

## 平面マップグラフ

陳 致中, MICHELANGELO GRIGNI, AND CHRISTOS H. PAPADIMITRIOU  
東京電機大学理工学部数理学科, EMORY UNIVERSITY, AND UC BERKELEY

### Abstract

地図上の2国家が国境線または国境点を共有しているならば、その2国家が隣国だと定義する。この隣国関係を表わした抽象グラフが与えられたとき、元の地図を構成する問題を考える。この問題がNPに属することを示す。さらに、元の地図において高々四つの国家が国境点を共有するという制限付きの場合を解く多項式時間アルゴリズムを提案する。制限無しの場合がNP困難か多項式時間解決可能かは未知である。

## Planar Map Graphs

ZHI-ZHONG CHEN, MICHELANGELO GRIGNI, AND CHRISTOS H. PAPADIMITRIOU  
TOKYO DENKI UNIVERSITY, EMORY UNIVERSITY, AND UC BERKELEY

### Abstract

We introduce and study a modified notion of planarity, in which two regions of a map are considered adjacent when they share any *point* of their boundaries (not an *edge*, as standard planarity requires). We seek to characterize the abstract graphs realized by such map adjacencies. We prove some preliminary properties of such graphs, and give a polynomial time algorithm for the following restricted problem: given an abstract graph, decide whether it is realized by a map in which at most four regions meet at any point. The general recognition problem remains open.

## 1 Introduction

### 1.1 Motivation: Topological Inference

Suppose that you are told that four planar regions relate in the following way:  $A$  is inside  $B$ ;  $B$  overlaps  $C$ ;  $C$  touches  $D$  on the outside;  $D$  overlaps  $B$ ;  $D$  is disjoint from  $A$ ; and  $C$  overlaps  $A$ . All four planar regions are “bubbles” with no holes (to be rigorous: disc homeomorphs). Is this possible? If so, we would like a model, a picture of four regions so related; if not, a proof of impossibility.

This deceptively simple extension of propositional logic is known as the *topological inference problem* [5], and its special cases, extensions, and variants are studied in the area of geographic information systems [3, 4, 10, 5, 11]. Despite much effort (and claims in the literature [12, 4]...), no decision algorithm and finite axiomatization for this problem is known — although the problem becomes both finitely axiomatizable and polynomial-time decidable in any number of dimensions other than two. In fact, the following special case has been open since the 1960’s [2]: We are given the status of all pairs of regions (we call this the *fully conjunctive case*) when two regions either overlap or are disjoint (that is, no two regions contain one another or touch on the outside). This problem is known as the *string graph problem*, because the information can be captured as a graph with the regions as nodes (overlaps/disjoint corresponds to adjacent/non-adjacent), and we can assume that the regions are in fact one-dimensional planar curves. In other words, we are seeking a recognition algorithm for the intersection graphs of planar curves. As we mentioned, it is open whether this problem is decidable; it is known that there are infinitely

many forbidden subgraphs; that recognition is at least NP-hard [7]; and that there are string graphs that require exponentially many string intersections for their realization [8].

The difficulty of the string graph problem exposes the fact that the complexity of topological inference stems to a large extent from the messy “overlaps” relation. But many practical applications are so structured that no two regions in them overlap (think of political maps, for example). What if we had a fully conjunctive formula in which the only relations between two regions that are allowed are “touches on the outside” and “disjoint”? In other words, *which graphs are the intersection graphs of closed disc homeomorphs with disjoint interiors?* This is the problem we study in this paper. It follows from our results that it is in NP (Corollary 2); however, whether it is in P is a most important and intriguing open problem, which we solve in an interesting and natural special case.

## 1.2 Motivation: Planarity, Revisited

Planarity is undoubtedly one of the most basic, ancient, and influential concepts in graph theory. The four color conjecture has been arguably the most famous and productive open problem in the area, recognizing planar graphs motivated the development of such basic methods as depth-first search and *pq*-trees, and planarity plays a central role in the recent work of Robertson and Seymour. Planar graphs may be defined as the intersection graphs of planar regions with disjoint interiors *such that no four regions meet at a point*. But what if the emphasized condition is removed? We obtain a very natural, intriguing, and heretofore little-studied class of graphs that we call *planar map graphs*. For example, the adjacency graph of the United States is a fine example of a planar map graph (in fact, in the special category of 4-planar graphs defined and studied later) which is non-planar (the “corner states” Arizona–New Mexico–Colorado–Utah form a  $K_4$ , which, together with Montana, creates a  $K_5$  minor). Actually, it is trivial to construct planar map graphs that are non-planar, since a *pizza* (Figure 1(a)) yields an arbitrarily large

clique.

