using Proposition 3(i), we compute from the complete AT-graph A a rotation system of its simple realization (if such drawing exists).

2. Using Proposition 3, we construct a partial star-cut representation of the drawing. More precisely, we compute the order (and rotations) of crossings of every edge with the edges of the star $S(v)$ (the orders of crossings on the edges of $S(v)$ will be determined later). In other words, we determine the homotopy type of every edge in a surface which is obtained from the sphere after drilling a hole around each vertex of the drawing.

3. First, we test each edge for self-intersections. Then for each pair of edges e, f we compute $\text{CR}(e, f)$, the minimum number of crossings of e and f that are forced by their homotopy types. It is known that for arbitrary 2-dimensional surface with or without boundary, the numbers $\text{CR}(e, f)$ can be computed in polynomial time [3, 4, 10, 23, 28]. We show the algorithm here for our special case, where the surface is the sphere with n holes and the homotopy types of the curves are given by the sequence of crossings with the edges of $S(v)$.

From the homotopy class of each edge, we can choose a representative so that in the resulting drawing the number of crossings of each pair of edges e, f attains the lower bound $\text{CR}(e, f)$. The existence of such representatives can be proved, for example, from the following fact.

Lemma 4 [5, 12] *Let C_1 and C_2 be two simple curves on a surface S such that the endpoints of C_1 and C_2 lie on the boundary of S . If C_1 can be homotoped (rel boundary of S) to decrease the number of crossings with C_2 , then there are two sub-arcs of C_1 and C_2 bounding a disc in S whose interior is disjoint with C_1 and C_2 .*

Since the parity of the number of crossings of each pair of edges is also determined by their homotopy types, by comparing the numbers $\text{CR}(e, f)$ with the AT-graph A we can decide whether A is simply realizable.

4. To find some particular simple realization of A , we first find such orders of crossings on the edges of $S(v)$ that induce $\text{CR}(e, f)$ crossings for each pair of edges e, f (this is, in fact, an effective proof of the result in the previous step).

5. Finally, we determine the orders of crossings also on all edges not incident with v .

The Algorithm

Let A be a given complete AT-graph with the vertex set $\{1, 2, \dots, n\}$.

Step 1: Rotation System

In the first step of the algorithm, we compute the (abstract) rotation system $\mathcal{R}(A)$, i.e., the rotation system of a simple realization of A , if it exists, by an effective implementation of the proof of Proposition 3(i).

- In order for $\mathcal{R}(A)$ to be determined uniquely, we assume that $\mathcal{R}(1)$, the (abstract) rotation of the vertex 1, contains the subsequence $(2, 3, 4)$.
- Order the quintuples of the vertices of A lexicographically and denote them by $Q_1, Q_2, \dots, Q_{\binom{n}{5}}$.

- For every induced subgraph $B_k = A[Q_k]$, $k = 1, 2, \dots, \binom{n}{5}$, check if it is one of the five simply realizable 5-vertex AT-graphs (their drawings are shown in Fig. 2). If not, the algorithm terminates and answers “NO”, i.e., that A is not simply realizable. Otherwise we determine the rotation system $\mathcal{R}(B_k)$: of the two possible mutually inverse rotation systems, we choose the one which is compatible with the rotation systems $\mathcal{R}(B_1), \mathcal{R}(B_2), \dots, \mathcal{R}(B_{k-1})$. (By the choice of the ordering of the quintuples Q_i , there exists $k' < k$ such that $|Q_k \cap Q_{k'}| = 4$. If u, v, w, z are the vertices of the intersection $Q_k \cap Q_{k'}$, then $\mathcal{R}(B_k)$ determines the order of the elements v, w, z in the rotation of u in $B_{k'}$, which then determines $\mathcal{R}(B_{k'})$.)
- For each vertex $v \in V(A)$, compute the rotation $\mathcal{R}(v)$ from the rotation systems $\mathcal{R}(B_k)$, such that $v \in Q_k$: we choose a “reference vertex” $u \neq v$ and consider all subsequences of elements u, w, z ($w, z \in V(A) \setminus \{u, v\}$, $w \neq z$) in the rotations of v in the rotation systems $\mathcal{R}(B_k)$. These ordered triples determine a complete oriented graph $G_{u,v}$ on the set $V(A) \setminus \{u, v\}$. The rotation of v is then determined by the topological order of the vertices of $G_{u,v}$, which can be found in linear time. If $G_{u,v}$ has an oriented cycle, the algorithm terminates and answers “NO”.

