


The Hairy Ball Problem is PPAD-Complete

Paul W. Goldberg 

Department of Computer Science, University of Oxford, United Kingdom
Paul.Goldberg@cs.ox.ac.uk

Alexandros Hollender 

Department of Computer Science, University of Oxford, United Kingdom
alexandros.hollender@cs.ox.ac.uk

Abstract

The Hairy Ball Theorem states that every continuous tangent vector field on an even-dimensional sphere must have a zero. We prove that the associated computational problem of computing an approximate zero is PPAD-complete. We also give a FIXP-hardness result for the general exact computation problem.

In order to show that this problem lies in PPAD, we provide new results on multiple-source variants of END-OF-LINE, the canonical PPAD-complete problem. In particular, finding an approximate zero of a Hairy Ball vector field on an even-dimensional sphere reduces to a 2-source END-OF-LINE problem. If the domain is changed to be the torus of genus $g \geq 2$ instead (where the Hairy Ball Theorem also holds), then the problem reduces to a $2(g - 1)$ -source END-OF-LINE problem.

These multiple-source END-OF-LINE results are of independent interest and provide new tools for showing membership in PPAD. In particular, we use them to provide the first full proof of PPAD-completeness for the IMBALANCE problem defined by Beame et al. in 1998.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Computational Complexity, TFNP, PPAD, End-of-Line

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.60

Related Version A full version of the paper is available at <https://arxiv.org/abs/1902.07657>.

Funding *Alexandros Hollender*: Supported by an EPSRC doctoral studentship (Reference 1892947).

1 Introduction

The Hairy Ball Theorem (HBT) is a well-known topological theorem stating that there is no non-vanishing continuous tangent vector field on an even-dimensional k -sphere. It has various informal statements such as “you can’t comb a hairy ball flat without creating a cowlick”¹, or “there is a point on the surface of the earth with zero horizontal wind velocity”. The HBT is superficially reminiscent of the Borsuk-Ulam Theorem, stating that given any continuous mapping from the 2-sphere to the plane, there are two antipodal points that map to the same value. (Informally, “there are two antipodal points on the surface of the earth where the temperature and pressure are the same”). As we shall see, the present paper highlights a fundamental difference between the two, in terms of the complexity class naturally associated with each of them.

The HBT was first proved in 1885 by Poincaré [29] for the case $k = 2$. The theorem as stated for all even k was proved in 1912 by Brouwer [5]. Accordingly, this result is sometimes also called the Poincaré-Brouwer theorem. In fact, the result proved by Poincaré [29] is stronger than stated above. It follows from it that for any (sufficiently well-behaved) 2-dimensional manifold with genus $g \neq 1$, any continuous tangent vector field must have a zero.

¹ https://en.wikipedia.org/wiki/Hairy_ball_theorem



In particular, this means that the HBT also holds for the torus of genus g for $g \geq 2$, i.e. the 2-dimensional torus with g holes. It is easy to see that it does not hold for the standard single-hole torus.

Over the years, various papers in the *American Mathematical Monthly* have presented alternative proofs of the Hairy Ball Theorem and variants, for example [22, 26, 4, 14, 24, 9].

Topological existence results (such as the HBT, Borsuk-Ulam, and the Brouwer and Banach fixpoint theorems) have a very interesting relationship with complexity classes of search problems in which any instance has a guaranteed solution. Any such theorem has a corresponding computational challenge, of searching for such a solution, given a circuit that computes an appropriate function. The assumption that these complexity classes are distinct from each other (the ones of main interest here being PPAD and PPA, discussed below in more detail) provides a taxonomy of these theorems. Our results highlight a fundamental distinction between the HBT and Borsuk-Ulam, by showing that the corresponding search problem for the HBT is characterised by the complexity class PPAD, in contrast to Borsuk-Ulam, which is characterised by PPA [1]. The complexity-theoretic analysis of topological search problems provides a well-defined sense in which the HBT is “Brouwer-like” rather than “Borsuk-Ulam-like”. It has previously been noted that the HBT may be used to prove Brouwer’s fixed point theorem [26], but not the other way around.

1.1 Background on NP total search and PPAD

The complexity class TFNP is the set of all *total* function computation problems in NP: functions where *every* input has an efficiently-checkable solution (in Section 2.2 we give a precise definition). Many problems in TFNP appear to be computationally difficult, notably FACTORING, the problem of computing a prime factorisation of a given number, also NASH, the problem of computing a Nash equilibrium of a game. However, such problems are unlikely to be NP-hard, due to the 1991 result of Megiddo and Papadimitriou [25] showing that TFNP problems cannot be NP-hard unless NP is equal to co-NP. This basic fact, that hard TFNP problems are in a very strong sense “NP-intermediate”, provides TFNP’s strong theoretical appeal. This has led to the classification of these problems in terms of certain syntactic subclasses of TFNP, whose problems are shown to be total due to some basic combinatorial principle. The best-known of these classes are PLS, PPP, PPAD, and PPA, identified by Papadimitriou in 1994 [28].

- PPAD consists of problems whose totality is based on the principle that given a source in a directed graph whose vertices have in-degree and out-degree at most 1, there exists another degree-1 vertex. Its canonical problem END-OF-LINE consists of an exponentially-large graph of this kind, presented concisely via a circuit.
- PPA differs from PPAD in that the graph need not be directed; being a more general principle, PPA is thus a superset of PPAD. Its canonical problem LEAF is similar, only the graph is undirected.

Subsequently, many TFNP problems of interest were shown PPAD-complete [10, 7, 23, 13], while more recently others were shown PPA-complete [18, 17]. Despite their similar definitions, PPAD and PPA are usually conjectured to be different, and (along with other syntactic TFNP subclasses) are separated by oracles [3].

