

# Recognizing String Graphs in NP

Marcus Schaefer\*

Eric Sedgwick†

Daniel Štefankovič‡

## ABSTRACT

A *string graph* is the intersection graph of a set of curves in the plane. Each curve is represented by a vertex, and an edge between two vertices means that the corresponding curves intersect. We show that string graphs can be recognized in **NP**. The recognition problem was not known to be decidable until very recently, when two independent papers established exponential upper bounds on the number of intersections needed to realize a string graph [18, 20]. These results implied that the recognition problem lies in **NEXP**. In the present paper we improve this by showing that the recognition problem for string graphs is in **NP**, and therefore **NP**-complete, since Kratochvíl [12] showed that the recognition problem is **NP**-hard. The result has consequences for the computational complexity of problems in graph drawing, and topological inference.

## 1. STRINGS, DRAWINGS, AND DIAGRAMS

A *string graph* is the intersection graph of a set of curves in the plane. A (*Jordan*) *curve*, or *string*, is a set homeomorphic to  $[0, 1]$ . Given a collection of curves  $(C_i)_{i \in I}$  in the plane, the corresponding intersection graph is  $(I, \{\{i, j\} : C_i \text{ and } C_j \text{ intersect}\})$ . The *size* of a collection of curves is the number of intersection points (we assume that no three curves intersect in the same point). A graph isomorphic to the intersection graph of a collection of curves in the plane is called a *string graph*.

The string graph problem asks how string graphs can be recognized. The problem made its first explicit appearance

\*mschaefer@cs.depaul.edu, Department of Computer Science, DePaul University, 243 South Wabash, Chicago, Illinois 60604, USA.

†esedgwick@cs.depaul.edu, Department of Computer Science, DePaul University, 243 South Wabash, Chicago, Illinois 60604, USA.

‡stefanko@cs.uchicago.edu, Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, Illinois 60637, USA, and Department of Computer Science, Comenius University, Bratislava, Slovakia.

in a 1966 paper by Sinden on circuit layout [21], although a similar question had been suggested earlier by Benzer on genetic structures [1]. The string graph problem was introduced to the combinatorial community by Ron Graham in 1976 [9].

From a combinatorial point of view we are interested in  $c_s(G)$ , the smallest number of intersections of a set of curves realizing a string graph. For graphs  $G$  that are not string graphs, we let  $c_s(G)$  be infinity. With that we can define  $c_s(n) = \max\{c_s(G) : G \text{ is a string graph on } n \text{ vertices}\}$ . A computable upper bound on  $c_s(n)$  implies decidability of the string graph problem. In 1991 Kratochvíl and Matoušek [14] showed rather surprisingly, that  $c_s(n) \geq 2^{cn}$  for some constant  $c$ , and conjectured that  $c_s(n) \leq 2^{cn^k}$  for some  $k$ . The papers by Pach and Tóth [18], and Schaefer and Štefankovič [20] established upper bounds of this form, implying decidability of the string graph problem in nondeterministic exponential time.

The string graph problem is closely related to a graph drawing problem, a connection we will make use of later. Given a graph  $G = (V, E)$  and a set  $R \subseteq \binom{E}{2} = \{\{e, f\} : e, f \in E\}$  on  $E$ , we call a drawing  $D$  of  $G$  in the plane a *weak realization* of  $(G, R)$  if only pairs of edges which are in  $R$  are allowed to intersect in  $D$  (they do not have to intersect, however). In this case we call  $(G, R)$  *weakly realizable*. We say that  $D$  is a *realization* of  $G$  if exactly the pairs of edges in  $R$  intersect in  $D$ .<sup>1</sup> Let us define  $c_w(G, R)$  as the smallest number of intersections in a weak realization of  $(G, R)$ ,  $c_w(G) = \max\{c_w(G, R) : (G, R) \text{ has a weak realization}\}$ , and  $c_w(m) = \max\{c_w(G) : G \text{ has } m \text{ edges}\}$ .

The string graph problem can be reduced in polynomial time to the weak realizability problem [16, 12]. The reduction is as follows. Given a graph  $G = (V, E)$ , let  $G' = (V \cup E, \{\{u, e\} : u \in e \in E\})$ , and  $R = \{\{\{u, e\}, \{v, f\}\} : \{u, v\} \in E\}$ . Then  $G$  is a string graph if and only if  $(G', R)$  is weakly realizable.

In Theorem 4.4 we show that the weak realizability problem is in **NP**. Because of the reduction of the string graph problem to the weak realizability problem, and Kratochvíl's proof of **NP**-hardness of the string graph problem [12] this implies the following corollaries.

