Locally 2-dimensional Sperner Problems Complete for the Polynomial Parity Argument Classes *

Katalin Friedl[1], Gábor Ivanyos[2], Miklos Santha[3], and Yves F. Verhoeven[3,4]

[1] BME, H-1521 Budapest, P.O.Box 91., Hungary
friedl@cs.bme.hu
[2] MTA SZTAKI, H-1518 Budapest, P.O. Box 63., Hungary
Gabor.Ivanyos@sztaki.hu
[3] CNRS–LRI, UMR 8623, bâtiment 490, Université Paris XI, 91405 Orsay, France
santha@lri.fr
[4] ENST, 46 rue Barrault, 75013 Paris, France
yves.verhoeven@normalesup.org

Abstract. In this paper, we define three Sperner problems on specific surfaces and prove that they are complete respectively for the classes PPAD, PPADS and PPA. This is the first time that locally 2-dimensional Sperner problems are proved to be complete for any of the polynomial parity argument classes.

1 Introduction

The complexity class TFNP, the family of all total NP-search problems, was introduced by Megiddo and Papadimitriou [9]. It contains several important, computationally probably hard problems for which no classical polynomial time algorithms are known. On the other hand, these problems are also somewhat easy in the sense that they can not be NP-hard unless NP = co-NP. The class TFNP is a semantic complexity class and thus doesn't seem to have complete problems. It is therefore natural to look for syntactically definable subclasses of TFNP. Indeed, several such subclasses have been identified along the lines of the mathematical proofs establishing the existence of a solution. The important subclasses Polynomial Pigeonhole Principle (PPP) and Polynomial Local Search (PLS) were defined respectively in [12] and [7]. The elements of PPP are problems which by their combinatorial nature obey the pigeonhole principle and therefore have a solution. In a PLS problem, one is looking for a local optimum for a particular objective function, in some easily computable neighborhood structure.

The parity argument subclasses PPA, PPAD, and PSK of TFNP were defined by Papadimitriou in [11, 12]. The class PSK was renamed PPADS in [1]. These

* Research supported by the EU 5th framework programs RESQ IST-2001-37559, Centre of Excellence ICAI-CT-2000-70025, the EU 6th framework program QAP, the OTKA grants T42559, T46234, and by the ACI CR 2002-40, ACI SI 2003-24, ANR Blanc AlgoQP grants of the French Research Ministry.

classes can be characterized by some simple graph theoretical principles. The class Polynomial Parity Argument (PPA) is the class of NP search problems, where the existence of the solution is guaranteed by the fact that in every finite graph whose vertices are of degree at most two, the number of leaves is even. The class PPAD is the directed version of PPA, and its basic search problem is the following: in a directed graph, where the in-degree and the out-degree of every vertex is at most one, given a source, find another source or a sink. In the class PPADS the basic search problem is more restricted than in PPAD: given a source, find a sink.

Another point that makes the parity argument classes interesting is that there are several natural problems from different branches of mathematics that belong to them. For example, in a graph with odd degrees, when a Hamiltonian path is given, a theorem of Smith [15] ensures that there is another Hamiltonian path. It turns out that finding this second path belongs to the class PPA [12]. A search problem coming from a modulo 2 version of Chevalley's theorem [12] from number theory is also in PPA. Complete problems in PPAD are the search versions of Brouwer's fixed point theorem, Kakutani's fixed point theorem, Borsuk-Ulam theorem, and Nash equilibrium (see [12]).

The classical Sperner's Lemma [14] states that in a triangle with a regular triangulation whose vertices are labeled with three colors, there is always a trichromatic triangle. This lemma is of special interest since some customary proofs for the above topological fixed point theorems rely on its combinatorial content. However, it is unknown whether the corresponding search problem, that Papadimitriou [12] calls **2D-SPERNER**, is complete in PPAD. Variants of Sperner's Lemma also give rise to other problems in the parity argument classes. Papadimitriou [12] has proved that a 3-dimensional analogue of **2D-SPERNER** is in fact complete in PPAD. In [6], Grigni described a non-oriented version of 3-dimensional Sperner's Lemma that is complete for the class PPA. In this paper we show that appropriately chosen locally 2-dimensional versions of the problem are already complete for PPAD, for PPADS, and for PPA, respectively.