It takes a little work even to show that the class of planar map graphs is in NP—but it is (Corollary 2). We want to establish that it is in P, that is, to find a polynomial-time recognition algorithm for planar map graphs. As we point out in Section 2, a naive reduction to ordinary planarity by “decomposing” pizzas does not work, *because maximal cliques in planar map graphs are not necessarily pizzas*. This complicates tremendously the recognition task, whose polynomial solution we, unfortunately, can at present only conjecture.

But suppose that we restrict our political maps so that *no more than  $k$  regions meet at a point*; we call the resulting class *k-planar graphs*. Thus, 3-planar graphs are precisely the ordinary planar graphs, and the U.S.A. is a 4-planar graph. *Our main result is a polynomial-time recognition algorithm for 4-planar graphs*. The algorithm is *very* complicated, as it must rely on a detailed case analysis of each maximal clique and its “immediate environment” (cliques intersecting it, and connected components in the complement graph).

It is an interesting philosophical question, why the forefathers of graph theory never bothered to define this class, despite the fact that it is, in our opinion, equally natural to ordinary planarity. We can think of three possible explanations: (a) one of those random lucky turns in intellectual history; (b) the result of deep foresight on the nastiness of the problem; or (c) the desire to state the four color conjecture—trivially false in the context of planar map graphs.

## 1.3 The Results of this Paper

In Section 2 we present a characterization of planar map graphs as the *half-squares of planar bipartite graphs* (Theorem 1). The half square of a planar bipartite graph is simply the square of the graph (two nodes are adjacent iff there is a path of length 2 in the original graph connecting them) restricted to one of the two sides of the bipartition. With a little more thought, this implies that planar map graph recognition is in NP (Corollary 2).

It would appear that planar map graphs can be recognized by the following *naive algorithm*:

1. Find the set  $\mathcal{C}$  of all maximal cliques of  $G$  with four or more nodes.
2. If  $|\mathcal{C}| \geq 12n$  then reply “not a planar map graph”.
3. Omit from  $G$  all edges that are in a clique in  $\mathcal{C}$ .
4. For each maximal clique  $C \in \mathcal{C}$ , add a vertex  $v_C$ , and edges from it to all nodes of  $C$ .
5. Test the resulting graph for planarity, and return the result.

That is, we identify all points at which more than three regions meet, and replace each with a fictitious region, connected to all of them (the graph theoretic analog of the circular piece in the middle of the pizza one sees in some restaurants). The naive algorithm is based on the following facts: (1) planar map graphs have  $O(n)$  maximal cliques, and (2) the maximal cliques of any graph can be output with polynomial delay between consecutive specimens output [6].

The reason why the naive algorithm fails is because *a maximal clique in a planar map graph can be realized in ways other than the pizza*, namely as a *pizza with crust*, a *hamantaschen*, and a *rice ball*, see Figure 1. Theorem 3 uses the characterization of Theorem 1 and planar graph theory techniques to prove that *these four are all possible realizations of a clique*.

In Section 3 we prove our main result, that 4-planar graphs can be recognized in polynomial time (Theorem 5). Our algorithm builds on the basic structure of the naive algorithm, examining each maximal clique of the graph in some carefully designed order: First cliques of size 6, then 5, then 4 (it is easy to see that 4-planar graphs have no cliques larger than six). For each clique, it considers its “environment” (intersecting cliques, and components of a certain “complement graph”) and succeeds—often after very sophisticated, but always linear-time, analysis—to *make progress*. There are five basic kinds of progress:

- We identify a maximal clique which must be realized as a pizza (and eventually treated by the naive algorithm).
- We identify four regions (as we call the nodes of the input graph) that must meet at a point in a specific cyclical order.
- We reduce the problem to one with fewer regions.
- More interestingly (and, it turns out, more often), we decompose the graph into components, and reduce the problem to testing whether each component is a 4-planar graph. The reason such decompositions are possible is that all realizations of maximal cliques in Figure 1, except for the pizza, have only triangular “holes” (unoccupied planar regions within which more regions can be embedded). Thus each component resulting from its deletion can be separately checked for 4-planarity.
- Finally, in certain more complicated cases we identify a way of recursing on a similar maximal clique, albeit in a smaller graph.