Step 2: Partial Star-cut Representation

At this stage, we know that if A is simply realizable, then it has a simple realization with the computed rotation system $\mathcal{R}(A)$. But it may still happen that $\mathcal{R}(A)$ is not realizable as a rotation system of a simple complete topological graph. To decide this, we try to find an isomorphism class of some simple realization of A by constructing its star-cut representation.

By Proposition 3, we can determine the order of crossings of each edge with an arbitrary star $S(v)$, and also the rotation of all crossings on the edges of $S(v)$.

- Fix an arbitrary vertex $v \in V(A)$ and denote the other vertices of A by w_1, w_2, \dots, w_{n-1} , such that $\mathcal{R}(v) = (w_1, w_2, \dots, w_{n-1})$.
- Fix an orientation for each edge $w_i w_j$, $i < j$, by choosing w_i as an initial vertex.
- For every edge $e = w_i w_{i'}$ and every two edges $vw_j, vw_{j'}$ which cross e , determine the order $O_e(j, j')$ of crossings of e with vw_j and $vw_{j'}$ from the AT-graph $A[\{v, w_i, w_{i'}, w_j, w_{j'}\}]$.
- For every edge $e = w_i w_{i'}$, the orders $O_e(j, j')$ define a complete oriented graph on the $s_v(e)$ edges incident with v and crossing e . If this graph has an oriented cycle, terminate and answer “NO”, otherwise construct a topological order O_e of its vertices (i.e., the order in which e crosses the edges incident with v).
- For every crossing c_e^j of the edges $e = w_i w_{i'}$ and vw_j determine its rotation $\mathcal{R}(c_e^j)$, from the rotation system $\mathcal{R}(A[w_i, w_{i'}, w_j, v])$.

Now we are ready to start a construction of a star-cut representation of a possible simple realization of A , which would be obtained by cutting the sphere along the edges of the star $S(v)$.

- Draw a convex $2(n-1)$ -gon D and denote its boundary cycle as C . Denote the vertices of C counter-clockwise by $v_1, w_1, v_2, w_2, \dots, v_{n-1}, w_{n-1}$. For $i = 1, 2, \dots, n-1$, denote by f_{2i-1} the open edge $v_i w_i$, and by f_{2i} the open edge $w_i v_{i+1}$ (where $v_n = v_1$).