1.2 Our results and their significance

Given the long-standing interest in the Hairy Ball Theorem, it is natural to study the corresponding computational search problem. In this paper, we prove that computing

an approximate zero of a Hairy Ball vector field is PPAD-complete. While many PPAD-completeness results already exist, a noteworthy novelty of our results is that we find that computing HBT solutions corresponds with *multiple-source* variants of the END-OF-LINE problem: given a large directed graph implicitly represented by a circuit, suppose you are shown several sources and told to find another degree-1 vertex. This is in contrast with previous PPAD-complete problems that naturally reduce to standard single-source END-OF-LINE.

In Section 6 we prove that these multiple-source END-OF-LINE variants are PPAD-complete (membership of PPAD being the tricky aspect). Our results make progress on the general question (studied in [19]) of whether there exist combinatorial principles indicating totality of search problems, that are fundamentally different from the known ones that give rise to complexity classes such as PPAD. In particular, in Section 6.3, we note that a proof of PPAD-completeness for the IMBALANCE problem by Beame et al. [3] is incomplete and provide a full proof using our results.

The generalisation of Poincaré’s result to higher dimensions is called the Poincaré-Hopf theorem (see e.g. [20]). This theorem relates the number and types of zeros of a vector field on a manifold with its Euler characteristic, a topological invariant. In particular, if the Euler characteristic of a manifold is not 0, then any continuous tangent vector field on the surface must have a zero. The Euler characteristic of even-dimensional spheres is 2, while it is $2(1 - g)$ for 2-dimensional toruses of genus $g \geq 2$. For odd-dimensional spheres it is 0.

We believe that the reduction to multiple-source END-OF-LINE is not an artefact of our techniques, but instead intrinsically related to the Euler characteristic of the domain. Indeed, the reduction from the HBT problem on even-dimensional spheres to END-OF-LINE yields 2 sources (Section 4). On the other hand, if we consider the HBT problem on the 2-dimensional torus of genus $g \geq 2$, then we obtain $2(g - 1)$ sources (Section 7 of the full version). The connection between HBT and directed graph problems has previously only appeared in a proof for the 2-dimensional sphere case [22].

Finally, we note that PPAD-hardness is obtained by constructing a HBT vector field from multiple copies of a discrete Brouwer fixpoint problem. The usage of multiple copies is a new conceptual feature, closely related to the multi-source aspect. Using the same high-level idea, we also provide a FIXP-hardness result for the problem of computing an *exact* solution (Section 5.2).

1.3 Other related work

Banach’s Fixed Point Theorem [2] says that a *contraction map* has a unique fixpoint. Its corresponding computational problem CONTRACTION, is to find a fixed point of a given contraction map. Some versions of CONTRACTION have been shown complete for CLS, a subclass of PPAD [11, 12, 16]. The search for Brouwer fixpoints (including discretised versions of Brouwer functions) is PPAD-complete for most variants of the problem [28, 7], which is why we say the HBT is “Brouwer-like”. Finally — in contrast — the computational problem of searching for a Borsuk-Ulam solution is PPA-complete [1]. Other topological existence results that have PPA-complete search problems include the Hobby-Rice theorem [18] and the Ham Sandwich Theorem [17].

2 Preliminaries

Let k be a positive integer. For $x \in \mathbb{R}^k$, $\|x\|_2$, $\|x\|_1$ and $\|x\|_\infty$ denote the standard ℓ_2 -norm, ℓ_1 -norm and ℓ_∞ -norm respectively. For $x, y \in \mathbb{R}^k$, $\langle x, y \rangle := \sum_{i=1}^m x_i y_i$ denotes the inner

product.

The k -dimensional unit sphere in \mathbb{R}^{k+1} (or k -sphere) is denoted $S^k = \{x \in \mathbb{R}^{k+1} : \|x\|_2 = 1\}$. A continuous tangent vector field on S^k is a continuous function $f : S^k \rightarrow \mathbb{R}^{k+1}$ such that for all $x \in S^k$ we have $\langle f(x), x \rangle = 0$. The Hairy Ball Theorem can be stated as follows:

► **Theorem 1** (Poincaré [29]–Brouwer [5]). *If $k \geq 2$ is even, then for any continuous tangent vector field $f : S^k \rightarrow \mathbb{R}^{k+1}$, there exists $x \in S^k$ such that $f(x) = 0$.*

2.1 Model of Computation

We work in the standard Turing machine model. All numbers appearing in computations are rational numbers where the numerator and denominator are integers represented in binary. For a rational number x , $\text{size}(x)$ denotes the size of the representation of x , i.e. the sum of the representation length of its numerator and denominator in binary. For an arithmetic circuit F , $\text{size}(F)$ denotes the number of gates in the circuit added to the representation length of any rational constants used by the circuit.

2.2 Formal definition of TFNP

A computational search problem is given by a binary relation $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$, interpreted as follows: $y \in \{0, 1\}^*$ is a solution to instance $x \in \{0, 1\}^*$, if and only if $(x, y) \in R$. The search problem R is in FNP (*Functions in NP*), if R is polynomial-time computable (i.e. $(x, y) \in R$ can be decided in polynomial time in $|x| + |y|$) and there exists some polynomial p such that $(x, y) \in R \implies |y| \leq p(|x|)$. Here $\{0, 1\}^*$ denotes all finite length bit-strings and $|x|$ is the length of bit-string x .