**COROLLARY 1.1.** *The string graph problem is complete for NP.*

**COROLLARY 1.2.** *The weak realizability problem is complete for NP.*

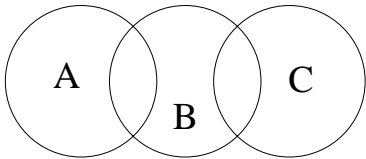
<sup>1</sup>Kratochvíl [13, 11, 12] calls  $(G, R)$  an *abstract topological graph*, and uses the word *feasible* for weakly realizable.

The corollaries imply that the weak realizability problem can be reduced to the string graph problem in polynomial time. No natural polynomial time reduction witnessing this relationship is known (although there is an **NP**-reduction).

The weak realizability problem is a generalization of the concept of *crossing number* of a graph  $G$ , which is the smallest number of intersections necessary to draw  $G$  in the plane. Garey and Johnson showed that computing the crossing number is **NP**-complete [7]. Many variants of this problem have been considered in the literature, including the *pairwise crossing number* (or *crossing pairs number*), which is the smallest number of pairs of edges that need to intersect to draw  $G$ . Pach and Tóth recently showed that computing the pairwise crossing number is **NP**-hard [17]. Since there is an **NP**-reduction from this problem to the weak realizability problem, we have the following corollary.

**COROLLARY 1.3.** *The pairwise crossing number problem is complete for **NP**.*

The string graph problem is also related to Euler (or Venn) diagrams, and through these to topological inference. Given a specification of the relationships of concepts, such as “some  $A$  is  $B$ , some  $B$  is  $C$ , but no  $A$  is  $C$ ”, we can ask whether there is a diagram illustrating the relationship of the concepts (regions homeomorphic to the unit disk). In this particular case Figure 1 illustrates the given situation.



**Figure 1:** Some  $A$  is  $B$ , some  $B$  is  $C$ , but no  $A$  is  $C$ .

This problem is polynomial-time equivalent to the string graph problem. Topological inference allows a more refined set of predicates to describe relationship between regions, but even in this case a reduction to the string graph problem can be established, giving us the following result.

**COROLLARY 1.4.** *The existential theory of diagrams and the existential fragment of topological inference are complete for **NP**.*

Details of this reduction (which is an **NP**-reduction rather than a polynomial time one) and the definitions involved can be found in the journal version of [20]. Several restricted versions of this problem were shown to be solvable in **P** and **NP** earlier, but the general problem was not known to be decidable [10, 2, 22].

For the proof of our main theorem, Theorem 4.4, the same approach as in our earlier paper [20] proves successful: we reinterpret the problem as a problem over words. The necessary background material on words and word equations is covered in Section 2. The topological aspects of the proof are covered in Section 3.

## 2. WORD EQUATIONS

Let  $\Sigma$  be an alphabet of symbols and  $\Theta$  be an alphabet of variables. The alphabets  $\Sigma$  and  $\Theta$  are disjoint. A *word*

*equation*  $u = v$  is a pair of words  $(u, v) \in (\Sigma \cup \Theta)^* \times (\Sigma \cup \Theta)^*$ . The *size of the equation*  $u = v$  is  $|u| + |v|$ . A *solution of the word equation*  $u = v$  is a morphism  $h : (\Sigma \cup \Theta)^* \rightarrow \Sigma^*$  such that  $h(a) = a$  for all  $a \in \Sigma$  and  $h(u) = h(v)$  ( $h$  being a morphism means that  $h(wz) = h(w)h(z)$  for any  $w, z \in (\Sigma \cup \Theta)^*$ ). The *length of the solution*  $h$  is  $\sum_{x \in \Theta} |h(x)|$ .

A *word equation with specified lengths* is a word equation  $u = v$  and a function  $f : \Theta \rightarrow \mathbb{N}$ . The solution  $h$  has to respect the lengths, i.e. we require  $|h(x)| = f(x)$  for all  $x \in \Theta$ .

Let  $w$  be a word in  $\Sigma^*$ . We can write  $w = c_1 f_1 c_2 \dots c_k f_k$  where the  $c_i$  are characters in  $\Sigma$ , and the  $f_i$  are subwords of  $w$ . More precisely,  $c_1 = w[1]$  and  $f_i$  is the longest prefix of  $f_i c_{i+1} \dots f_k$  which occurs in  $c_1 f_1 \dots f_{i-1} c_i$ . The *Lempel-Ziv (LZ) encoding* of  $w$  is  $LZ(w) = c_1[a_1, b_1]c_2 \dots c_k[a_k, b_k]$  where  $f_i = w[a_i \dots b_i]$ . The *size of the encoding* is  $|LZ(w)| = k(\log |w| + \log |\Sigma| + 1)$ . Note that some words can be compressed exponentially.

