A Sperner Lemma Complete for PPA

Michelangelo Grigni [1]

Department of Mathematics and Computer Science, Emory University, Atlanta GA 30322, USA

Abstract

The class PPA characterizes search problems whose solution is guaranteed by the lemma that "every finite graph with an odd degree vertex has another." The smaller class PPAD is defined similarly for directed graphs. While PPAD has several natural complete problems corresponding to classical existence theorems in topology, no such complete problems were known for PPA. Here we overcome the difficulty by considering non-orientable spaces: Sperner's lemma for non-orientable 3-manifolds is complete for PPA.

Key words: theory of computation, formal languages

1 Introduction

Papadimitriou [1] defined the class PPA of NP search problems where a solution must exist by a *parity lemma*: "every finite graph with an odd-degree vertex has another." That is, an exponential size graph is defined by a Turing machine which answers adjacency list queries in polynomial time; we are given one odd-degree vertex, and the search problem is to find another. By a simple reduction, we may assume that the graph has maximum degree two, and the Turing machine simply returns the list of neighbors of a given node.

The class PPAD is defined similarly for directed graphs, and by the same reduction we may assume that every node has indegree and outdegree at most one. The search problem becomes: given a directed graph with a known source (a vertex with an out-edge but no in-edge), find another source or a sink. By ignoring the directions on the edges, we see that PPA contains PPAD.

[1] Supported by NSF grant CCR-9820931. E-mail: mic@mathcs.emory.edu.

The classes PPA and PPAD resemble the earlier class PLS [2], where the goal is to find a local minimum (or maximum) of a cost function on the graph. All these classes lie within TFNP [3], the class of search problems where a solution always exists. Since TFNP is unlikely to have complete problems [1], we are motivated to find complete problems for interesting subclasses of TFNP.

Problems in PPA and PPAD have an obvious time-consuming solution: walk from the given vertex to the other end of its path. It seems likely that PPAD is strictly contained by PPA, although an actual separation would imply P$\neq$NP. As weaker evidence, it is known that PPA^G strictly contains PPAD^G for generic oracles G [4]. Roughly stated, PPAD problems may be easier because of the following slight advantage: if you jump to a random vertex, you know which way to walk to avoid returning to the original source vertex.

This "advantage" occurs naturally in topology: several classical existence results (Sperner, Brouwer, Kakutani, Borsuk-Ulam, Nash, Arrow-Debreu) have computational search versions that turn out to be PPAD-complete [1]. However, no such problems were found complete for PPA. We show that the directed nature of these theorems depends on the orientability of the underlying topological space: specifically, Sperner's Lemma for non-orientable 3-manifolds is PPA-complete.

2 The Class PPA

Suppose M is a nondeterministic polynomial time Turing machine; on an input x, M may either fail, or it may succeed with some output string y. M defines a search problem: given x, find such a y if one exists. The class of such search problems is FNP. If we are guaranteed that for every x there exists a y, then the search problem is *total*, and lies in the subclass TFNP. However, this guarantee is nonrecursive, and TFNP apparently has no natural complete problems. In search of interesting problems inside TFNP, Papadimitriou [1] defined several syntactic subclasses of TFNP with natural complete problems, where the existence of a solution guaranteed by an "existence lemma."

In particular, the class PPA is based on the parity argument. The class PPA consists of the following core problems (one for each such machine M), closed under polynomial time reductions.

For an input string x, the *configuration space* $C(x) = \{0,1\}^{|x|}$ serves as a set of graph vertices. We consider polynomial time machines M meeting the following restrictions for all inputs x:

- For $u \in C(x)$, $M(x,u)$ returns the list of the neighbors of u as a tuple:

$\langle v, w\rangle$, $\langle v\rangle$, or $\langle\rangle$ (where $v < w$ and $v, w \in C(x) - \{u\}$).
- For $u, v \in C(x)$, $v \in M(x, u)$ whenever $u \in M(x, v)$.
- $M(x, \bar{0})$ is a one-tuple (where $\bar{0} = 0^{|x|}$), so vertex $\bar{0}$ has degree one.