- To separate the endpoints of the edges not incident with v , cut a small disc around each vertex w_i , and draw a part of its boundary inside D as an arc g_i .
- Denote the edges of A not incident with v by $e_1, e_2, \dots, e_{\binom{n-1}{2}}$. For each edge e_i , define $s_v(e_i) + 1$ pseudochoords $e_{i,1}, e_{i,2}, \dots, e_{i,s_v(e_i)+1}$. We interpret $e_{i,j}$ as a portion of the edge e_i between the $(j - 1)$ th and the j th crossing of e_i with some edge incident with v . The 0th and the $(s_v(e_i) + 1)$ th crossing is the initial and the terminal end-point of e_i ; these end-points are now considered to lie on the arcs g_k . Considering $e_{i,j}$ oriented consistently with e_i , denote the initial vertex of $e_{i,j}$ by $a_{i,j}$ and the terminal vertex by $b_{i,j}$. Note that $a_{i,j+1}$ and $b_{i,j}$ correspond to the same crossing (the j th crossing of the edge e_i with some edge incident with v), which we denote by $c_{i,j}$.
- From the orders O_{e_i} and from the rotations of the crossings $c_{i,j}$ determine, for each $k = 1, 2, \dots, 2(n - 1)$, the set of end-points $a_{i,j}, b_{i,j}$ lying on the edge f_k .
- For each $k = 1, 2, \dots, n - 1$, construct a sequence O_{g_k} of the one-element sets $\{a_{i,1}\}, \{b_{i,s_v(e_i)+1}\}$ containing the end-points lying on g_k in clockwise order (determined by the rotation $\mathcal{R}(w_k)$).
- Construct a cyclic sequence O_C as a concatenation of the sequences $(f_1), O_{g_1}, (f_2, f_3), O_{g_2}, (f_4, f_5), \dots, O_{g_{n-1}}, (f_{2(n-1)})$. (The elements of O_C are the sets $\{a_{i,1}\}, \{b_{i,s_v(e_i)+1}\}$ and f_j .)
- For every pseudochoord $e_{i,j}$, construct its *type* $t(e_{i,j})$ which is defined as a pair (X, X') such that the sets X, X' are elements of O_C and $a_{i,j} \in X, b_{i,j} \in X'$. Note that if (X, X') is a type of some pseudochoord $e_{i,j}$, then $X \neq X'$.

The knowledge of the types $t(e_{i,j})$ now suffices to determine the realizability of the AT-graph A (in a polynomial time).

Step 3: Minimal Crossing Numbers and the Final Decision

We say that the types (X, X') and (Y, Y') are

- interlacing* if all the sets X, X', Y, Y' are distinct and if one of the cyclic sequences $(X, Y, X', Y'), (X, Y', X', Y)$ is a subsequence of O_C ,
- avoiding* if they are not interlacing and all the sets X, X', Y, Y' are distinct,
- parallel* if $(X, X') = (Y, Y')$ or $(X, X') = (Y', Y)$, and
- adjacent* otherwise, i.e., if exactly one of the following equalities holds: $X = Y, X = Y', X' = Y$ or $X' = Y'$.

See Fig. 4 for examples.

Clearly, if the types of two pseudochoords $e_{i,j}, e_{i',j'}$ are interlacing, then $e_{i,j}$ and $e_{i',j'}$ are forced to cross (if drawn inside D). Similarly, if the types $t(e_{i,j}), t(e_{i',j'})$ are avoiding, then the pseudochoords $e_{i,j}$ and $e_{i',j'}$ have no common crossing.

Two pseudochoords which are portions of the same edge e_i can have only interlacing or avoiding types. Hence, we can easily test the edge e_i for self-intersections.

- For all i, j, j' ($i \leq \binom{n-1}{2}$); $j, j' \leq s_v(e_i) + 1$), if the types $t(e_{i,j}), t(e_{i,j'})$ are interlacing, terminate and answer “NO”.

Two pseudochoords belonging to different edges can have parallel or adjacent types. In that case, their crossing status is not uniquely determined and depends on the order

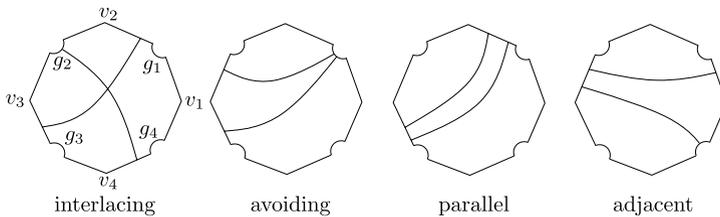


Fig. 4 Four categories of pairs of types of pseudochords



Fig. 5 A crossing and a non-crossing (i, i') -ladder (the fat lines represent distinct edges of the star $S(v)$)

of their end-points on the edge(s) f_k containing an end-point of both pseudochords. To obtain some information about crossings between these pseudochords, we group them into larger structures.