The case analysis involved is very tedious (over a hundred cases must be examined); in Section 3 we include a top-level summary without detailed proofs; for a draft of the complete proof see [1]. The objects studied in the case analysis are *partial maps*, that is, sets of planar regions corresponding to the part of the graph being examined, with space left for embedding the rest. We refine the maps by bringing in more regions until we reach a *final map*, one in which all unoccupied holes have at most three regions around them (and thus the graph can be decomposed in a lossless way)—or until we make progress in any one of the other four ways listed above. It turns out that the methods are very different for the three clique sizes.

The straight-forward analysis of the running time of the algorithm yields an  $O(n^3)$  upper bound. It can be probably reduce to  $O(n^2)$  by a more careful analysis, with some hope of bringing it down to  $O(n \log n)$  (the best known running time for enumerating all maximal cliques, see [6]).

## 2 Planar Map Graphs

### 2.1 A Characterization

Consider a collection  $\mathcal{R}$  of  $n$  regions in the plane, each homeomorphic to a disc, so that no two regions overlap except possibly on their boundaries; these adjacencies define a planar map graph  $G$ . A typical boundary point is shared by one or two regions, however there may also be exceptional points where three or more regions touch. Consider the sequence of adjacency changes around any one region, ignoring "empty" stretches. A simple argument shows this sequence is finite (in fact linear); hence a finite collection  $\mathcal{P}$  of points witnesses all adjacencies among the regions of  $\mathcal{R}$ .

In each region  $R$  we choose a representative interior point, and connect it with arcs through the interior of  $R$  to the points of  $\mathcal{P}$  bounding  $R$ . In this way we construct a bipartite planar graph  $G' = (\mathcal{R}, \mathcal{P}, E')$ , so that any two regions  $R_1$  and  $R_2$  overlap iff they have distance two in  $G'$ . Thus  $G$  equals  $G'^2|_{\mathcal{R}}$ , the square of  $G'$  restricted to  $\mathcal{R}$ .

Conversely, given a bipartite planar graph  $G'$ , we may reverse the construction to find a corresponding arrangement of regions and bounding points. Hence we have:

**Theorem 1** *A graph is a planar map graph iff it is the half square of a planar bipartite graph.* ■

**Corollary 2** *The recognition problem for planar map graphs is in NP.*

We also make some simple initial remarks:

- In the bipartite graph representation, bounding points of degree three may be replaced by points of degree two.
- If  $G$  has no 4-clique, then it is a planar map graph iff it is a planar graph.
- A planar map graph may contain cliques of arbitrary size.
- From the previous two remarks, it is clear that the "planar map graph" property is

not monotone, and hence cannot be characterized by forbidden subgraphs or minors.

### 2.2 Cliques in Planar Map Graphs

Consider a planar map clique of size  $n$ , it may be realized in one of the four following ways:

1. The  $n$  regions share a single boundary point. We call this the *pizza* (Figure 1(a)).
2. Some  $n - 1$  regions share a single boundary point, and the one remaining region is arbitrarily connected to them at other points. We call this the *pizza with crust* (Figure 1(b)).
3. If  $n \geq 6$ , there may be three points supporting all adjacencies in the clique, with at most  $n - 2$  regions at any one point. In particular, there are at most two regions adjacent to all three of the points. We call this the *hamantaschen* (Figure 1(c)).
4. An ordinary planar clique (that is, with no points of degree more than three), such as the *rice ball* (the planar  $K_4$ , Figure 1(d)).

**Theorem 3** *A planar map graph clique must be one of the above four types.*

**Corollary 4** *The number of cliques of size 4 or more in a planar map graph with  $n$  nodes is at most  $12n$ .* ■

### 2.3 $k$ -Planar Graphs

Our attempts at a polynomial-time algorithm for recognizing planar map graphs have failed (see the last section for a discussion). Consider however the interesting special case in which the maps are restricted to be such that no more than  $k$  regions share a point. We call the class of graphs that are realized by such maps  *$k$ -planar graphs*. It is easy to see that 3-planar graphs are the ordinary planar graphs, and that the USA graph is 4-planar. It is easy to extend Theorem 1, to characterize  $k$ -planar graphs as the half-squares of bipartite planar graphs whose right-hand side has degrees  $k$  or less.