Let  $h : (\Sigma \cup \Theta)^* \rightarrow \Sigma^*$  be a solution of an equation  $u = v$ . The LZ encoding of  $h$  is the sequence of LZ encodings of  $h(x)$  for all  $x \in \Theta$ . The size of the encoding is  $|LZ(h)| = \sum_{x \in \Theta} |LZ(h(x))|$ . The usefulness of LZ encoding for word equations is demonstrated by following two results.

**THEOREM 2.1** ([8]). *Let  $u = v$  be a word equation. For an LZ encoding of a morphism  $h$  we can check whether  $h$  is a solution of the equation in time polynomial in  $|LZ(h)|$ .*

**THEOREM 2.2** ([19]). *Let  $u = v$  be a word equation with lengths specified by a function  $f$ . Assume that  $u = v$  has a solution respecting the lengths given by  $f$ . Then there is a solution  $h$  respecting the lengths such that  $|LZ(h)|$  is polynomial in the size of the binary encoding of  $f$  and the size of the equation. Moreover, the lexicographically least such solution can be found in polynomial time.*

Given an equation with specified lengths there might be solutions which can not be LZ compressed. However Theorem 2.2 says that there is a solution which can be LZ compressed. In particular if the equation has a unique solution then that solution can be LZ compressed. Note that it is easy to encode several equations into one equation [15, Proposition 12.1.8], hence Theorems 2.1 and 2.2 hold for systems of equations as well.

We will need the following two results which easily follow from [8].

**LEMMA 2.3.** *For an LZ-encoding  $LZ(w)$  of  $w$  we can test whether  $w$  is a palindrome in time polynomial in  $|LZ(w)|$ .*

**LEMMA 2.4.** *Given an LZ encoding  $LZ(w)$  of  $w$  and  $a \in \Sigma$ , we can compute the number of occurrences of  $a$  in  $w$  in time polynomial in  $|LZ(w)|$ .*

## 3. COMPUTATIONAL TOPOLOGY

In the following let  $M$  be a compact orientable surface with boundary. A simple arc  $\gamma$  such that both its endpoints  $\gamma(0), \gamma(1)$  are on the boundary  $\partial M$  and the internal points  $\gamma(x), 0 < x < 1$  are in the interior  $\overset{\circ}{M}$  is called a *properly embedded arc*. Two properly embedded arcs  $\gamma_1, \gamma_2$  are *isotopic rel. boundary* ( $\gamma_1 \sim \gamma_2$ ) if there is a continuous deformation of  $\gamma_1$  to  $\gamma_2$  which does not move the endpoints. The *isotopy class* of  $\gamma$  is the set of properly embedded arcs isotopic to

$\gamma$ . Given two properly embedded arcs  $\gamma_1, \gamma_2$  the *intersection number* of  $\gamma_1$  and  $\gamma_2$  is

$$i(\gamma_1, \gamma_2) = \min_{c_i \sim \gamma_i} |c_1 \cap c_2|.$$

We say that two properly embedded arcs  $\gamma_1, \gamma_2$  are *isotopically disjoint* if  $i(\gamma_1, \gamma_2) = 0$ .

Let  $\gamma_1, \gamma_2$  be two properly embedded arcs. A *bigon*  $B$  bounded by  $\gamma_1, \gamma_2$  is a disc which has exactly two intersections (of  $\gamma_1$  and  $\gamma_2$ ) on the boundary  $\partial B$ .

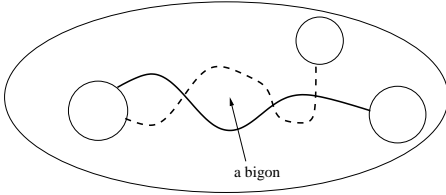


Figure 2: Example of a bigon.

We will need the following standard result (see e.g. [6]).