Let $e_i, e_{i'}$ be two fixed edges. A *positive (i, i') -ladder* is an inclusion-maximal sequence $((e_{i,j}, e_{i',j'}), (e_{i,j+1}, e_{i',j'+1}), \dots, (e_{i,j+k}, e_{i',j'+k}))$, such that $k \geq 1$ and for each $l \in \{0, 1, \dots, k - 1\}$ the two end-points $b_{i,j+l}$ and $b_{i',j'+l}$ ($a_{i,j+l+1}$ and $a_{i',j'+l+1}$) lie on a common edge f_p of C . It means that for each $l \in \{1, \dots, k - 1\}$, the edges $e_{i,j+l}$ and $e_{i',j'+l}$ have parallel types, and the edges $e_{i,j}$ and $e_{i',j'}$ have adjacent types, as well as the edges $e_{i,j+k}$ and $e_{i',j'+k}$. Similarly, we define a *negative (i, i') -ladder* as an inclusion-maximal sequence $((e_{i,j}, e_{i',j'}), (e_{i,j+1}, e_{i',j'-1}), \dots, (e_{i,j+k}, e_{i',j'-k}))$, such that $k \geq 1$ and for each $l \in \{0, 1, \dots, k - 1\}$ the two end-points $b_{i,j+l}$ and $a_{i',j'-l}$ ($a_{i,j+l+1}$ and $b_{i',j'-l-1}$) lie on a common edge f_p of C . Each (positive or negative) (i, i') -ladder corresponds to two maximal portions of the edges $e_i, e_{i'}$ which cross the same edges incident with v in the same order and from the same direction.

We call the (i, i') -ladder *crossing* if the two corresponding portions of edges are forced to cross, and *non-crossing* otherwise; see Fig. 5. We can determine whether the (i, i') -ladder is crossing or not from the types of its pairs of pseudochords (we show this only for positive ladders, the other case is similar).

Lemma 5 Let $L = ((e_{i,j}, e_{i',j'}), (e_{i,j+1}, e_{i',j'+1}), \dots, (e_{i,j+k}, e_{i',j'+k}))$ be a positive (i, i') -ladder, let $t(e_{i,j}) = (X, Z)$, $t(e_{i',j'}) = (Y, Z)$, $t(e_{i,j+k}) = (P, Q)$, and $t(e_{i',j'+k}) = (P, R)$. Define $t(L)$ as the number from $\{0, 1\}$ such that $t(L) = 0$ if and only if the sequences (X, Y, Z) and (P, Q, R) have the same orientation in O_C , i.e., either (X, Y, Z) and (P, Q, R) are both subsequences of O_C or both (X, Z, Y) and (P, R, Q) are subsequences of O_C . Then L is non-crossing if $k + t(L)$ is even, and crossing if $k + t(L)$ is odd.

Proof The proof is quite straightforward; the statement follows from the fact that for each $l \in \{0, 1, \dots, k - 1\}$ the order of the end-points $b_{i,j+l}, b_{i',j'+l}$ on the common

edge f_k of the cycle C is opposite to the order of the end-points $a_{i,j+l+1}, a_{i',j'+l+1}$ on the edge f_{k+o} ($o \in \{-1, 1\}$) adjacent to f_k and representing the same edge of the star $S(v)$. □

Clearly, every pair $(e_{i,j}, e_{i',j'})$ of pseudochords with adjacent or parallel types belongs to exactly one (i, i') -ladder. It follows that the set $P_{i,i'} = \{(e_{i,j}, e_{i',j'}) ; 1 \leq j \leq s_v(e_i) + 1, 1 \leq j' \leq s_v(e_{i'}) + 1\}$ can be uniquely partitioned into (i, i') -ladders and one-element sets consisting of pairs of pseudochords with interlacing or avoiding types. For each set T from this partition, we are able to determine the parity of the total number of crossings among the pseudochords from T . Hence, we are able to determine the parity of the total number of crossings between the edges e_i and $e_{i'}$, and also a lower bound for this total number of crossings.