In the next section we focus on 4-planar graphs and their recognition algorithm. It follows from Theorem 3 that 4-planar graphs have no 7-cliques, that all 6-cliques are hamantaschens, all 5-cliques are pizzas with crust, and all 4-cliques are either pizzas, or three regions touching at three points and enclosing a fourth (variants of the rice ball). Finally, an eight-node example, omitted in this abstract, shows that 4-planar graphs are non-monotone (in that deletion of an edge may turn a 4-planar graph into a graph that is not 4-planar), and hence polynomial-time recognition does not follow from first principles.

### 3 Recognition of 4-Planar Graphs

We sketch the proof of our main result:

**Theorem 5** *4-planar graphs can be recognized in polynomial time.*

#### 3.1 Preliminaries

Let  $G$  be a graph. A *map*  $\mathcal{L}$  is a finite set of planar regions that are disc homeomorphs with disjoint interiors. A map is a *realization* of  $G$  (or a *map of*  $G$ ) if its regions are in one-to-one correspondence to the vertices of  $G$ , and in which two regions touch each other iff the corresponding vertices are adjacent in  $G$ . A map of  $G$  is called a *4-map of*  $G$  if no five regions meet each other at a point. To prove the theorem, we must design a polynomial-time algorithm which given  $G$ , constructs a 4-map of  $G$  if one exists, and reports “failure” otherwise. Since it is trivial to check whether a given map is a realization of a given graph, we may assume that  $G$  has a 4-map and only need to show how to find one. Without loss of generality, we may further assume that  $G$  is biconnected.

We call vertices of  $G$  *regions*. For a region  $c \in V(G)$ ,  $N_G(c)$  denotes the set of regions adjacent to  $c$  in  $G$ . Let  $U \subseteq V(G)$  and  $F \subseteq E(G)$ .  $N_G(U) = \cup_{c \in U} N_G(c)$ , and  $G[U]$  denotes the subgraph of  $G$  induced by  $U$ .  $G - U - F$  denotes the graph obtained from  $G$  by deleting the edges in  $F$  and the regions (together with

the edges incident to them) in  $U$ . For a subset  $W$  of  $U$ ,  $\mathcal{C}_{U,F}^G(W) = \{c \in V(G) - U \mid W = N_G(c) \cap U, \text{ where } K \text{ is the connected component of } G - U - F \text{ containing } c\}$ . When  $U$  or  $F$  is empty, we drop it from the notations  $G - U - F$  and  $\mathcal{C}_{U,F}^G(W)$ .

An *extensible 4-map* of  $G[U]$  is a 4-map of  $G[U]$  that can be extended to a 4-map of  $G$ . For  $k = 2, 3, 4$ , a *k-point* in a map is a point at which exactly  $k$  regions meet. A maximal clique of size  $k$  is denoted by  $MC_k$  (recall that  $G$  has no  $MC_k$  with  $k \geq 7$ ). Let  $l$  be a positive integer. We say that two maximal cliques  $C$  and  $C'$  are *l-sharing* if  $|C \cap C'| = l$ .

**Definition 1** A *correct 4-point* is a cyclicly ordered list  $\langle c_0, \dots, c_3, c_0 \rangle$  of four regions in  $G$  such that  $G$  has a 4-map in which (1) the four regions  $c_0$  through  $c_3$  meet at a single point (say,  $p$ ) in this order and (2) whenever  $c_0$  and  $c_2$  (or  $c_1$  and  $c_3$ , respectively) together with two other regions  $d'$  and  $d''$  meet at a point  $q \neq p$ , the cyclic order of the four regions around  $q$  is  $c_0, d', c_2, d''$ ,  $c_0$  (respectively,  $c_1, d', c_3, d''$ ,  $c_1$ ). *Removing* a correct 4-point entails adding a new region and replacing the 4-clique by a wheel (in the indicated cyclic order) centered in the new region.

**Lemma 1** Let  $G'$  be the graph obtained from  $G$  by removing a correct 4-point  $P = \langle c_0, \dots, c_3, c_0 \rangle$ . Then, (1)  $G'$  has a 4-map, (2) if  $G'$  has neither  $MC_5$  nor  $MC_6$   $G'$  has fewer  $MC_4$ 's, and (3) given an arbitrary 4-map of  $G'$ , we can construct a 4-map of  $G$  in linear time.