This work was completed early 2005 [5]. Recently it has been announced by Chen and Deng that they have proven the PPAD completeness of **2D-SPERNER** in reference 2 in [2].

2 Results

An *NP-search problem* is specified by a polynomial time relation $\mathcal{R}(x, y)$, such that for some polynomial $p(n)$, for every x and y such that $\mathcal{R}(x, y)$, we have $|y| \leq p(|x|)$. Given an input x to the problem, the task is to find a y such that $\mathcal{R}(x, y)$ if there is one, and else report failure. We call an NP-search problem *total* if for every x there exists a solution y. The class of total NP-search problems is called TFNP by Megiddo and Papadimitriou [9].

For two problems $\mathcal{R}_1$, $\mathcal{R}_2$ in TFNP, we say that $\mathcal{R}_1$ is *reducible to* $\mathcal{R}_2$ if there exist two functions f and g computable in polynomial time such that $f(x)$

is a legal input to $\mathcal{R}_2$ whenever x is an input to $\mathcal{R}_1$, and $\mathcal{R}_2(f(x), y)$ implies $\mathcal{R}_1(x, g(x, y))$.

The parity argument classes are defined via concrete problems, by closure under reduction. The **LEAF** problem is defined as follows. The input is a pair $(M, 0^k)$ where M is the description of a polynomial time Turing machine that on every input outputs a set of size at most 2, and k is a positive integer. Moreover, M is such that $M(0^k) = \{1^k\}$, and $0^k \in M(1^k)$. Such an input specifies an undirected graph $G_k = (V, E)$, where $V = \{0, 1\}^k$, and $\{u, v\}$ is in E if $u \in M(v)$, and $v \in M(u)$. The output of the problem is a leaf of G_k different from 0^k. The class PPA is the set of total search problems reducible to **LEAF**. In the search problems defining the classes PPADS and PPAD, the Turing machine defines a directed graph, where the in-degree and the out-degree of every vertex is at most one, and where 0^k is always a source. The output in the case of PPADS is a sink, and in the case of PPAD a sink or source different from 0^k.

After some preliminaries in Section 3 the definitions of the three Sperner problems of interest for us will be given in Section 4: **OSPS** and **SOSPS** for the oriented cases, and **SPS** for the non-oriented case. Our main results are proven in Section 5: **OSPS** is complete for PPAD (**Theorem 2**) and **SOSPS** is complete for PPADS (**Theorem 3**). The proof of the completeness of **SPS** for PPA is left for the full paper.

The results of this paper are motivated by an open problem of Papadimitriou in [12], asking whether **2D-SPERNER** is PPAD-complete. The main reason why the 3-dimensional Sperner problem could be proved complete in PPAD is that there exists an embedding of the complete graph of any size in the 3-dimensional Euclidean space without any two edges crossing. Of course, such an embedding is impossible in the plane, and it is not clear how to circumvent this difficulty when one tries to extend Papadimitriou's proof in 2 dimensions. Our approach consists in exhibiting such an embedding in compact 2-dimensional manifolds, i.e. surfaces, of non-zero genus, and proving the completeness of Sperner problems on these surfaces for the classes PPAD, PPADS and PPA. Therefore, our results show that the difficulty of the Sperner problems is independent of the local dimension of the instance, if it is at least 2.

3 Preliminaries

Unless otherwise stated, the graphs considered in the paper will be undirected. If S is any set, $\equiv$ is an equivalence relation over S, and a is an element of S, then $[a]_\equiv$ denotes the equivalence class of a in S for the relation $\equiv$.

3.1 Surfaces

Definition 1 (triangles). *Let $\mathfrak{R}$ be the equivalence relation over triples of distinct elements such that we have $(a, b, c)\,\mathfrak{R}\,(a', b', c')$ if (a', b', c') is obtained from (a, b, c) by cyclic permutation. An equivalence class T of $\mathfrak{R}$ is called a* triangle.