The class TFNP (*Total Functions in NP* [25]) contains all search problems R that are in FNP and are *total*, i.e. every instance has at least one solution. Formally, this corresponds to requiring that for every $x \in \{0, 1\}^*$ there exists $y \in \{0, 1\}^*$ such that $(x, y) \in R$.

Let R and S be total search problems in TFNP. We say that R (many-one) reduces to S , if there exist polynomial-time computable functions f, g such that

$$(f(x), y) \in S \implies (x, g(x, y)) \in R.$$

Note that if S is polynomial-time solvable, then so is R . We say that two problems R and S are (polynomial-time) equivalent, if R reduces to S and S reduces to R .

To be PPAD-complete, a problem must be equivalent to END-OF-LINE (Definition 12); in Section 6 we show that the multiple-source version MS-EOL (Definition 13) is equivalent.

3 The Hairy-Ball Problem

3.1 The k D-Hairy-Ball problem

The Hairy Ball Theorem naturally yields a corresponding computational problem. We are given a continuous tangent vector field f on the unit sphere and have to find a point where it is zero. In trying to formalise this, some issues need to be addressed. First, one has to decide how the vector field should be represented in the input. Here we take the usual approach of assuming that it is represented as an arithmetic circuit.

Before we discuss the types of gates that we want to allow in the circuit, let us briefly handle the second issue: the vector field might not have a rational zero. Indeed, consider the following example: at $x \in S^2$ the vector field is simply the vector $(1, 1, 1)$ projected onto the tangent space of S^2 at x . In this case, the only solutions are $\pm(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$. Thus,

we cannot hope to always output an exact solution. We bypass this problem by asking for an *approximate* solution instead, i.e. a point $x \in S^2$ such that $\|f(x)\|_\infty \leq \varepsilon$ for some $\varepsilon > 0$ provided in the input. This notion of approximate solution is the standard one used when studying topological existence theorems in the context of TFNP (e.g. Brouwer's fixed point theorem or the Borsuk-Ulam theorem).

As mentioned above, the vector field will be represented as an arithmetic circuit. In the case of S^2 , the circuit will have three input gates and three output gates. The arithmetic circuit will be allowed to use gates $\{+, -, \times \zeta, \max, \min\}$ and rational constants. All the gates have fan-in 2, except $\times \zeta$ which has fan-in 1 and corresponds to multiplication by a rational constant ζ . Note that such a circuit is polynomially equivalent to a circuit only using gates $\{+, \times \zeta, \max\}$ and rational constants, since the other gates can be efficiently simulated using these. These circuits correspond to LINEAR-FIXP-type circuits that are known to be sufficient to obtain PPAD-hardness of BROUWER [15]. A discussion about why we don't use more powerful gates in our definition can be found in Section 3.2 of the full version.

This type of circuit yields piece-wise affine functions that are continuous. Furthermore, it has the following nice property: for any such arithmetic circuit F , and any rational x , we can compute $F(x)$ exactly in polynomial time in $\text{size}(F)$ and $\text{size}(x)$. One potential issue is that F might not be *tangent* to the sphere, but this is easy to fix by simply considering the vector field given by the projection of F onto the corresponding tangent space to the sphere. Thus, we define the computational problem as follows:

► **Definition 2** (k D-HAIRY-BALL). *Let $k \geq 2$ be even. The k D-HAIRY-BALL problem is defined as: given $\varepsilon > 0$ and an arithmetic circuit F with $k + 1$ inputs and outputs, using gates $\{+, \times \zeta, \max\}$ and rational constants, find $x \in S^k$ such that $\|P_x[F(x)]\|_\infty \leq \varepsilon$.*

Here $P_x[\cdot]$ denotes the projection onto the tangent space to the sphere S^k at $x \in S^k$. Note that for any $v \in \mathbb{R}^{k+1}$, we have $P_x[v] = v - \langle v, x \rangle x$, because $\|x\|_2 = 1$. Thus, the projection of any rational vector v onto the tangent space at rational $x \in S^k$ can be computed exactly in polynomial time in $\text{size}(v)$ and $\text{size}(x)$. Note that we are looking for a solution with respect to the ℓ_∞ -norm, but we could also have used the ℓ_2 - or ℓ_1 -norm, since all these versions are computationally equivalent.

k D-HAIRY-BALL lies in TFNP. Clearly, any solution can be checked in polynomial time. Totality of k D-HAIRY-BALL will immediately follow when we prove that it lies in PPAD (Corollary 8). The following Lemma is proved in Section 3.1 of the full version.

► **Lemma 3.** *Let $k \geq 2$ be even. Let F be an arithmetic circuit with $k + 1$ inputs and outputs, using gates $\{+, \times \zeta, \max\}$ and rational constants. Then, the function $S^k \rightarrow \mathbb{R}^{k+1}$, $x \mapsto P_x[F(x)]$ is Lipschitz-continuous with Lipschitz constant $L = k \cdot 2^{\text{size}(F)^2+3}$ (w.r.t. ℓ_∞ -norm).*

Our main result is Theorem 4. Containment in PPAD, which turns out to be the most challenging part of this result, is presented in Section 4 (using the multiple-source END-OF-LINE results of Section 6). PPAD-hardness is presented in Section 5.

► **Theorem 4.** *For all even $k \geq 2$, k D-HAIRY-BALL is PPAD-complete.*

4 The Hairy-Ball Problem is in PPAD

In this section we present our main result: the problem of computing an approximate Hairy Ball solution reduces to END-OF-LINE, the canonical PPAD-complete problem.