LEMMA 3.1. *If two properly embedded arcs  $\gamma_1, \gamma_2$  intersect more than  $i(\gamma_1, \gamma_2)$  times then they bound a bigon.*

LEMMA 3.2. *Let  $\gamma_1, \dots, \gamma_n$  be properly embedded arcs. Then there are  $c_1, \dots, c_n$  in general position such that  $c_i \sim \gamma_i$ ,  $i \in [n]$  and  $|c_i \cap c_j| = i(\gamma_i, \gamma_j)$ ,  $i, j \in [n]$ .*

PROOF. Let  $c_1, \dots, c_n$  be in general position such that  $\sum |c_i \cap c_j|$  is minimized. If there are two properly embedded arcs  $c_i, c_j$  which intersect more than  $i(\gamma_i, \gamma_j)$  times then they bound a bigon  $B$ . Let  $e, f$  be the sides of the bigon. W.l.o.g. assume that  $B$  is the smallest bigon w.r.t. containment. Then any properly embedded arc  $c$  which crosses  $e$  also crosses  $f$ , otherwise  $B$  would not be smallest. However then we can isotope (rel. boundary) the properly embedded arc  $c_i$  to decrease the total number of intersections, a contradiction. See Figure 3. ■

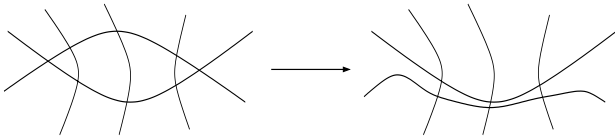


Figure 3: Decreasing the number of intersections.

Let  $T$  be a triangulation of the compact surface  $M$ . Let  $\gamma$  be a properly embedded arc. We say that  $\gamma$  is *normal* w.r.t.  $T$  if all the intersections with  $T$  are transversal and if  $\gamma$  enters a triangle  $t \in T$  via edge  $e$  then it leaves  $t$  via an edge different from  $e$ .

LEMMA 3.3. *Let  $T$  be triangulation of a surface  $M$ . Let  $\gamma$  be a properly embedded arc. Then there is  $c \sim \gamma$  which is normal w.r.t.  $T$ .*

PROOF. Let  $c \sim \gamma$  which minimizes the number of intersections with  $T$ . If  $c$  enters and leaves  $t \in T$  through the same edge  $e$  then the number of intersections of  $c$  with  $T$  can be reduced, a contradiction. ■

Given a properly embedded arc  $\gamma$  which is normal w.r.t.  $T$  we can label each edge of the triangulation with the number of intersections of  $\gamma$  with the edge. In each triangle the labels of its edges determine the behavior of the curve inside the triangle up to isotopy (for this we need that  $\gamma$  is normal w.r.t.  $T$  and that  $\gamma$  is not self-intersecting). Hence the numbering on  $T$  determines the isotopy class of  $\gamma$ . The size of the representation is the total bitlength of the labels. Each isotopy class may have many different representations.

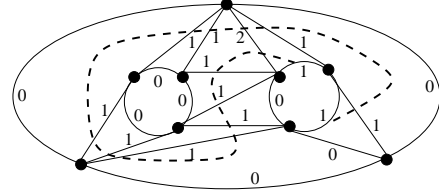


Figure 4: Example of a numbering.

Call a numbering  $\ell : E_T \rightarrow \mathbb{N}$  of  $T$  *valid* if there is a properly embedded arc  $\gamma$  in general position w.r.t.  $T$  which intersects each  $e \in T$ ,  $\ell(e)$  times.

Let  $\ell$  be a valid numbering. The sum of the labels of edges from  $E_T \cap \partial M$  is 2. For each triangle  $t \in T$  the labels  $a, b, c$  of edges of  $t$  satisfy  $a + b \geq c$ ,  $a + c \geq b$ ,  $b + c \geq a$  and  $a + b + c$  is even. These conditions are necessary for validity, but not sufficient. Call a labeling satisfying the conditions *semi-valid*. Any semi-valid labeling defines a properly embedded arc and a (possibly empty) set of closed curves.

We associate the following system of word equations with the triangulation  $T$ . The system will emulate behavior of a set of labeled curves on  $M$ . We assume that the curves do not intersect. For each oriented edge  $(u, v) \in T$  there is a variable  $x_{u,v}$  encoding the order in which the curves intersect on  $(u, v)$ . Let  $t \in T$  be a triangle with vertices  $u, v, w$ . We add six variables  $y_{t,u}, y_{t,v}, y_{t,w}, y_{u,t}, y_{v,t}, y_{w,t}$ . See Figure 5. For example, the variable  $y_{t,u}$  encodes the segments of curves between the oriented edges  $(w, u)$  and  $(v, u)$ . We have the following equations  $x_{u,v} = y_{u,t}y_{t,v}$ ,  $x_{v,u} = y_{v,t}y_{t,u}$ ,  $x_{v,w} = y_{v,t}y_{t,w}$ ,  $x_{w,v} = y_{w,t}y_{t,v}$ ,  $x_{u,w} = y_{u,t}y_{t,w}$ ,  $x_{w,u} = y_{w,t}y_{t,u}$ . Note that if we know the lengths of the  $x$  variables, then we can calculate the lengths of the other variables, for example  $|y_{u,t}| = (|x_{u,v}| + |x_{u,w}| - |x_{v,w}|)/2$ .