If T is the equivalence class of (a, b, c), then $\overline{T}$ denotes the equivalence class of (a, c, b).

For a pair (a, b), let $\overline{(a, b)}$ denote the pair (b, a). For every triangle T and elements a and b, $(a, b) \prec T$ indicates that there exists an element c such that $(a, b, c)\,\mathfrak{R}\,T$, and $\{a, b\} \prec T$ indicates that either $(a, b) \prec T$ or $\overline{(a, b)} \prec T$.

A finite set of triangles $\mathcal{T}$ is called a triangle arrangement*. If $\mathcal{T}$ is a triangle arrangement, its* skeleton graph *$G_{\mathcal{T}}$ is the graph $G_{\mathcal{T}} = (V, E)$, where $V = \bigcup_{T \in \mathcal{T}} T$, and $\{a, b\}$ is an edge if there is a triangle $T \in \mathcal{T}$ such that $\{a, b\} \prec T$. A* vertex *(*resp. edge*) of $\mathcal{T}$ is a vertex (*resp. *edge) of the skeleton graph of $\mathcal{T}$.*

We will often specify a triangle T, which is an equivalence class of $\mathfrak{R}$, by a an element of T.

Definition 2 (pseudosurfaces). *A* pseudo-surface *$\mathcal{T}$ is a triangle arrangement $\mathcal{T}$ such that for every edge (a, b) of E there are at most two different triangles $T \in \mathcal{T}$ such that $\{a, b\} \prec T$. The pseudo-surface $\mathcal{T}$ is* oriented *if for every two triangles T and T' in $\mathcal{T}$ and every edge $\{a, b\} \in E$, when $(a, b) \prec T$ and $(a, b) \prec T'$ we have $T = T'$. The* boundary *of $\mathcal{T}$, denoted by $\partial\mathcal{T}$, is the set of all edges $e \in E$ for which there exists exactly one triangle $T \in \mathcal{T}$ with $e \prec T$. The* dual graph *$H_{\mathcal{T}}$* of *$\mathcal{T}$ is the graph $H_{\mathcal{T}} = (\mathcal{T}, E')$ such that there is an edge between two triangles $T \neq T'$ in $H_{\mathcal{T}}$ if there are two vertices a and b in $\mathcal{T}$ such that $\{a, b\} \prec T$ and $\{a, b\} \prec T'$.*

Definition 3. *A* surface *$\mathcal{S}$ is a pseudo-surface such that $H_{\mathcal{S}}$ is connected and $\partial\mathcal{S}$ is a union of disjoint cycles of $G_{\mathcal{S}}$.*

Notice that our definition of surface coincides with the usual definition of triangulated surface.

3.2 Flow graphs

Definition 4. *Let $\mathcal{S}$ be a surface, V be set of vertices of $\mathcal{S}$, $H_{\mathcal{S}} = (V', E')$ be its dual graph. A function $\ell : V \to \{0, 1, 2\}$ is called a* labeling of *$\mathcal{S}$. A triangle $T \in \mathcal{S}$ is said to be* fully labeled *if it is equivalent to a triple (a, b, c) such that $\{\ell(a), \ell(b), \ell(c)\} = \{0, 1, 2\}$. A fully labeled triangle T has* direct orientation *if there exists (a, b, c) in its equivalence class such that $(\ell(a), \ell(b), \ell(c)) = (0, 1, 2)$. Otherwise, it has* indirect orientation.

The undirected flow graph *$U_{\mathcal{S}} = (V', E'')$* of *$\mathcal{S}$* (relatively to *$\ell$) is a subgraph of $H_{\mathcal{S}}$, such that there is an edge between two triangles T and T' of $\mathcal{S}$ if there are two vertices a and b of $\mathcal{S}$ such that $\{a, b\} \prec T$, $\{a, b\} \prec T'$, and $\{\ell(a), \ell(b)\} = \{0, 1\}$.*

If $\mathcal{S}$ is oriented, then we define the directed flow graph *$D_{\mathcal{S}} = (V', E''')$* of *$\mathcal{S}$* (relatively to *$\ell$) as a a directed graph, such that there is an edge between two triangles T and T' of $\mathcal{S}$ if there are two vertices a and b of $\mathcal{S}$ such that $(a, b) \prec T$, $\overline{(a, b)} \prec T'$, and $(\ell(a), \ell(b)) = (0, 1)$.*

The proof of the following theorem is straightforward.