From a purely mathematical standpoint, our proof can be used to provide a (fairly cumbersome) proof of the Hairy Ball theorem by using Brouwer's fixed point theorem. Indeed, it is known [28] that END-OF-LINE reduces to BROUWER (in fact, even 2D-BROUWER [7]). Thus, given a Hairy Ball function f , using our reduction and Brouwer's fixed point theorem, one can prove the existence of a point x_k such that $\|f(x_k)\| \leq 1/2^k$ for any k (using the fact that f must be uniformly continuous since the sphere is compact). Then, since any sequence in a compact set must have a converging subsequence it follows that there must exist x such that $f(x) = 0$. Finding a more direct way to deduce the Hairy Ball theorem from Brouwer's fixed point theorem is an interesting open question.

Our goal is to prove that the Hairy Ball problem lies in PPAD in a setting that is as general and encompassing as possible. The way the function is represented, as a circuit or otherwise, should not play a role. Thus, we are only going to make two assumptions about the tangent vector field: that it can be evaluated in polynomial time and that it is polynomially continuous in some well-defined sense. The first assumption is very natural: if we are given a Hairy Ball function, we expect to be able to evaluate it efficiently. The motivation for the second assumption is that if we omit it, then there is no guarantee that there will exist an approximate solution with representation size that is polynomial in the input size.

We now define these assumptions formally, following the analogous definitions by Etessami and Yannakakis [15] for Brouwer fixed point problems. Let \mathcal{F} be a class of Hairy Ball functions $f : S^k \rightarrow \mathbb{R}^{k+1}$ (i.e. continuous tangent vector fields) with $k \geq 2$ even. Note that here k is not fixed for all $f \in \mathcal{F}$, but we assume that $k \leq \text{size}(f)$. For any $f \in \mathcal{F}$, $\text{size}(f)$ denotes the length of the representation of f in \mathcal{F} . In the case of k D-HAIRY-BALL, k is fixed and \mathcal{F} is the class of all such functions represented using arithmetic circuits with gates $\{+, \times, \zeta, \max\}$ (with the projection onto the tangent space at the end). In that case, $\text{size}(f)$ is the size of the circuit representing f . Recall that for rational vector x , $\text{size}(x)$ is the length of the representation of x .

► **Definition 5** ([15]). *Let \mathcal{F} be a class of Hairy Ball functions.*

- \mathcal{F} is polynomially computable, if there exists some polynomial p such that for any $f \in \mathcal{F}$ and any rational input $x \in S^k$, $f(x)$ can be computed in time $p(\text{size}(f) + \text{size}(x))$.
- \mathcal{F} is polynomially continuous, if there exists some polynomial q such that for any $f \in \mathcal{F}$ and any rational $\varepsilon > 0$, there exists a rational $\delta > 0$ with $\text{size}(\delta) \leq q(\text{size}(f) + \text{size}(\varepsilon))$ such that for all $x, y \in S^k$ we have $\|x - y\|_\infty \leq \delta \implies \|f(x) - f(y)\|_\infty \leq \varepsilon$.

Note that k D-HAIRY-BALL yields a class \mathcal{F} that is both polynomially computable and polynomially continuous (by Lemma 3).

► **Definition 6.** *Let \mathcal{F} be a class of Hairy Ball functions. The problem $\text{HAIRY-BALL}(\mathcal{F})$ is defined as: given $f \in \mathcal{F}$ and $\varepsilon > 0$, find $x \in S^k$ such that $\|f(x)\|_\infty \leq \varepsilon$.*

For simplicity we assume that we can recognise whether some string in $\{0, 1\}^*$ represents an element $f \in \mathcal{F}$ in polynomial time. If this does not hold, then $\text{HAIRY-BALL}(\mathcal{F})$ has to be studied as a *promise* problem. The reduction to END-OF-LINE given in the proof below still holds. However, this does not imply that the problem lies in PPAD, because TFNP requires the problem to be total without any promise.

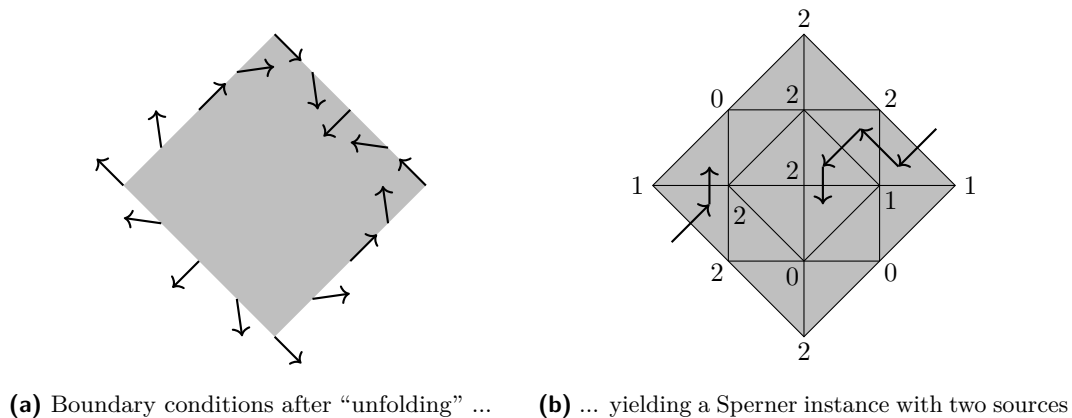
► **Theorem 7.** *Let \mathcal{F} be a class of Hairy Ball functions that is polynomially computable and polynomially continuous. Then, $\text{HAIRY-BALL}(\mathcal{F})$ lies in PPAD.*

► **Corollary 8.** *For all even $k \geq 2$, k D-HAIRY-BALL lies in PPAD.*

Proof Overview for Theorem 7. The proof can be subdivided into two parts. In the first part, we reduce k D-HAIRY-BALL to a 2-source END-OF-LINE problem. In the second part, we show that the 2-source version reduces to the standard version of END-OF-LINE, where a single source is known. Surprisingly, the reduction from 2 sources to 1 is non-trivial. The proof for this is presented separately in Section 6. In fact, we prove the more general result: as long as the number of known sources in an END-OF-LINE instance is polynomial, we can reduce to standard END-OF-LINE. Various implications of this result are also presented in Section 6.