Given x, the search problem is to find a degree-one vertex in $C(x) - \{\bar{0}\}$.

Since PPA is closed under reductions, we may effectively search larger graphs by padding x, and also do further computation to simplify y. Although the above restrictions on M are nonrecursive, we may get the same class in a syntactic way by considering a recursive enumeration of machines M with runtime "sanity–checking" to make their outputs conform to the restrictions.

3 The General Sperner Problem

We need some notions from topology; precise definitions may be found in texts such as [5]. A *d-manifold* is a topological space covered by open neighborhoods homeomorphic to the Euclidean space $\mathbb{R}^d$. We also consider d-manifolds *with boundary*, where we allow neighborhoods homeomorphic to the half-space $\{x \in \mathbb{R}^d \mid x_1 \geq 0\}$. For example, the Möbius strip (take a long rectangle and identify the two short sides with opposite orientation, to form a loop with a half-twist) is a 2-manifold whose boundary is a single loop.

In a Euclidean space, a *d-simplex* is the convex closure of $d+1$ affinely independent points. A *face* of a d-simplex is the convex closure of a proper subset of its corner points, and a *facet* is a face with d corners. For example a 3-simplex is a tetrahedron, its four facets are triangles, and its other faces are points and edges. Within a manifold, a d-simplex is a homeomorphic image of a Euclidean d-simplex.

Given a d-manifold, a *d-triangulation* is a finite collection of d-simplices covering the manifold, such that each pair of simplices is either disjoint or intersecting on a common face. Each facet is shared by at most two of the d-simplices; if it appears in only one d-simplex, it is in the boundary (and called a boundary facet). For example, a cube may be partitioned into six tetrahedra sharing a diagonal; therefore the cube has a 3-triangulation with eight points and twelve boundary triangles.

Given a d-triangulation, we may color its points with the colors $\{0, 1, \ldots, d\}$. A *full-color simplex* is a d-simplex with all $d+1$ colors at its corner points, and a *full-color facet* is a facet with all the colors $\{1, \ldots, d\}$ at its corners. Suppose we are given a d-triangulation, a coloring with no full-color boundary facet, and a full-color simplex. Then *Sperner's Lemma* states that there exists another full-color simplex.

This follows from the parity argument applied to the graph whose vertices are the d-simplices, and two d-simplices are adjacent if they share a full-color facet. This graph has maximum degree two, and the full-color simplices are exactly the vertices with degree one.

Assuming an effective polynomial time presentation of an exponential size d-triangulation, a coloring, and a known full-color d-simplex (all depending on x), we get a computational search problem SPERNER, which lies in PPA by the preceding argument. Furthermore, if the triangulation is effectively orientable, so that we can efficiently compute the orientation of any facet, then the graph is directed (with no degree-2 source or sink), and this restricted version of SPERNER lies in PPAD. For a special case where our manifold is a tetrahedron and the triangulation depends only on $|x|$, Papadimitriou showed that SPERNER is PPAD-complete.

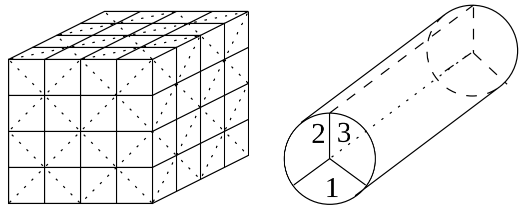
4 The Construction

Here we show that a special case of SPERNER is hard for PPA. This argument uses several elements from the result of Papadimitriou, in particular the "tri-colored tubes." First we fix a space and a standard triangulation, depending only on an even size parameter N.

We choose a non-orientable 3-manifold X homeomorphic to the product of the Möbius strip and the closed unit interval $I = [0, 1]$. We construct X from the unit cube I^3 by identifying the boundary points $(0, x_2, x_3)$ and $(1, 1 - x_2, x_3)$ for all $(x_2, x_3) \in I^2$; call these points in X the "orientation-reversing face." The remaining faces of the cube form the boundary of X; in our colorings all boundary points will get color 0. Let X inherit the coordinates from I^3. We remark that any non-orientable 3-manifold contains a subset homeomorphic to X in the neighborhood of an orientation-reversing path.