Theorem 1. *Let $\mathcal{S}$ be a surface, and ℓ be a labeling of $\mathcal{S}$. Then,*

(i) the degree of every vertex of the undirected flow graph $U_{\mathcal{S}}$ is at most 2,
(ii) if $\mathcal{S}$ is oriented, then the in-degree and out-degree of every vertex of the directed flow graph $D_{\mathcal{S}}$ are at most 1.

Corollary 1 (Sperner's lemma for surfaces with empty boundary). *Let $\mathcal{S}$ be a surface with empty boundary, and ℓ be a labeling of $\mathcal{S}$. Then,*

(i) the number of fully labeled triangles in the undirected flow graph $U_{\mathcal{S}}$ is even,
(ii) if $\mathcal{S}$ is oriented, then there are as many fully labeled triangles with direct orientation as fully labeled triangles with indirect orientation in the directed flow graph $D_{\mathcal{S}}$.

Proof. First, observe that the fully labeled triangles in $\mathcal{S}$ are exactly the nodes of degree 1 in $U_{\mathcal{S}}$, and that the fully labeled triangles having direct (*resp.* indirect) orientation in $\mathcal{S}$ are exactly the nodes of out-degree (*resp.* in-degree) 1 in $D_{\mathcal{S}}$. Since by Theorem 1 *(i)* in $U_{\mathcal{S}}$ the maximal degree is at most two, the number vertices having degree 1 is even. By Theorem 1 *(ii)* in $D_{\mathcal{S}}$ the in- and outdegrees are at most 1, therefore there has to be the same number of sources as sinks.

3.3 Rotation systems

Definition 5. *Let $G = (V, E)$ be a graph. For every vertex $v \in V$, a* local rotation *of G* at *v is a cyclic permutation π_v of the neighbors of v in G. A* rotation system *for G is a set $\Pi = \{\pi_v \,|\, v \in V\}$ of local rotations. Let $\mathcal{T}$ be a triangle arrangement, and v be a vertex of $\mathcal{T}$. A local rotation π_v of $G_{\mathcal{T}}$ at v is a* local orientation *of $\mathcal{T}$* at *v if, for every neighbor v' of v in $G_{\mathcal{T}}$, $(v', v, \pi_v(v'))$ is a triangle of $\mathcal{T}$.*

Fact 1. *Let $\mathcal{S}$ be an oriented surface with empty boundary, and let v be a vertex of $\mathcal{S}$. There exists a unique local orientation π_v of $\mathcal{S}$ at v such that, for every neighbor v' of v in $G_{\mathcal{S}}$, $(v', v, \pi_v(v'))$ is a triangle of $\mathcal{S}$.*

Definition 6. *Let $\mathcal{S}$ be an oriented surface with empty boundary. The rotation system defined in Fact 1 is called* the rotation system of *$\mathcal{S}$.*

Definition 7. *Let $(G_n)_{n\in\mathbb{N}} = (V_n, E_n)_{n\in\mathbb{N}}$ be a family of undirected graphs where $|V_n| = n$, and $\Pi_n = \{\pi_v \,|\, v \in V_n\}$ be a rotation system for G_n. The rotation system Π_n is said to be* efficiently computable *if there exists a Turing machine M such that*

(i) on input n and pair (v, v'), with $\{v, v'\} \in E_n$, computes the vertices v'' and v''' such that $\pi_v(v') = v''$ and $\pi_v^{-1}(v') = v'''$ using time polynomial in $\log n$,
(ii) on input n and triple (v, v', v''), with $\{v, v'\}$ and $\{v, v''\}$ in E_n, computes the smallest non-negative integer i such that $\pi_v^i(v') = v''$ using time polynomial in $\log n$. Later, we will refer to the integer i by $\log_{v'}^{\pi_v}(v'')$.