We now give some details about the first part of the proof, in which we reduce k D-HAIRY-BALL to 2-source END-OF-LINE through a Sperner argument. The inspiration for this comes from a proof of the 2-dimensional Hairy Ball Theorem via a version of Sperner’s Lemma, given by Jarvis and Tanton [22]. Our contribution here is two-fold: we extend their proof to any higher (even) dimension and we turn it into a polynomial-time reduction.

In order to obtain a polynomial-time reduction, instead of working directly on the sphere, we use a stereographic projection to “unfold” the sphere $S^k (\subset \mathbb{R}^{k+1})$ into the space \mathbb{R}^k , along with the vector field. Then, we consider a sufficiently large cross-polytope C of \mathbb{R}^k and prove that the “unfolded” vector field satisfies certain boundary conditions. In the case $k = 2$, this corresponds to the vector field making two full rotations when we move along the boundary of C (see Figure 1a). Next, we pick an efficient triangulation of C and a suitable colouring of its nodes. The last step then requires us to prove that this colouring yields exactly two starting points (on the boundary) for Sperner paths (see Figure 1b) that lead to *panchromatic* simplices. Using standard Sperner-arguments this yields a 2-source END-OF-LINE instance. The full proof for any even k can be found in Section 4.2 of the full version. Note that, as expected, the proof does not work if k is odd. Indeed, the construction then yields a starting point and an ending point on the boundary, instead of two starting points. ◀



■ **Figure 1** An example for the proof of Theorem 7 in the case $k = 2$. The region C is represented in grey.

5 Computational Hardness for Hairy Ball Problems

It is possible to prove Brouwer’s fixed point theorem using the Hairy Ball Theorem as follows (see [26] for the full details). Let $B_k \subset \mathbb{R}^k$ be the unit ball. If we assume that a function

$f : B_k \rightarrow B_k$ does not have any fixed point, then we can construct a Hairy Ball function $g : S^k \rightarrow \mathbb{R}^k$ that does not have a zero. Brouwer's theorem follows by contradiction. The main idea for the construction of g is the following. Consider $f'(x) = f(x) - x$ and assume that it points directly inward on the boundary of B_k . Take one copy of B_k with the vector field f' and one copy with the vector field $-f'$, and glue their boundaries together. The resulting object can be deformed to yield the sphere S^k and the vector fields will perfectly match on the glued region. Thus, assuming that f' has no zero, g will have no zero either.

By making this idea fully constructive and efficient, we obtain reductions from Brouwer problems to Hairy Ball problems. Thus, existing PPAD- and FIXP-hardness results for Brouwer also hold for the corresponding Hairy Ball problems. We note that these reductions always involve using *two* copies of a Brouwer instance to obtain a single Hairy Ball instance. This further supports our claim that the fact that we obtain two sources when reducing k D-HAIRY-BALL to END-OF-LINE (Section 4) is not an artefact of our reduction, but intrinsic to the problem.

5.1 PPAD-hardness

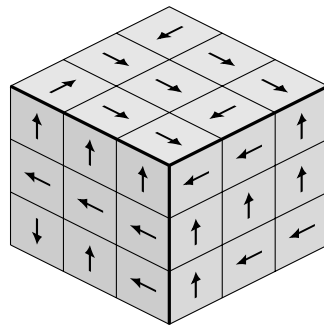
► **Theorem 9.** *For all even $k \geq 2$, k D-HAIRY-BALL is PPAD-hard.*

Note that this result is particularly strong, because the type of circuit allowed in the definition of k D-HAIRY-BALL (Definition 2) is particularly weak. Furthermore, the hardness is proved for inversely exponential ε/L (where L is the Lipschitz constant of the function), which is the best we can hope for in the fixed dimension case. Indeed, if ε/L is inversely polynomial, then the following is a polynomial time algorithm that solves the problem: divide the domain into a sufficiently small (but polynomial) number of regions and check an arbitrary point in each region. In the case where the dimension is not fixed, it seems likely that the problem should be hard even for constant ε/L , but we do not investigate this in this paper.

Proof Overview. One way to prove this result is to take an instance F of a 2D-BROUWER problem, which is known to be PPAD-complete [7], and embed a copy F and a copy $-F$ on the south and north hemisphere of S^2 respectively. However, since our 2D-HAIRY-BALL circuit can only use gates in $\{+, \times, \zeta, \max\}$, we first shrink the domain of F so that we embed it in a small region around the south pole. This ensures that even after projection onto the tangent space, no bogus solutions will appear. We do the same for $-F$ in the north pole and then define G on the rest of the sphere in such a way that no solution appears there.

In spirit, we follow this general proof idea, but take a slight detour, because it gives us the chance to define and study a discrete analog to 2D-HAIRY-BALL: the 2D-HAIRY-CUBE problem. Intuitively, this problem is obtained from 2D-HAIRY-BALL the same way that discrete 2D-BROUWER is obtained from continuous 2D-BROUWER. The domain is discretised by a grid and a circuit computes the local direction of the function in every grid-square. The natural way to discretise the sphere S^2 is to replace it by a cube with a grid on each face. The advantage of 2D-HAIRY-CUBE is that PPAD-hardness is easy to prove: just put a (slightly modified) discrete 2D-BROUWER instance on one face, and the inverse instance on the opposite face. Defining the instance on the remaining faces is trivial in this case. In Section 6 of the full version we define the problem and prove PPAD-hardness. An illustration of a 2D-HAIRY-CUBE instance is given in Figure 2.