#### 3.2 Outline of the algorithm

We say that a 4-map  $\mathcal{L}$  of  $G[U]$  can be *transformed* to another 4-map  $\mathcal{L}'$  of  $G[U]$  if whenever  $\mathcal{L}$  is extensible, so is  $\mathcal{L}'$ . A map is said to be *explicit* if all points in it are distinct except that for one or more holes enclosed between exactly two regions, the two 2-points on the boundary of each of these holes may actually be identical; a map that is not explicit is *rough*. A explicit map  $\mathcal{L}$  is said to be *final* if there is no 3-point in it and every hole in it is enclosed by at most 3 regions. Recall that  $G$  is assumed to have a 4-map realization. Our algorithm starts

by enumerating all the maximal cliques of size  $\geq 4$  in  $G$  —by Corollary 4 there are  $O(|V(G)|)$  of them. We deal with the  $MC_6$ 's,  $MC_5$ 's, and  $MC_4$ 's in  $G$ , in this order.

**$MC_6$ 's.** Let  $C = \{c_1, c_2, \dots, c_6\}$  be an  $MC_6$  in  $G$ . It is easy to see that every extensible 4-map of  $C$  can be transformed into another of the form shown in Figure 2. As in all displayed maps of cliques, in this figure the regions 1, 2, 3, 4, 5, and 6, are a permutation of the nodes in the clique. A typical map that we display during the case analysis is in fact an equivalence class of maps, in the sense that different points in it may or may not coincide. However, Figure 2 is *explicit*; by this we mean that different points in it represent distinct points of the map —with a single exception: The two points delimiting a hole between two regions, such as  $p$  and  $q$  in this figure, could coincide. Figures that are not explicit are called *rough*. We call an explicit map *final* if there is no 3-point in it and every hole in it is enclosed by at most 3 regions. Notice that Figure 2 is final. Our treatment of  $MC_6$ 's is based on the following result:

**Theorem 6** Let  $S = \{(1,2), (3,4), (5,6), (2,3), (2,5), (3,5), (1,4), (1,6), (4,6)\}$ , and  $T = \{(2,3,5), (1,4,6)\}$ . Then, for every permutation  $\pi = (1, \dots, 6)$  of  $(c_1, \dots, c_6)$ , the 4-map in Figure 2 is extensible iff the family  $\mathcal{F} = \{C_G^{\mathcal{C}}(\{i,j\}) \mid (i,j) \in S\} \cup \{C_G^{\mathcal{C}}(\{i,j,k\}) \mid (i,j,k) \in T\}$  is a partition of  $V(G) - C$ .

By Theorem 6, we can compute an extensible 4-map of  $C$  in linear time. Then we recursively find a realization of the subgraph of  $G$  induced by  $\{i,j\} \cup C_G^{\mathcal{C}}(\{i,j\})$  for every pair  $(i,j) \in S$ , and one of the subgraph of  $G$  induced by  $\{i,j,k\} \cup C_G^{\mathcal{C}}(\{i,j,k\})$  for every triple  $(i,j,k) \in T$ ; each of the graphs in the recursive calls has fewer  $MC_6$ 's than  $G$ , and the total number of regions in these graphs is larger than that in  $G$  by only a constant.

**$MC_5$ 's.** We have thus removed all  $MC_6$ 's from  $G$ . Our algorithm then proceeds to removing  $MC_5$ 's from  $G$ . It is not difficult to see that the five regions in every  $MC_5$  must form

a “pizza with crust” in every 4-map of  $G$ . (A hamantaschen of five regions is actually a pizza with crust.) Thus, in every extensible 4-map of an  $MC_5$   $C$ , there is a point shared by exactly four regions in  $C$ . This motivates the following definition:

**Definition 2** Let  $C$  be an  $MC_5$  in  $G$ . A *correct center* of  $C$  is a cyclicly ordered list  $\langle c_0, \dots, c_3, c_0 \rangle$  of four regions in  $C$  such that  $C$  has an extensible 4-map in which the four regions  $c_0$  through  $c_3$  meet at a single point in this order. A *correct crust* of  $C$  is a region  $c \in C$  such that the four regions in  $C - \{c\}$  constitute a correct center of  $C$  (in some way).

To remove an  $MC_5$   $C$  from  $G$ , the basic idea is to find an extensible 4-map of  $C$  and then remove its center.