Lemma 1. *If m is an integer that is equal to 7 modulo 12, then the complete graph K_m is the skeleton graph of an oriented surface $\mathcal{S}_m$ with empty boundary. Moreover, the rotation system of $\mathcal{S}_m$ can be efficiently computed.*

The surface $\mathcal{S}_m$ is completely specified by giving an appropriate rotation system for K_m. There are actually several such rotation systems [3, 8]. The proof of the efficient computability of the rotation system is straightforward. It is based on the constructions in [10, 3]. We omit the details.

3.4 Regular subdivisions

In the following definition, we will formalize the notion of "a regular subdivision" of a surface, which consists in substituting every triangle of the surface with a "regular subdivision" of it, as shown on Figure 1, such that the small triangles of the subdivision have the same orientation as the large triangle that is subdivided.

We will make use of the free Abelian monoid $\mathbb{N}[V]$ over the set of vertices V of a surface $\mathcal{S}$: the elements are those of the form $\sum_{v\in V} c_v \cdot v$, where c_v is a non-negative integer, and v is a vertex of $\mathcal{S}$. For any subset $V' \subseteq V$ and positive integer r let $\mathbb{N}_r[V']$ denote those elements $\sum_{v\in V'} c_v \cdot v$ of $\mathbb{N}[V]$ such that $\sum_{v\in V'} c_v = r$. If $s = \sum_{v\in V} s_v \cdot v$ and $t = \sum_{v\in V} t_v \cdot v$ are two elements of $\mathbb{N}[V]$, we denote by $d(s,t)$ the distance $1/2 \sum_{v\in V} |s_v - t_v|$.

Definition 8. *Let $\mathcal{S}$ be a surface, and r be a positive integer. Let $\mathcal{S}^{(r)}$ be a triangle arrangement whose triangles are of the form (s_1, s_2, s_3) with $\{s_1, s_2, s_3\} \subseteq \mathbb{N}_r[\{a, b, c\}]$, for some triangle (a, b, c) in $\mathcal{S}$, such that there exists $\varepsilon \in \{-1, 1\}$ with $s_2 = s_1 + \varepsilon(a - b)$ and $s_3 = s_1 + \varepsilon(a - c)$. We call $\mathcal{S}^{(r)}$ the* regular r-subdivision *of $\mathcal{S}$.*

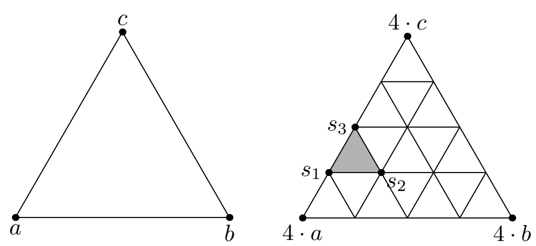

Fig. 1: A triangle (a, b, c) and its regular 4-subdivision.

Notice that two vertices of $\mathcal{S}^{(r)}$ are neighbors if and only if they are at distance 1. It implies that the distance between two vertices in the skeleton graph of $\mathcal{S}^{(r)}$ is equal to their distance according to d.

4 Sperner Problems

The NP-search problems for which we prove completeness in Section 5 are the following. The surface $\mathcal{S}_m$ is the one given by Lemma 1. Its skeleton graph is K_m. The surface $\mathcal{S}_m^{(4)}$ is the regular 4-subdivision of $\mathcal{S}_m$.

Oriented Sperner Problem for the Surface $\mathcal{S}_m^{(4)}$ (OSPS)

Input: an integer m equal to 7 modulo 12, the description of a Turing machine M that on input vertex v of $\mathcal{S}_m^{(4)}$ outputs a label $\ell(v)$ in $\{0,1,2\}$ using time polynomial in $\log m$, and also a fully labeled triangle T of $\mathcal{S}_m^{(4)}$, which has indirect orientation.