The next step is reducing 2D-HAIRY-CUBE to 2D-HAIRY-BALL. Even though it seems natural that this should hold, the reduction is technically tedious. In particular, we have to simulate a Boolean circuit using an arithmetic circuit, but the input bits cannot always be



■ **Figure 2** Partial view of a 2D-HAIRY-CUBE instance.

computed exactly. Thus, we need to use an averaging trick (see [10, 8]). The details are in Appendix A of the full version.

This yields PPAD-hardness for 2D-HAIRY-BALL. The final step is to extend this to k D-HAIRY-BALL, by showing that k D-HAIRY-BALL reduces to $(k + 2)$ D-HAIRY-BALL. The proof can be found in Appendix B of the full version.

◀

5.2 FIXP-hardness

Up to this point we have only considered the problem of computing an *approximate* Hairy Ball solution. However, there are other computational problems one could consider, e.g. computing a point that is close to an *exact* solution, or computing the first n bits of an *exact* solution.

The corresponding problems for Brouwer fixed points have been studied by Etessami and Yannakakis [15]. In particular, they define the class FIXP that captures the complexity of computing an exact fixed point of a function given by an arithmetic circuit and mapping the unit cube into itself. They prove that computing an exact Nash equilibrium of a 3-player game is FIXP-complete. In doing so, they use a special type of reduction, called an SL-reduction, that ensures that the reduction also holds for three problems that can be studied in the standard Turing machine model: the “strong approximation problem” (i.e. find a point close to an exact solution), the “partial computation” problem (i.e. compute the first n bits of an exact solution) and various decision problems. This means that computing a strong approximation for 3-player Nash is as hard as computing a strong approximation of a Brouwer function given by an arithmetic circuit. We prove that the corresponding problems for the Hairy Ball Theorem are at least as hard as their Brouwer counterparts.

► **Definition 10.** *The EXACT-HAIRY-BALL problem is defined as: given an arithmetic circuit F (with gates $\{+, -, \times, /, \max, \min\}$, rational constants and that never divides by 0) that computes a tangent vector field $S^k \rightarrow \mathbb{R}^{k+1}$, k even, find $x \in S^k$ such that $F(x) = 0$.*

Note that the vector field will be continuous since we never divide by 0. The vector field can be syntactically forced to be tangent to the sphere because we can compute the projection exactly using this kind of circuit.

► **Theorem 11.** *EXACT-HAIRY-BALL is FIXP-hard. Furthermore, the corresponding strong approximation, partial computation and decision problems are also hard for the corresponding versions of FIXP (as defined in [15]).*

Proof Overview. We embed two copies of a Brouwer instance (after some preprocessing) on the sphere such that any exact Hairy Ball solution yields an exact Brouwer fixed point. The details can be found in Section 5.2 of the full version. Note that this reduction makes use of the \times and $/$ gates and cannot be used to prove PPAD-hardness of k D-HAIRY-BALL. ◀

Following Etessami and Yannakakis, we could define a class HB that captures the complexity of computing an exact Hairy Ball solution (and corresponding versions of the class for the related problems) by taking the set of all problems that reduce to EXACT-HAIRY-BALL. Then, Theorem 11 is saying that $\text{FIXP} \subseteq \text{HB}$. Note that the three discrete problems that can be studied in the Turing machine model lie in PSPACE, by using the same technique as in [15, Proposition 4.2].

6 End-of-Line: One Source to Rule Them All

END-OF-LINE is the canonical problem used to define PPAD. Investigating variants of the problem is of independent interest, in particular in order to gain a better understanding of PPAD and how it relates to other similar subclasses of TFNP. An additional motivation for studying these variants is given by this paper, since a multiple-source variant of END-OF-LINE is used to prove that finding an approximate Hairy Ball solution lies in PPAD (Section 4).

This section is an improved version of the corresponding content in the technical report [21]. In [21] we use these results to show that a computational problem related to the Mutilated Chessboard puzzle is PPAD-complete.

6.1 The End-of-Line problem

The END-OF-LINE problem is informally defined as follows: given a directed graph where each vertex has in- and out-degree at most 1 and given a known source of this graph, find a sink or another source. The problem is computationally challenging, because the graph is not given explicitly in the input. Instead, we are given an implicit concise representation of the graph through circuits that compute the predecessor and successor of a vertex in the graph. In what follows, we sometimes interpret the input and output of the circuits, which are elements in $\{0, 1\}^n$, as the numbers $\{0, 1, \dots, 2^n - 1\}$.

► **Definition 12** (END-OF-LINE [10]). *The END-OF-LINE problem is defined as: given Boolean circuits S, P with n input bits and n output bits and such that $P(0) = 0 \neq S(0)$, find x such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0$.*

The circuits define a graph as follows. There is a directed edge from vertex x to y ($x \neq y$), if and only if $S(x) = y$ and $P(y) = x$. Note that any badly defined edge, i.e. $S(x) = y$ and $P(y) \neq x$, or $P(y) = x$ and $S(x) \neq y$, qualifies as a solution of END-OF-LINE as defined above (because $P(S(x)) \neq x$ or $S(P(x)) \neq x$ respectively). Note that 0 is a source of the graph, unless $P(S(0)) \neq 0$, in which case 0 is a valid solution to the problem as stated above.