LEMMA 3.4. *Given a numbering  $\ell : E_T \rightarrow \mathbb{N}$  we can test whether  $\ell$  is valid in polynomial time.*

PROOF. We first verify that  $\ell$  is semi-valid, and reject  $\ell$  if it is not.

Let  $\Sigma = \{a, b\}$ . Take the set of equations associated with  $T$  over  $\Sigma$ . For each  $e = (u, v) \in E_T$  specify  $|x_{u,v}| = \ell(e)$ . For each edge  $e = (u, v) \in E_T \cap \partial M$  we specify  $x_{u,v} = b^{\ell(e)}$ .

We claim that if  $\ell$  is valid then the system of equations has a unique solution. Take the properly embedded arc  $\gamma$  which realizes  $\ell$ . Number the intersections of  $\gamma$  with  $T$  in the order in which they occur on  $\gamma$ . Each intersection corresponds to

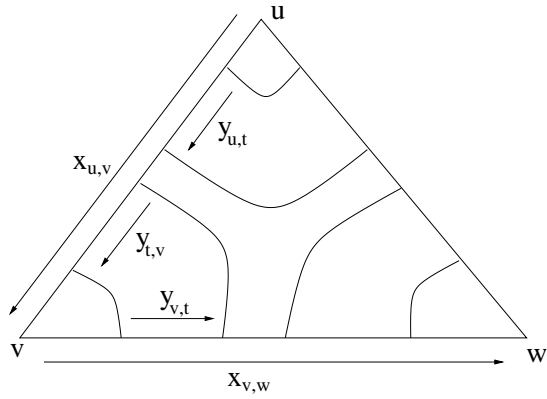


Figure 5: The variables for a triangle  $t$ .

a position in some variable. By induction on the number of intersections it follows that each position in every variable is forced to be  $b$ .

On the other hand if  $\ell$  is not valid then, because it is semi-valid, there is a solution to the set of word equations. However, a lexicographically smallest solution will now contain  $as$  (corresponding to the closed curves, disconnected from the endpoints  $b$ ). Because of Theorem 2.2 we can compute the lexicographically least solution in polynomial time, and—using Lemma 2.4—check that it does not contain any occurrences of  $a$ .

Thus by solving the system we can check whether  $\ell$  is valid. ■

LEMMA 3.5. *Let  $\gamma_1, \gamma_2$  be properly embedded arcs given by valid numberings  $\ell_1, \ell_2$ . If  $\gamma_1, \gamma_2$  do not intersect then we can verify that  $i(\gamma_1, \gamma_2) = 0$  in polynomial time.*

PROOF. Let  $\Sigma = \{a, b\}$ . Take the set of equations associated with  $T$  over  $\Sigma$ . For each  $e = (u, v) \in E_T$  specify  $|x_{u,v}| = \ell_1(e) + \ell_2(e)$ . For each edge  $e = (u, v) \in E_T \cap \partial M$  we specify  $x_{u,v} = w$  where  $w$  represents the order in which  $\gamma_1$  and  $\gamma_2$  occur on  $e$ .

Let  $h$  be a solution of the system. Assume that the number of occurrences of  $a$  in  $h(x_{u,v})$  is  $\ell_1(u, v)$  and  $x_{u,v} = x_{v,u}^R$  for all  $(u, v) \in E_T$ . Then  $\gamma_1, \gamma_2$  are isotopically disjoint.

If  $\gamma_1, \gamma_2$  are disjoint then the system has a unique solution  $h$ . The proof is analogous to the argument in the proof of Lemma 3.4.

Since the solution  $h$  is unique we can find  $LZ(h)$  in polynomial time by Theorem 2.2. For each  $e = (u, v) \in E$  we verify that the number of occurrences of  $a$  in  $h(x_{u,v})$  is  $\ell_1(e)$ , and  $x_{u,v} = x_{v,u}^R$  using Lemma 2.3, and Lemma 2.4. ■

## 4. WEAK REALIZABILITY

We need the following bound on the number of intersections in a drawing of a weak realization with the smallest number of intersections.