To find an extensible 4-map of  $C$ , our algorithm constructs a rough 4-map  $\mathcal{L}$  of  $C$ , and then calls the following procedure with argument  $\mathcal{S} = \{\mathcal{L}\}$ :

**Procedure *Make\_Final*( $\mathcal{S}$ )**

1. By distinguishing certain cases, from the rough 4-maps in  $\mathcal{S}$ , construct a set of explicit 4-maps (of the same set of regions as in the 4-maps in  $\mathcal{S}$ ) at least one of which must be extensible whenever an extensible 4-map (of the same set of regions) exists. Update  $\mathcal{S}$  to be the set of the constructed explicit 4-maps.
2. If some 4-map in  $\mathcal{S}$  is not final, then perform the following steps:
  - 2.1. Select a certain set  $A$  of regions that has not appeared in the 4-maps in  $\mathcal{S}$ .
  - 2.2. For each 4-map  $\mathcal{L} \in \mathcal{S}$ , if there is no way to add the regions in  $A$  into  $\mathcal{L}$ , then delete  $\mathcal{L}$  from  $\mathcal{S}$ ; otherwise, add the regions in  $A$  into  $\mathcal{L}$ .
  - 2.3. If  $\mathcal{S}$  is empty, then return “failure”; otherwise, goto step 1.
3. For each final 4-map in  $\mathcal{S}$ , based on a certain necessary and sufficient condition (analogous to Theorem 6), decide whether the 4-map is extensible or not.

To examine procedure *Make\_Final* more closely, let  $C = \{c_1, c_2, \dots, c_5\}$  be an  $MC_5$  in  $G$  and let us follow it for one iteration. Figure 3 shows one of the starting rough 4-maps of  $C$ . This figure is rough, because, for example, any two adjacent points from among the five contact points in the upper half-perimeter of the circle could coincide. Our algorithm sets  $S$  to be the set of this rough 4-map and calls *Make\_Final(S)*. To construct a set of explicit 4-maps from the rough 4-map in  $S$ , procedure *Make\_Final* distinguishes three cases based on  $n_{C,4s}$ , the number of  $MC_5$ 's 4-sharing with  $C$  in  $G$ .

*Case 1:*  $n_{C,4s} = 2$ . Then, every extensible 4-map of  $C$  can be transformed to one of the last three 4-maps in Figure 4 each of which is explicit. At the end of step 1 (of the first iteration of procedure *Make\_Final*),  $S$  becomes the set of these three explicit 4-maps. Let the two  $MC_5$ 's 4-sharing with  $C$  be  $C_1$  and  $C_2$ . Let  $C_1 - C = \{c_6\}$ ,  $C_2 - C = \{c_7\}$ ,  $C - C_1 = \{c_1\}$ , and  $C - C_2 = \{c_4\}$ . Then, procedure *Make\_Final* adds  $c_6$  and  $c_7$  to the three 4-maps in  $S$  and gets three larger 4-maps as shown in Figure 5. Figure 5(a) is final while the other two are rough. With  $S$  being the set of the three rough 4-maps in Figure 5, procedure *Make\_Final* proceeds to the second iteration. We can prove that after at most two further iterations, procedure *Make\_Final* will (1) find an extensible 4-map of  $C$ , (2) report "failure", or (3) succeed in decomposing  $G$  into graphs of fewer vertices or fewer  $MC_5$ 's and then recurse on each.

*Case 2<sup>1</sup>:*  $n_{C,4s} = 1$ . Then, every extensible 4-map of  $C$  can be transformed to one of the last four 4-maps in Figure 4 each of which is explicit. At the end of step 1,  $S$  becomes the set of these four explicit 4-maps. Let the  $MC_5$  4-sharing with  $C$  be  $C_1$ . Let  $C_1 - C = \{c_6\}$  and  $C - C_1 = \{c_4\}$ . Then, pro-

<sup>1</sup>Actually, only after removing all the  $MC_5$ 's 4-sharing with exactly two  $MC_5$ 's in  $G$ , our algorithm proceeds to removing those  $MC_5$ 's 4-sharing with exactly one  $MC_5$  in  $G$ . Thus, during the construction of an extensible 4-map of an  $MC_5$  4-sharing with exactly one  $MC_5$  in  $G$ , our algorithm often makes use of the fact that every  $MC_5$  in the current graph is 4-sharing with at most one  $MC_5$ .

cedure *Make\_Final* adds  $c_6$  to the four 4-maps in  $S$  and gets four larger 4-maps. With  $S$  being the set of the four rough 4-maps, procedure *Make\_Final* proceeds to the second iteration. We can prove that after at most two further iterations procedure *Make\_Final* will either find an extensible 4-map of  $C$  or report "failure".