We are now ready to describe the last steps of the recognition algorithm.

- For every two edges $e_i, e_{i'}$ ($1 \leq i < i' \leq \binom{n-1}{2}$) do the following:
 - Determine the partition of $P_{i,i'}$ into (i, i') -ladders and pairs with interlacing or avoiding types. For each (i, i') -ladder from this partition, determine whether it is crossing or non-crossing.
 - Compute $CR(e_i, e_{i'})$, the sum of the number of crossing (i, i') -ladders and the number of pairs of pseudochords from $P_{i,i'}$ with interlacing types.
 - Define $CR_A(e_i, e_{i'}) \in \{0, 1\}$ so that $CR_A(e_i, e_{i'}) = 0$ if the edges $e_i, e_{i'}$ form a non-crossing pair in the abstract graph A and $CR_A(e_i, e_{i'}) = 1$ if the edges $e_i, e_{i'}$ form a crossing pair in A .
- If there exist edges $e_i, e_{i'}$ such that $CR(e_i, e_{i'}) \neq CR_A(e_i, e_{i'})$, terminate and answer “NO”. Otherwise answer “YES”.

Clearly, if the algorithm answers “NO”, the abstract graph A is not realizable. It remains to prove that if for every two edges $e_i, e_{i'}$ the equality $CR(e_i, e_{i'}) = CR_A(e_i, e_{i'})$ holds, then there exists a choice of the counter-clockwise orders O_{f_k} of the end-points of the pseudochords on the edges f_k , such that the induced number of crossings between any two edges $e_i, e_{i'}$ attains the lower bound $CR(e_i, e_{i'})$. The orders O_{f_k} , together with the orders O_{w_k} , determine a counter-clockwise (perimetric) order PO_C of all the end-points $a_{i,j}, b_{i,j}$ on the cycle C . For each pair of pseudochords, PO_C determines whether they cross or not. Note that for every given perimetric order PO_C the arrangement of the pseudochords can be realized: for example, the pseudochords can be drawn as straight-line segments (i.e., as actual chords of the polygon D).

For every $k = 1, 2, \dots, (n - 1)$, the edges f_{2k-1} and f_{2k} represent the same edge, vw_k , of the graph A . Thus, the order $O_{f_{2k}}$ is an *almost-inverse* of $O_{f_{2k-1}}$, i.e., $O_{f_{2k}}$ is the inverse of the order, which we obtain from $O_{f_{2k-1}}$ by replacing each end-point $a_{i,j}$ ($b_{i,j}$) with the end-point $b_{i,j-1}$ ($a_{i,j+1}$) corresponding to the same crossing on the edge vw_k . Hence, PO_C is now uniquely determined by the orders $O_{f_2}, O_{f_4}, \dots, O_{f_{2(n-1)}}$.

Claim 6 *Let $O_{f_2}, O_{f_4}, \dots, O_{f_{2(n-1)}}$ be the orders which minimize the total number of crossings between pseudochords induced by PO_C . Then for every two edges $e_i, e_{i'}$, the order PO_C induces exactly $CR(e_i, e_{i'})$ crossings together on all the pairs of pseudochords from $P_{i,i'}$.*