THEOREM 4.1 ([20]). *Let  $G$  be a graph with  $m$  edges,  $R \subseteq \binom{E}{2}$  such that  $(G, R)$  is weakly realizable, and let  $D$  be a weak realization of  $(G, R)$  with the minimal number of intersections. Then for any edge  $e \in G$  there are less than  $2^m$  intersections on the curve realizing  $e$  in  $D$ .*

To apply our topological results, we will translate the weak realizability problem into a more topological version. Let  $G = (V, E)$  be a graph. Let  $M$  be the surface obtained from the plane by drilling  $|V|$  holes, each hole is labeled by a vertex of  $G$ . Let  $R \subseteq \binom{E}{2}$ . A set  $S$  of properly embedded arcs on  $M$  is called a *weak realization with holes* of  $(G, R)$  if for each  $e = \{u, v\} \in E$  there is a properly embedded arc in  $S$  connecting hole  $u$  to hole  $v$  and if  $(e, f) \notin R$  then the properly embedded arcs  $e, f$  are isotopically disjoint.

Given a weak realization  $D$  we can drill small holes in place of the vertices to obtain a weak realization with holes. Given a weak realization with holes by Lemma 3.2 there is a weak realization with holes in which for  $(e, f) \notin R$  the properly embedded arcs  $e, f$  are disjoint. Contracting the holes we obtain a weak realization of  $(G, R)$ .

PROPOSITION 4.2.  *$(G, R)$  is weakly realizable iff there is a weak realization with holes.*

LEMMA 4.3. *Let  $G$  be a graph with  $m$  edges and  $n$  vertices. Assume that  $(G, R)$  has a weak realization with holes. Let  $M$  be the surface obtained from the plane by drilling  $|V|$  holes. Let  $T$  be a minimal triangulation of  $M$ . Then there is a weak realization with holes of  $(G, R)$  such that there are at most  $2^{12n+m}$  intersections on each edge of  $T$ .*

PROOF. We can construct a triangulation  $T$  with  $3n$  vertices, using 3 vertices for each boundary component (hole) and no vertices in the interior of the surface. By a simple application of Euler's formula  $T$  has  $9n - 6$  edges.

Consider the following weak realization problem. Graph  $H$  has  $V_H = V_T \cup V_G$ . We have  $E_T \cup E_G \subseteq E_H$ . Moreover, there are edges connecting  $v \in V_G$  to all vertices of  $T$  which lie on the hole labeled by  $v$ . The pairs  $P$  of edges which are allowed to intersect are the following.

- All pairs in  $R$  are allowed to intersect.
- For every  $v \in G$  fix an edge  $e_v \in T$  on the boundary of the hole labeled by  $v$ . Then  $e_v$  can intersect with edges in  $E_G$  going to  $v$ .
- Any edge in  $T$  which is not on the boundary  $\partial M$  can intersect any edge in  $E_G$ .

Clearly  $(G, R)$  is weakly realizable iff  $(H, P)$  is weakly realizable.

Consider the realization of  $H$  with the smallest number of intersections. By Theorem 4.1 there is a realization such that on each edge there are at most  $2^{|E_H|} \leq 2^{12n+m}$  intersections. ■

THEOREM 4.4. *The weak realizability problem is in NP.*

PROOF. Let  $(G, R)$  be an instance of the weak realizability problem. Assume that  $(G, R)$  is weakly realizable. Hence by proposition 4.2 it has a weak realization with holes. Let  $T$  be a minimal triangulation of the surface  $M$ . By Lemma 4.3 there is a weak realization with holes in which each edge of  $T$  is intersected at most  $2^{12n+m}$  times. Hence for any arc  $\gamma$  and any edge  $e \in T$  the binary encoding of the number of intersections of  $\gamma$  with  $e$  has size polynomial in  $n$ . To verify weak realizability with holes it is therefore sufficient to guess, for each edge of  $G$ , a labeling of the edges of  $T$  whose

numbers are at most  $2^{12n+m}$ . We can then use Lemma 3.5, for any  $(e, f) \notin R$  to check in polynomial time that  $e, f$  are isotopically disjoint for the set of labelings we guessed. This, by Lemma 3.2, is sufficient to guarantee the existence of a weak realization of  $(G, R)$ . ■

## 5. STRING GRAPHS ON SURFACES AND TRACE MONOIDS

In proving Theorem 4.4 on string graphs in the plane there was only one place where we used the fact that the surface was a plane, namely when we applied Theorem 4.1 which has only been shown for the plane. The rest of the proof—Lemma 3.2 and the assumption that we can triangulate the surface in Theorem 4.4—hold true for any compact surface. However, we do not currently have any bounds on the number of intersections necessary to realize a string graph in a surface other than the plane, since the methods in the proof of Theorem 4.4 do not seem to apply to surfaces of higher genus than the plane. We make the following conjecture extending the original conjecture of Kratchovil and Matoušek [14].