*Case 3:*  $n_{C,4s} = 0$ . This is the last and most involved case for  $MC_5$ 's, as we must further distinguish four cases based on  $n_{C,3s}$ , the number of  $MC_4$ 's 3-sharing with  $C$  in  $G$ .

$MC_4$ 's. Once we have removed  $MC_6$ 's and  $MC_5$ 's from  $G$ , we proceed to the  $MC_4$ 's. This is in fact the most complex and tedious part of the algorithm and the case analysis.

## 4 Discussion and Open Problems

There is an interesting variant of the problem, in which we require that the union of the regions be a simply connected region, with no holes—that is to say, we do not allow "lakes" between the regions. There is a similar characterization as that of Theorem 1 for this case; the only difference is that now all internal faces of the planar bipartite graph must have length four and six. A variant of our algorithm works in this case as well.

There are two more interesting generalizations of the problem, motivated by topological inference: What if the relation between certain pairs of regions (touch/do not touch) is left unspecified—that is, we are given a graph with "don't care" edges? And what if we also allow *inclusion* relationships between regions? We conjecture that the first problem is NP-complete (for the general problem, and the 4-planar special case), while the latter polynomial.

## References

- [1] Z. Chen, M. Grigni. C. H. Papadimitriou "Planar map graphs," <http://www.mathcs.emory.edu/~mic/pmg/>.

[2] Ehrlich, G.; Even, S.; Tarjan, R. E. "Intersection graphs of curves in the plane" *J. Combinatorial Theory Ser. B* 21 (1976), no. 1, 8–20.

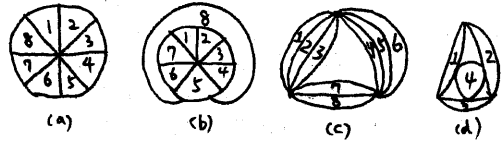


Figure 1.

[3] M. J. Egenhofer. Reasoning about binary topological relations. In Gunther, O. and Schek, H.J. (eds.), *Advances in Spatial Databases, SSD'91 Proceedings*, Springer Verlag, 143–160, 1991.

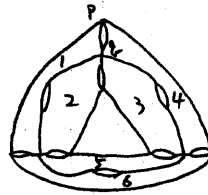


Figure 2

[4] M. J. Egenhofer and Jayant Sharma. Assessing the consistency of complete and incomplete topological information. *Geographical Systems*, 1:47–68, 1993.

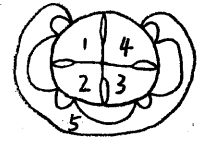


Figure 3

[5] M. Grigni, D. Papadias and C. H. Papadimitriou. Topological Inference. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 901–906, 1995.



(a)

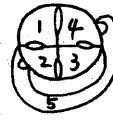


(b)



(c)

[6] D. S. Johnson, M. Yannakakis, C. H. Papadimitriou "On generating all maximal independent sets," *IPL* 27, 3, pp. 119–123, 1988.



(d)

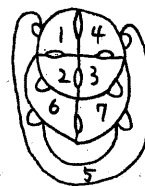


(e)

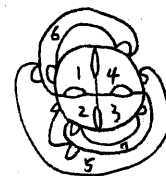
[7] J. Kratochvíl. String graphs II: Recognizing string graphs is NP-hard. *Journal of Comb. Theory, Series B*, 52(1):67–78, 1991.

Figure 4

[8] J. Kratochvíl and J. Matoušek. String graphs requiring exponential representations. *Journal of Comb. Theory, Series B*, 53(1):1–4, 1991.



(a)

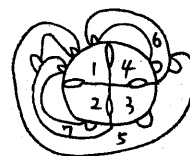


(b)

[9] D. Papadias and T. Sellis. The Qualitative Representation of Spatial Knowledge in Two-Dimensional Space. *Very Large Data Bases Journal, Special Issue on Spatial Databases*, 4:479–516, 1994.

[10] C. H. papadimitriou, D. Suciú, V. Vianu "Topological Queries in Spatial Databases", *Proc. 1996 PODS*. To appear in the special *JCSS* issue, 1997.

[11] Terence R. Smith and Keith K. Park. Algebraic approach to spatial reasoning. *International Journal Geographical Information Systems*, 6:177–192, 1992.



(c)

Figure 5