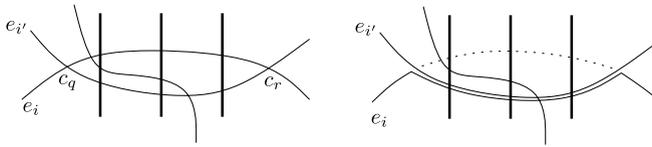


Fig. 6 An empty lens allows us to decrease the number of crossings by 2

Proof Suppose that it is not the case. Then for some two edges $e_i, e_{i'}$, there exists an (i, i') -ladder L with at least two crossings induced by PO_C . Suppose, without loss of generality, that L is a positive ladder $((e_{i,j}, e_{i',j}), (e_{i,j+1}, e_{i',j'+1}), \dots, (e_{i,j+k}, e_{i',j'+k}))$. Let $q < r$ be the least two integers such that PO_C induces a crossing c_q between $e_{i,j+q}$ and $e_{i',j'+q}$, and a crossing c_r between $e_{i,j+r}$ and $e_{i',j'+r}$. In the topological graph G represented by this pseudochord arrangement, the two portions e'_i, e'_i of the edges $e_i, e_{i'}$ between the crossings c_q and c_r form an *empty lens* $L_{q,r}$, i.e., a region bounded by the curves e'_i, e'_i , which contains no vertex of G . Hence, the total number of crossings of every other edge of G with the curves e'_i and e'_i is even. Assume that the lens $L_{q,r}$ is inclusion-minimal (over all pairs of edges $e_i, e_{i'}$). Then every connected component of every edge intersecting $L_{q,r}$ has one end-point on e'_i and the other end-point on e'_i . Hence, every edge of G has the same number of crossings with e'_i as with e'_i . It follows that by redrawing e'_i along the curve e'_i , we decrease the total number of crossings in G by two (we get rid of the crossings c_q and c_r) and we do not change the type of any pseudochord in the corresponding star-cut representation of G ; see Fig. 6. The redrawing of the curve e'_i corresponds to the translations of the end-points $b_{i,j+q}, b_{i,j+q+1}, \dots, b_{i,j+r-1}$ ($a_{i,j+q+1}, a_{i,j+q+2}, \dots, a_{i,j+r}$) next to the end-points $b_{i',j'+q}, b_{i',j'+q+1}, \dots, b_{i',j'+r-1}$ ($a_{i',j'+q+1}, a_{i',j'+q+2}, \dots, a_{i',j'+r}$) in the corresponding orders O_{f_k} (the translated end-point is moved “just behind” the other end-point). We have constructed a perimetric order PO'_C which induces less crossings than PO_C , a contradiction. \square

Corollary 7 *If the algorithm answers “YES”, then the abstract graph A is realizable.*

Step 4: Constructing the Orders of Xrossings on $S(v)$

The proof of Claim 6 also gives an idea how to construct the perimetric order PO_C algorithmically:

- Choose an arbitrary set of orders $O_{f_2}, O_{f_4}, \dots, O_{f_{2(n-1)}}$ and compute the related orders $O_{f_1}, O_{f_3}, \dots, O_{f_{2n-3}}$.
- While there exists some (i, i') -ladder with at least two induced crossings, find an inclusion-minimal lens $L_{q,r}$ and change the orders of the corresponding end-points in the corresponding orders O_{f_k} , as in the proof of Claim 6.
- Return the resulting perimetric order PO_C .

Step 5: Constructing the Orders of Crossings on All Other Edges

The algorithm can be extended so that it finds some isomorphism class of the arrangement with the perimetric order PO_C . It remains to find one possible set of orders of crossings of the pseudochords with the other pseudochords.

Some difficulties with the computation of the orders may occur if the pseudochords were drawn as straight-line segments, because we could obtain pairs of crossings very close to each other (closer than the precision of our representation of real numbers), and they would become indistinguishable for the algorithm. So we choose a different, more topological approach and compute the orders recursively:

- Choose an arbitrary pseudochord p and from the perimetric order PO_C identify the set $\{p_1, p_2, \dots, p_k\}$ of all pseudochords that cross p .
- Cut the circle C into two arcs, C_1 and C_2 , by the end-points of p and define two circles $C'_1 = C_1 \cup p$ and $C'_2 = C_2 \cup p$. Partition the perimetric order PO_C into two orders O_{C_1} and O_{C_2} of the end-points on the arcs C_1 and C_2 .
- Cut each pseudochord $p_i, i = 1, 2, \dots, k$, into two portions with one end-point on p and the second end-point on C . Define two mutually almost-inverse orders O_p^1 and O_p^2 of these new end-points on p such that the portions of the pseudochords p_i between p and C_1 do not cross (O_p^1 is a counter-clockwise order with respect to the circle C'_1 and it can be deduced from O_{C_1}).
- Define $PO_{C'_1}$ as a concatenation of O_{C_1} and O_p^1 , and $PO_{C'_2}$ as a concatenation of O_{C_2} and O_p^2 .
- Recursively compute the orders of crossings on the pseudochords in the two arrangements with the perimetric orders $PO_{C'_1}$, $PO_{C'_2}$ and merge the computed orders for the portions of pseudochords p_i together.

It is quite straightforward to verify that each step of the algorithm can be performed in polynomial time. Using a bounded number of quantifications over subsets (of vertices, edges, etc.) of bounded size, each step can be decomposed into a polynomial number of elementary tasks; either those solvable in constant time, or simple subroutines such as searching in a polynomial list or topological sorting of a partially ordered set. More concrete estimates on running time would require to describe the particular implementation and data structures in much more detail, and it would only increase the technical complexity of the paper.

Acknowledgements I am grateful to Pavel Valtr for introducing me to the topic of topological graphs, for many valuable comments and suggestions and for the help with the preparation of this paper. I am also grateful to Jan Kratochvíl for helpful discussions about abstract topological graphs and NP-hardness and for presenting me his results. I would also like to thank the referees for their thorough reviews, for providing additional references, and for many fruitful suggestions that helped improve the presentation of the paper.

References

1. Aichholzer, O., Aurenhammer, F., Krasser, H.: On the crossing number of complete graphs. *Computing* **76**(1–2), 165–176 (2006)