**CONJECTURE 5.1.** *A graph that has a weak realization on a compact surface can be realized on that surface with at most  $O(2^{n^k})$  intersections for some fixed  $k > 0$ .*

If the conjecture were true, then the proof of Theorem 4.4 would generalize to compact surfaces of any genus, and we would obtain that the weak realization problem, and therefore the string graph problem, are in NP for arbitrary orientable surfaces. Meanwhile we can only state the following result, for which we let  $c_s^S(G)$  be the generalization of  $c_s(G)$  to a surface  $S$ .

**THEOREM 5.2.** *String graphs on a compact surface  $S$  can be recognized in  $\text{NTIME}(\log c_s^S(n))$ , where  $n$  is the number of vertices of the string graph.*

In the absence of any upper bound on  $c_s^S$  for any surface other than the plane this means that we face the question of decidability anew. This time we use a different approach already suggested by the proof of Theorem 4.1. We will use the connection between weak realizability and equations on words. Words over an alphabet form a monoid with concatenation as the operation on the monoid.

Our successful use of monoids in the proof of Theorem 4.4 relied on three facts: the exponential upper bound on  $c_w$  which allowed us to guess the lengths of the variables involved in the word equations; the fact that we only had to verify arcs that do not intersect making it possible to phrase the problem as an equation between words; and that we could assume the solution to be unique, resolving the question of how to deal with the operator  $\cdot^R$ .

Since we have none of these guarantees in the general case, we need stronger results on word equations:

- we cannot make assumptions on the lengths of variables,
- arcs can intersect, that is some pairs of letters must be allowed to commute, and
- we have to allow equations using  $\cdot^R$ .

The main problem here is the second item: allowing selected pairs of letters to commute. Let  $F$  be the free monoid over the alphabet  $\Sigma$ , and  $I$  be an irreflexive, symmetric relation on  $\Sigma$ .  $I$  defines an equivalence relation  $\equiv_I$  on  $F$  by taking the transitive and reflexive closure of  $\equiv_I^1$  which is defined as  $u \equiv_I^1 v$  if and only if  $u = xaby$ , and  $v = xbay$  for some  $(a, b) \in I$ . That is,  $\equiv_I$  identifies words in  $F$  that can be obtained from each other by permuting pairs of letters in  $I$ . The monoid  $F/\equiv_I$  is called a *trace monoid*. Trace monoids have been under investigation for a while, and Matiyasevich showed in 1996 that the solvability of equations in trace monoids is decidable [4]. For our purpose we need a stronger result that allows equations with the operator  $\cdot^R$ . This result was only recently obtained by Diekert and Muscholl, building on work by Diekert, Gutiérrez, and Hagenah [5, 3].

**THEOREM 5.3** (DIEKERT, MUSCHOLL [5]). *Solvability of systems of equations in trace monoids with involution is decidable.*

Techniques from the papers by Diekert and Muscholl [5], and Diekert, Matiyasevich, and Muscholl [4] can probably be combined to get an EXPSPACE upper bound on the complexity of the decision procedure.

We can apply this result to the string graph problem on an arbitrary compact surface  $S$ . Our goal is to decide whether  $(G, R)$  is weakly realizable in  $S$ . We add  $|G|$  holes to the surface  $S$ , and fix a triangulation  $T$  of the resulting surface. As we did in the proof of Lemma 3.4 we introduce a variable  $x_{u,v}$  for each edge  $(u, v)$  in  $T$ , and variables  $y_{t,u}, y_{t,v}, y_{t,w}, y_{u,t}, y_{v,t}, y_{w,t}$  for each triangle  $t$  in  $T$ . (See Figure 5.) We require that  $x_{u,v} = y_{u,t}y_{t,v}$ ,  $x_{v,u} = y_{v,t}y_{t,u}$ ,  $x_{v,w} = y_{v,t}y_{t,w}$ ,  $x_{w,v} = y_{w,t}y_{t,v}$ , and  $x_{u,w} = y_{u,t}y_{t,w}$ ,  $x_{w,u} = y_{w,t}y_{t,u}$ . Furthermore, we ask that  $y_{t,u} = y_{u,t}^R$  for all vertices  $u$ , and triangles  $t$  of  $T$ . We associate each vertex  $h$  of  $G$  arbitrarily with an edge  $(u, v)$  of  $T$  on the boundary of  $S$  such that no two vertices are associated with edges that share a vertex. Let  $\Sigma = \{e : e \text{ is an edge of } G\}$ . For each vertex  $h$  of  $G$ , let  $E(h)$  be the set of edges that contain  $h$ . For each  $h$  fix an arbitrary permutation  $p_h$  of these edges (as letters in  $\Sigma$ ), and add the equation  $x_{u,v} = e_h$  to the system, where  $(u, v)$  is the edge associated with  $h$ . Consider the trace monoid over  $\Sigma$  with independence relation  $R$ . We claim that the set of equations we built has a solution in that trace monoid, if and only if  $(G, R)$  is weakly realizable in  $S$ . This, however, is immediate, since the allowable crossings in the trace monoid correspond to the allowed intersections in  $R$ .