Given N even, we define a regular triangulation of X of size $\Theta(N^3)$. Begin with a partition of the unit cube into N^3 cubelets of side-length $1/N$. Further partition the cubelets with all cutting planes of the form $x_i \pm x_j = 2k/N$, where $1 \le i < j \le 3$ and k is an integer (Figure 1(a)). Each cubelet is cut into six tetrahedra (as in the cube example of the previous section), and these tetrahedra define a triangulation of both I^3 and X.

An instance of this restricted SPERNER problem is a polynomial time Turing machine S describing a 4-coloring of the vertices of the triangulation. More precisely, for an input x, $N = 2^{p(|x|)}$ (for some polynomial p encoded in S) and for each $i, j, k \in \{0, \dots, N\}$, $S(x; i, j, k) \in \{0, 1, 2, 3\}$ is the color of the point at coordinates $(i/N, j/N, k/N)$, with the restriction that $S(x; 0, j, k) =$



(a) (b)

Fig. 1. The $N = 4$ triangulation, and a tri-color tube.

$S(x; N, N-j, k)$. Furthermore, there will be a full-color simplex at some known position.

The coloring we construct will have color 0 almost everywhere (thought of as transparent), with the other three colors appearing in tri-colored tubes as in Figure 1(b). These tubes are defined by a piece-wise linear center lines (taking occasional right turns) and a $\Theta(1/N)$ radius large enough so that no cubelet, and hence no tetrahedron, can be full-colored except at an end of a tube. Also the tubes must be sufficiently thin to avoid accidental overlapping; these constraints will lower bound N. For definiteness say that the wedge of color 3 always points in the $+x_3$ direction, thus the two wedges of color 1 and 2 swap places (in the x_2 direction) whenever a tube passes through the orientation-reversing face.

Given a PPA problem defined by a polytime machine M as in Section 2, we must construct a polytime machine S so that for each input x: from a solution of the SPERNER problem defined by S on x, in polynomial time we can recover a solution to the PPA problem defined by M on x.

First we sketch the construction. Given x, for each vertex v in the configuration space $\{0,1\}^{|x|}$ we associate some point p_v in X. Furthermore, we route tubes between these points in a simple enough manner so that given a point $q = (i/N, j/N, k/N) \in X$ (with coordinates inherited from I^3), a polytime machine can find whether some tube contains q, and hence its color.

A tri-colored tube has a local orientation from I^3, and inside X this orientation reverses precisely when the tube passes through the orientation-reversing face. In the neighborhood of each p_v there is at most one potential in-tube and a matching potential out-tube. If v has degree one, that is $M(x, v) = \langle u \rangle$, then we use only the out-tube to reach p_u. If v has degree two, say $M(x, v) = \langle u, w \rangle$, then we use the out-tube to reach p_u and the in-tube to reach p_w. The difficulty here is that since M describes an *un*directed graph, it could happen that $M(x, u) = \langle v, t \rangle$, so the tube between p_u and p_w must be an out-tube at both ends. Our remedy is to use the orientation reversing face: such out-out (and in-in) connecting tubes must pass through the orientation-reversing face in order to have the required local orientations at their two endpoints.

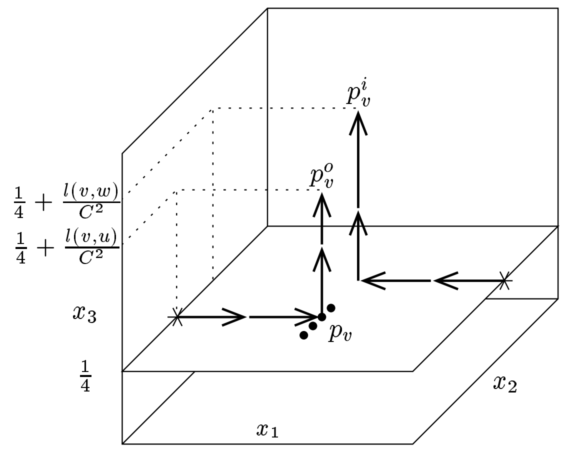


Fig. 2. Tubes from p_v when $M(x, v) = \langle v, w \rangle$.