Output: a fully labeled triangle $T' \neq T$ of $\mathcal{S}_m^{(4)}$.

Strict Oriented Sperner Problem for the Surface $\mathcal{S}_m^{(4)}$ (SOSPS)

Input: an integer m equal to 7 modulo 12, the description of a Turing machine M that on input vertex v of $\mathcal{S}_m^{(4)}$ outputs a label $\ell(v)$ in $\{0,1,2\}$ using time polynomial in $\log m$, and also a fully labeled triangle T of $\mathcal{S}_m^{(4)}$, which has indirect orientation.

Output: a fully labeled triangle T' of $\mathcal{S}_m^{(4)}$, which has direct orientation.

To prove completeness for a non-oriented Sperner problem, we will use the non-oriented surface $\mathcal{N}_m$, derived from the regular 12-subdivision $\mathcal{S}_m^{(12)}$ of $\mathcal{S}_m$ by adding some cross-caps. Its precise definition will not be given in this extended abstract.

Sperner Problem for the Surface $\mathcal{N}_m^{(12)}$ (SPS)

Input: an integer m equal to 7 modulo 12, the description of a Turing machine M that on input vertex v of $\mathcal{N}_m^{(12)}$ outputs a label $\ell(v)$ in $\{0,1,2\}$ using time polynomial in $\log m$, and also a fully labeled triangle T of $\mathcal{N}_m^{(12)}$.

Output: a fully labeled triangle $T' \neq T$ of $\mathcal{N}_m^{(12)}$.

We would like to emphasize that these Sperner problems are in fact not promise problems, since the input requirements can be syntactically enforced. Let us describe this in details for the case of **OSPS**. We can easily provide a syntactical way to force the Turing machine to always give a correct output. For instance, one can assume that every output value not in $\{0,1,2\}$ is interpreted as 0. We can also ensure syntactically that T is a fully labeled triangle which has indirect orientation with the help of an arbitrary polynomial time computable total order $<$ on the vertices of $\mathcal{S}_m^{(4)}$. Let $s_1 < s_2 < s_3$ be the vertices of T. The label of s_3 is fixed to 2. The vertex s_1 will get label 0 and s_2 label 1 if (s_1, s_2, s_3) is in the equivalence class T, and the labels are exchanged in the opposite case.

In fact, the membership of each of these problems in the class TFNP follows immediately from Corollary 1.

5 Completeness results for oriented Sperner problems

Let m be a positive integer equal to 7 modulo 12. We will work with the regular 4-subdivision $\mathcal{S}_m^{(4)}$ of $\mathcal{S}_m$.

Theorem 2. *The problem* **OSPS** *is PPAD-complete.*

Proof. To see membership in PPAD, we reduce **OSPS** to the natural complete problem for PPAD. First, notice that from Theorem 1, we know that the directed flow graph $D_{\mathcal{S}_m^{(4)}}$ has in- and out-degree at most 1 at every vertex. Notice also that, given a polynomial Turing machine that outputs the label of vertices of $\mathcal{S}_m^{(4)}$, it is easy to design a polynomial time Turing machine that, given a vertex T in the directed flow graph $D_{\mathcal{S}_m^{(4)}}$ outputs its predecessor and its successor, if they exist: the Turing machine only has to calculate the labels of the vertices in T, and to calculate which are the neighbors of T in $H_{\mathcal{S}_m^{(4)}}$. Finally, observe that, as we previously mentioned in the proof of Corollary 1, the fully labeled triangles having direct (*resp.* indirect) orientation in $\mathcal{S}$ are exactly the nodes of out-degree (*resp.* in-degree) 1 in $D_{\mathcal{S}_m^{(12)}}$. These three arguments show that there is a reduction (in the sense of total problems) from **OSPS** to the natural complete problem for **PPAD**.

We turn to the proof of completeness. Let k be any positive integer. Let $G = (V, E)$ be a graph which is specified by an instance of the natural complete problem for PPAD (see Section 2). It is an undirected graph over $V = \{0, 1\}^k$, such that each vertex has in-degree at most one, and out-degree at most one. Moreover, 0^k is a source in G. Let us denote by M the polynomial time Turing machine that, given a vertex $v \in V$, outputs its predecessor and its successor, if they exist. From G we make an instance of **OSPS** such that a solution can be efficiently turned into a source or a sink of the graph G different from 0^k.