2. Canny, J.: Some algebraic and geometric computations in PSPACE. In: STOC '88: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, pp. 460–469. ACM, New York (1988)
3. Cohen, M., Lustig, M.: Paths of geodesics and geometric intersection numbers I. In: Combinatorial Group Theory and Topology, Alta, Utah, 1984, pp. 479–500. Annals of Mathematics Studies, vol. 111. Princeton University Press, Princeton (1987)
4. Dehn, M.: Über Kurvensysteme auf Zweiseitigen Flächen mit Anwendung auf das Abbildungsproblem. In: Autogr. Vortrag in Math. Kolloquium, Breslau, 11 February 1922
5. Farb, B., Thurston, B.: Homeomorphisms and simple closed curves. Unpublished manuscript
6. Gioan, E.: Complete graph drawings up to triangle mutations. In: Graph-Theoretic Concepts in Computer Science. Lecture Notes in Computer Science, vol. 3787, pp. 139–150. Springer, Berlin (2005)
7. Gassner, E., Jünger, M., Percan, M., Schaefer, M., Schulz, M.: Simultaneous graph embeddings with fixed edges. In: Graph-Theoretic Concepts in Computer Science. Lecture Notes in Computer Science, vol. 4271, pp. 325–335. Springer, Berlin (2006)
8. Goodman, J.E.: Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Math.* **32**, 27–35 (1980)
9. Guy, R.K.: A combinatorial problem. *Nabla (Bull. Malay. Math. Soc.)* **7**, 68–72 (1960)
10. Hamidi-Tehrani, H.: Algorithms in the mapping class groups. PhD thesis, Columbia (1997)
11. Harborth, H., Mengersen, I.: Drawings of the complete graph with maximum number of crossings. In: Proceedings of the Twenty-third Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Boca Raton, FL, 1992. Congressus Numerantium, vol. 88, pp. 225–228 (1992)
12. Hass, J., Scott, P.: Intersections of curves on surfaces. *Isr. J. Math.* **51**(1–2), 90–120 (1985)
13. Kratochvíl, J.: String graphs. II. Recognizing string graphs is NP-hard. *J. Comb. Theory, Ser. B* **52**, 67–78 (1991)
14. Kratochvíl, J.: A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Appl. Math.* **52**, 233–252 (1994)
15. Kratochvíl, J., Lubiw, A., Nešetřil, J.: Noncrossing subgraphs in topological layouts. *SIAM J. Discrete Math.* **4**(2), 223–244 (1991)
16. Kratochvíl, J., Matoušek, J.: NP-hardness results for intersection graphs. *Comment. Math. Univ. Carol.* **30**, 761–773 (1989)
17. Kratochvíl, J., Matoušek, J.: String graphs requiring exponential representations. *J. Comb. Theory, Ser. B* **53**, 1–4 (1991)
18. Kratochvíl, J., Matoušek, J.: Intersection graphs of segments. *J. Comb. Theory, Ser. B* **62**(2), 289–315 (1994)
19. Kynčl, J.: The complexity of several realizability problems for abstract topological graphs. In: Graph Drawing 2007. Lecture Notes in Computer Science, vol. 4875, 137–158. Springer, Berlin (2008)
20. Kynčl, J.: Enumeration of simple complete topological graphs. *Eur. J. Comb.* **30**(7), 1676–1685 (2009)
21. Kynčl, J., Schaefer, M., Sedgwick, E., Štefankovič, D.: Spiralling and folding: the topological view. In preparation
22. Kynčl, J., Valtr, P.: On edges crossing few other edges in simple topological complete graphs. *Discrete Math.* **309**, 1917–1923 (2009)
23. Lustig, M.: Paths of geodesics and geometric intersection numbers II. In: Combinatorial Group Theory and Topology, Alta, Utah, 1984. Annals of Mathematics Studies, vol. 111, pp. 501–543. Princeton University Press, Princeton (1987)
24. Mněv, N.E.: The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In: Topology and Geometry—Rohlin Seminar. Lecture Notes in Mathematics, vol. 1346, pp. 527–543. Springer, Berlin (1988)
25. Pach, J., Solymosi, J., Tóth, G.: Unavoidable configurations in complete topological graphs. *Discrete Comput. Geom.* **30**, 311–320 (2003)
26. Pach, J., Tóth, G.: Recognizing string graphs is decidable. In: Discrete and Computational Geometry and Graph Drawing, Columbia, SC, 2001. Discrete and Computational Geometry, vol. 28(4), 593–606 (2002)
27. Pach, J., Tóth, G.: How many ways can one draw a graph? *Combinatorica* **26**(5), 559–576 (2006)
28. Reinhart, B.: Algorithms for Jordan curves on compact surfaces. *Ann. Math.* **75**, 209–222 (1962)
29. Richter, R.B., Thomassen, C.: Relations between crossing numbers of complete and complete bipartite graphs. *Am. Math. Mon.* **104**(2), 131–137 (1997)
30. Schaefer, M.: Complexity of some geometric and topological problems. In: Graph Drawing 2009. Lecture Notes in Computer Science, vol. 5849, pp. 334–344. Springer, Berlin (2010)

31. Schaefer, M., Sedgwick, E., Štefankovič, D.: Recognizing string graphs in NP. *J. Comput. Syst. Sci.* **67**, 365–380 (2003)
32. Schaefer, M., Štefankovič, D.: Decidability of string graphs. *J. Comput. Syst. Sci.* **68**, 319–334 (2004)
33. Shahrokhi, F., Székely, L.A., Vrfo, I.: Crossing numbers of graphs, lower bound techniques and algorithms: a survey. In: *Proc. DIMACS Workshop on Graph Drawing'94. Lecture Notes in Computer Science*, vol. 894, pp. 131–142. Springer, Berlin (1995)
34. Shor, P.W.: Stretchability of pseudolines is NP-hard. In: *Applied Geometry and Discrete Mathematics. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 4, pp. 531–554. Am. Math. Soc., Providence (1991)
35. Wagner, U.: On the rectilinear crossing number of complete graphs. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, 2003, pp. 583–588. ACM, New York (2003)