It is easy to check that this formal definition of the problem is computationally equivalent to the informal description given above. By definition, END-OF-LINE is PPAD-complete [28]. Furthermore, reduction from END-OF-LINE is a very common technique to show PPAD-hardness (e.g. [10, 7]).

6.2 Multiple-Source End-of-Line

What if, instead of just one, we already know *two* sources of an END-OF-LINE instance? We are still interested in finding any sink or any *other* source. Intuitively, the problem might

seem easier, because the existence of two sources implies the existence of at least two sinks, hence more potential solutions. In fact, it is easy to see that this problem is actually at least as hard as END-OF-LINE: just duplicate the whole END-OF-LINE instance.

The other direction, however, is not trivial. Indeed, if we interpret our 2-source END-OF-LINE instance as a standard END-OF-LINE instance (and pick one of the two sources as the standard source), then the other known source is a valid solution to END-OF-LINE, but not a valid solution to our original problem. In other words, it is not clear how to solve this problem if we are given access to an oracle solving END-OF-LINE, because the oracle could just return the other known source. We consider the following more general problem, where we are given an END-OF-LINE graph and an explicit list of known sources.

► **Definition 13** (MS-EOL). *The MULTIPLE-SOURCE END-OF-LINE problem, abbreviated MS-EOL, is defined as: given circuits S, P with n inputs and n outputs and $s_1, \dots, s_k \in \{0, 1\}^n$ such that $P(s_i) = s_i \neq S(s_i)$ for all i , find $x \in \{0, 1\}^n$ such that $P(S(x)) \neq x$ or $x \notin \{s_1, \dots, s_k\}$ such that $S(P(x)) \neq x$.*

In passing, let us note that in the undirected case this kind of generalisation is trivial. The undirected analogue of END-OF-LINE is LEAF: given an undirected graph where every vertex has degree at most 2 and given a vertex of degree 1, find another vertex of degree 1, i.e. another leaf. Assume that we know k leaves instead of just one. If k is even, then the problem is not even in TFNP. If k is odd, then we can add edges between known leaves until exactly one is left. Thus, the problem is equivalent to LEAF. This kind of reduction does not work for the directed case. Nevertheless, we obtain²:

► **Theorem 14.** MULTIPLE-SOURCE END-OF-LINE is equivalent to END-OF-LINE.

Proof Overview. We inductively construct a vertex set containing various combinations of subsets of the original graph. The successor and predecessor circuits are carefully constructed so as to ensure that any solution yields a solution to the original problem. The full proof can be found in Section 8.2 of the full version. ◀

► **Remark** (Multiple Known Sources and Sinks). A natural generalisation of MULTIPLE-SOURCE END-OF-LINE is the following problem: given an END-OF-LINE graph and a list of k known sources and m known sinks, find another source or sink. Note that for this problem to be in TFNP, we must require $k \neq m$. Using Theorem 14, it is easy to see that this problem is equivalent to END-OF-LINE. If $k > m$, then we add an edge from each of the m known sinks to some corresponding known source and obtain an instance with $k - m$ known sources and no known sinks. Similarly, if $k < m$, then we first reverse all directed edges and then apply the same trick.

The next two sections present additional consequences of Theorem 14.

6.3 The Imbalance problem

Up to this point, we have only considered graphs where every vertex has in- and out-degree at most 1. However, the principle that guarantees the existence of a solution in an END-OF-LINE graph can be generalised to higher degree graphs. If we are given a directed graph and an *unbalanced* vertex, i.e. a vertex with in-degree \neq out-degree, then there must exist another unbalanced vertex.

² This problem was discussed in an online thread (<https://cstheory.stackexchange.com/q/37481>). E. Jeřábek proved membership in PPADS and PPA- p for every prime p (but not membership in PPAD).

Beame et al. [3] defined the corresponding problem IMBALANCE, which is seemingly more general than END-OF-LINE. In this problem, every vertex is not constrained to have in- and out-degree at most 1. Instead, in- and out-degree are bounded by some polynomial of the input length³. We are given a vertex that is unbalanced and have to find another unbalanced vertex (which is guaranteed to exist). The problem can be informally defined as follows:

► **Definition 15** (IMBALANCE [3], informal). *The IMBALANCE problem is defined as: given a directed graph (represented concisely by predecessor and successor functions) and a vertex z that has in-degree \neq out-degree, find a vertex $x \neq z$ that also has in-degree \neq out-degree.*

Beame et al. [3] claim that IMBALANCE reduces to END-OF-LINE, using the same argument as for the corresponding problems on undirected graphs. However, if the graph is directed, a complication arises (that is not an issue in the undirected case). Indeed, their proof idea is incomplete, because they overlook the fact that their reduction yields an END-OF-LINE instance with *multiple* known sources. Using Theorem 14 we can provide a full proof of their claim.

► **Theorem 16.** *IMBALANCE is PPAD-complete.*

A formal definition of the problem and a full proof can be found in Section 8.3 of the full version.

6.4 Looking for multiple solutions

If we are given an END-OF-LINE instance with k known sources, then we can ask for k sinks or k unknown sources. The problem is total, because at least k sinks are guaranteed to exist.