**COROLLARY 5.4.** *The string graph problem, and the weak realizability problem are decidable on any compact surface.*

A stronger form of the corollary would be to bound the number of intersections needed in the size of a solution to the trace monoid, however the bounds for that are currently well beyond the exponential bound we conjectured earlier.

## 6. ACKNOWLEDGMENTS

We would like to thank Volker Diekert and Géza Tóth for helpful discussions.

## 7. REFERENCES

- [1] S. Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Science*, 45:1607–1620, 1959.

- [2] Z.-Z. Chen, M. Grigni, and C. Papadimitriou. Planar map graphs. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pages 514–523, New York, 1998. ACM Press.
- [3] V. Diekert, C. Gutiérrez, and C. Hagenah. Existential theory of equations with rational constraints in free groups is PSPACE-complete. In *Proceedings of the 18th International Symposium on Theoretical Aspects of Computer Science, STACS 2001*, 2001.
- [4] V. Diekert, Y. Matiyasevich, and A. Muscholl. Solving trace equations using lexicographical normal forms. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming (ICALP'97), Bologna (Italy) 1997*, Berlin-Heidelberg-New York, 1997. Springer.
- [5] V. Diekert and A. Muscholl. Solvability of equations in free partially commutative groups is decidable. In *ICALP 2001*, pages 543–554, 2001.
- [6] B. Farb and B. Thurston. Homeomorphisms and simple closed curves. *unpublished manuscript*, 1991.
- [7] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):312–316, 1983.
- [8] L. Gaśieniec, M. Karpinski, W. Plandowski, and W. Rytter. Efficient algorithms for lempel-ziv encoding. in *Proceedings of SWAT'96, LNCS 1097*, pages 392–403, 1996.
- [9] R. L. Graham. Problem 1. In *Open Problems at 5th Hungarian Colloquium on Combinatorics*, 1976.
- [10] M. Grigni, D. Papadias, and C. Papadimitriou. Topological inference. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 901–906, 1995.
- [11] J. Kratochvíl. String graphs. I. the number of critical nonstring graphs is infinite. *Journal of Combinatorial Theory, Series B*, 52, 1991.
- [12] J. Kratochvíl. String graphs. II. recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, Series B*, 52, 1991.
- [13] J. Kratochvíl. Crossing number of abstract topological graphs. *Lecture Notes in Computer Science*, 1547:238–245, 1998.
- [14] J. Kratochvíl and J. Matoušek. String graphs requiring exponential representations. *Journal of Combinatorial Theory, Series B*, 53:1–4, 1991.
- [15] M. Lothaire. Algebraic combinatorics on words. To be published by Cambridge University Press, 2002.
- [16] J. Matoušek, J. Nešetřil, and R. Thomas. On polynomial time decidability of induced-minor-closed classes. *Comment. Math. Univ. Carolin.*, 29(4):703–710, 1988.
- [17] J. Pach and G. Tóth. Which crossing number is it anyway? *Journal of Combinatorial Theory, Series B*, 80:225–246, 2000.
- [18] J. Pach and G. Tóth. Recognizing string graphs is decidable. In P. Mutzel, editor, *Graph Drawing 2001*, Lecture Notes in Computer Science. Springer, 2001. to appear.
- [19] W. Plandowski and W. Rytter. Application of Lempel-Ziv encodings to the solution of words equations. In *Automata, Languages and Programming*, pages 731–742, 1998.
- [20] M. Schaefer and D. Štefankovič. Decidability of string graphs. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC-2001)*, pages 241–246, 2001. Invited to the JCSS special issue on STOC'01.
- [21] F. W. Sinden. Topology of thin film circuits. *Bell System Technical Journal*, 45:1639–1662, 1966.
- [22] M. Thorup. Map graphs in polynomial time. In *IEEE Symposium on Foundations of Computer Science*, pages 396–405, 1998.