At this point, almost any reasonable tube routing arrangement will work; we require that the tubes avoid accidental intersections, and that we can efficiently decide which tube, if any, contains a given point. We propose the following simple scheme (more clever schemes could greatly improve the resulting N).

For each vertex v in the configuration space $\{0, 1\}^{|x|}$, interpret v as an integer in the range 0 to $C-1$, where $C = 2^{|x|}$. Furthermore, let $l(v, w) \in \{1, \dots, \binom{C}{2}\}$ be a numbering of the unordered pairs, so that $l(v, w) = l(w, v)$.

For each v let its point in X be $p_v = (1/2, (v + C)/4C, 1/4)$. If p_v needs an out-tube towards some p_u (that is, $M(x, v)$ is $\langle u \rangle$ or $\langle u, w \rangle$), route its initial segment "upwards" by increasing x_3, reaching $p_v^o = (1/2, (v + C)/4C, 1/4 + l(v, u)/C^2)$. Similarly if p_v needs an in-tube towards p_w, route this tube by first increasing x_1 by one (wrapping through the orientation-reversing face), and then up to the point $p_v^i = (1/2, (3C - v)/4C, 1/4 + l(v, w)/C^2)$. Note that since the in-tube wrapped around, they are now both locally "out" oriented (Figure 2).

Now to finish realizing an edge $\{v, w\}$, with $v < w$, we connect the two tubes inside the slice $x_3 = 1/4 + l(v, w)/C^2$. These may have started as in-tubes or out-tubes at p_v and p_w, but they are now both out-tubes reaching p'_v (one of p_v^i or p_v^o) and p'_w (one of p_w^i or p_w^o). This out-out connection must pass once more through the orientation reversing face: starting at p'_v, increasing x_1 by 3/4 (wrapping around), adjust x_2 to match p'_w, and finally increase x_1 by another 1/4 to close the connection. Note this path also avoids the other x_3-parallel tubes passing through this slice.

Note that there is at least one unterminated tube, in the neighborhood of $p_{\bar{0}}$, which has one out-tube and no in-tube. This gives us the known starting full-color simplex for SPERNER. We need $N = \Theta(C^2)$ to have enough different x_3 coordinates to realize all these tubes. Most importantly, by simply examining the coordinates of a point in X, we can quickly identify at least one of the two vertices (if any) involved in a tube containing the point, and then by consulting M, we can find the other endpoint and decide whether a tube contains the given point.

There are further technical issues: what is the appropriate radius of the tubes, particularly in neighborhoods of right turns; and how to slowly rotate the tube colors if necessary. These issues are already addressed by Papadimitriou.

5 Open Problems

We have seen that Sperner's lemma becomes (at least in the oracular setting) strictly more powerful when applied to a non-orientable space; is there a more general transformation of PPAD-complete problems to PPA-complete versions? In particular, is some non-orientable variation of Brouwer's fixed-point theorem complete for PPA?

We would still like a natural complete problem for either PPA or PPAD that does not have an explicit Turing machine in the input. The most promising candidate still seems to be SMITH: given a Hamiltonian cycle in a graph with all nodes of odd degree, find another.

References

[1] C. Papadimitriou, On the complexity of the parity argument and other inefficient proofs of existence, J. Comput. System Sci. 48 (3) (1994) 498–532.

[2] D. S. Johnson, C. H. Papadimitriou, M. Yannakakis, How easy is local search?, J. Comput. System Sci. 37 (1) (1988) 79–100.

[3] N. Megiddo, C. H. Papadimitriou, On total functions, existence theorems and computational complexity, Theoretical Comput. Sci. 81 (2) (1991) 317–324.

[4] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, T. Pitassi, The relative complexity of NP search problems, J. Comput. System Sci. 57 (1) (1998) 3–19.

[5] J. J. Rotman, An Introduction to Algebraic Topology, Vol. 119 of Gradute Texts in Mathematics, Springer, 1988.