► **Definition 17** (k -EOL). *Let $k \in \mathbb{N}$. The k -ENDS-OF-LINE problem, abbreviated k -EOL, is defined as: given circuits S, P with n inputs and n outputs and such that $P(z) = z \neq S(z)$ for all $z < k$, find distinct x_1, \dots, x_k such that $P(S(x_i)) \neq x_i$ for all i or $S(P(x_i)) \neq x_i \geq k$ for all i .*

Intuitively, this problem seems harder than END-OF-LINE or MS-EOL, because we are now looking for more than one solution. However, using Theorem 14 we can show:

► **Theorem 18.** *For any $k \in \mathbb{N}$, k -ENDS-OF-LINE is PPAD-complete.*

Proof Overview. The reduction to END-OF-LINE is obtained by constructing a Turing reduction (using Theorem 14) and then applying a result by Buss and Johnson [6] that PPAD is closed under Turing reductions. The details are in Section 8.4 of the full version. ◀

► **Remark.** The same proof also yields PPAD-completeness for the following problem. Fix some polynomial p . The problem is: given k sources, find k sinks or $p(k)$ sources. This seems quite surprising, as one might have expected this problem to be closer to PPADS.

Some analogous results for the class PPADS and its canonical complete problem SINK [27, 3] are presented in Section 8.4 of the full version. SINK is identical to END-OF-LINE, except that we only accept a sink as a solution and are not interested in other sources. In this case the results are easier to obtain, because there is no need for an analogue of Theorem 14.

³ Note that this trivially holds, if the input consists of circuits that explicitly output the predecessor and successor list.

7 Conclusion and Further Work

We have obtained a satisfying answer to the question of the computational complexity of the Hairy Ball Theorem, if we are looking for an approximate solution. For other solution concepts related to exact solutions, we have provided a FIXP-hardness result. This leaves open the question of whether the problem is FIXP-complete in this case. Indeed, our reduction from Hairy Ball to Brouwer only works for approximate solutions. A first step would be to try to reduce Hairy Ball to Borsuk-Ulam, even though no such (fully constructive) mathematical proof seems to be known.

Our results on multiple-source variants of the END-OF-LINE problem open the way for two new research directions. First, they provide a new tool for showing membership of PPAD, which can be used to put further problems in this class. It seems very unlikely that the Hairy Ball Theorem should be the only “natural” application of these results. Furthermore, a second interesting research direction is investigating the complexity of END-OF-LINE with a super-polynomial number of known sources (implicitly given in the input).

References

- 1 James Aisenberg, Maria Luisa Bonet, and Sam Buss. 2-D Tucker is PPA complete. Technical Report TR15-163, ECCC, 2015.
- 2 Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3(1):133–181, 1922.
- 3 Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998.
- 4 W. M. Boothby. On two classical theorems of algebraic topology. *The American Mathematical Monthly*, 78(3):237–249, 1971.
- 5 L. E. J. Brouwer. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1912.
- 6 Samuel R. Buss and Alan S. Johnson. Propositional proofs and reductions between NP search problems. *Annals of Pure and Applied Logic*, 163(9):1163–1182, 2012.
- 7 Xi Chen and Xiaotie Deng. On the complexity of 2d discrete fixed point problem. *Theoretical Computer Science*, 410(44):4448–4456, 2009.
- 8 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):1–57, 2009.
- 9 Eugene Curtin. Another short proof of the hairy ball theorem. *The American Mathematical Monthly*, 125(5):462–463, 2018.
- 10 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.
- 11 Constantinos Daskalakis and Christos H. Papadimitriou. Continuous local search. In *22nd SODA*, pages 790–804. SIAM, 2011.
- 12 Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. A converse to Banach’s fixed point theorem and its CLS-completeness. In *50th STOC*, pages 44–50. ACM, 2018.
- 13 Xiaotie Deng, Qi Qi, and Amin Saberi. Algorithmic solutions for envy-free cake cutting. *Operations Research*, 60(6):1461–1476, 2012.
- 14 Murray Eisenberg and Robert Guy. A proof of the hairy ball theorem. *The American Mathematical Monthly*, 86(7):571–574, 1979.
- 15 Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.
- 16 John Fearnley, Spencer Gordon, Ruta Mehta, and Rahul Savani. End of potential line. *arXiv preprint arXiv:1804.03450*, 2018.

- 17 Aris Filos-Ratsikas and Paul W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. *arXiv preprint arXiv:1805.12559*, 2018. (to appear at STOC 2019).
- 18 Aris Filos-Ratsikas and Paul W. Goldberg. Consensus halving is PPA-complete. In *50th STOC*, pages 51–64. ACM, 2018.
- 19 Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions. *Journal of Computer and System Sciences*, 94:167–192, 2018.
- 20 Victor Guillemin and Alan Pollack. *Differential topology*. Prentice-Hall, 1974.
- 21 Alexandros Hollender and Paul W. Goldberg. The complexity of multi-source variants of the End-of-Line problem, and the concise mutilated chessboard. Technical Report TR18-120, ECCC, 2018.
- 22 Tyler Jarvis and James Tanton. The hairy ball theorem via Sperner’s lemma. *American Mathematical Monthly*, pages 599–603, 2004.
- 23 Shiva Kintali, Laura J. Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. Reducibility among fractional stability problems. *SIAM Journal on Computing*, 42(6):2063–2113, 2013.
- 24 Peter McGrath. An extremely short proof of the hairy ball theorem. *The American Mathematical Monthly*, 123(5):502–503, 2016.
- 25 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.
- 26 John Milnor. Analytic proofs of the “hairy ball theorem” and the Brouwer fixed point theorem. *The American Mathematical Monthly*, 85(7):521–524, 1978.
- 27 Christos H. Papadimitriou. On graph-theoretic lemmata and complexity classes. In *31st FOCS*, pages 794–801. IEEE, 1990.
- 28 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
- 29 Henri Poincaré. Sur les courbes définies par les équations différentielles. *Journal de Mathématiques Pures et Appliquées*, 4:167–244, 1885.