Let m be the smallest integer greater than 2^k that is equal to 7 modulo 12. We assume that V is included in the set of vertices of $\mathcal{S}_m$. We denote by $\Pi = \{\pi_v \mid v \text{ vertex of } \mathcal{S}_m\}$ the rotation system for $\mathcal{S}_m$.

Informally, we give a labeling such that the directed flow graph $D_{\mathcal{S}_m^{(4)}}$ imitates the graph G as follows: if (a, b) is an edge of G, then there will be a path in $D_{\mathcal{S}_m^{(4)}}$ along the edges near the (a, b) side of the triangle "above" (a, b) (that is the triangle $\{a, b, \pi_a^{-1}(b)\}$). If moreover (b, c) is an edge in G then there will be a path around b in the direction given by the rotation system, leading to the triangle above (b, c). To manage the latter, we need a tool for deciding whether, for a vertex $d \notin \{a, b, c\}$, the edge $\{b, d\}$ is "between" $\{a, b\}$ and $\{b, c\}$ according to the rotation π_b. This tool is provided by the function $\log_a^{\pi_b}$ defined in Definition 7: the edge $\{b, d\}$ is between $\{a, b\}$ and $\{b, c\}$ if $0 < \log_a^{\pi_b}(d) < \log_a^{\pi_b}(c)$. The function $\log_a^{\pi_b}$ is efficiently computable by Lemma 1.

We design a Turing machine M' that for every vertex v in $\mathcal{S}_m^{(4)}$ outputs a label $\ell(v)$ in $\{0, 1, 2\}$, using M as a subroutine. Let (a, b, c) be a triangle in $\mathcal{S}_m$, $S = \{a, b, c\}$, and let i_a, i_b and i_c be three non-negative integers such that $i_a + i_b + i_c = 4$. Denote by σ the permutation $\binom{a,b,c}{b,c,a}$. Observe that the definition of the rotation system implies that for every $v \in \{a, b, c\}$ the equality $\pi_v(\sigma^{-1}(v)) = \sigma(v)$ holds. On input $z = i_a \cdot a + i_b \cdot b + i_c \cdot c$ the Turing machine M' outputs

$$\ell(z) = \begin{cases} 0 & \text{if } \exists v, v' \in S,\ i_v + i_{v'} = 4,\ (v, v') \in E, & (1) \\ 0 & \text{if } \exists v \in S,\ i_v = 4,\ \exists w \notin S,\ (v, w) \in E \text{ or } (w, v) \in E, & (2) \\ 1 & \text{if } \exists v \in S,\ (i_v, i_{\sigma(v)}) \in \{(2,1), (1,2)\},\ (v, \sigma(v)) \in E, & (3) \\ 1 & \text{if } \exists v \in S,\ \exists v' \in \{\sigma^{-1}(v), \sigma(v)\},\ (i_v, i_{v'}) = (3,1), \\ & \quad \exists w, w' \in V,\ (w, v), (v, w') \in E \text{ and } \log_w^{\pi_v}(v') < \log_w^{\pi_v}(w'), & (4) \\ 2 & \text{otherwise.} & (5) \end{cases}$$

Notice that conditions 1 and 2 can be matched simultaneously, but the value of ℓ is the same. Notice also that, although less obvious, it is impossible for conditions 1 and 4 to be matched simultaneously. The other pairs of conditions can not be matched simultaneously.

Finding the case in which z falls can be done in time polynomial in k, as the Turing machine M, on input $v \in \{a, b, c\}$, outputs the neighbors of v, and the rotation system Π can be efficiently computed.

Using these rules, we describe (see Figure 2) the possible cases for a triangle (a, b, c) in $\mathcal{S}_m$ (we assume that the rotation system is clockwise, and hence the orientation is counter-clockwise):

Case 1: (a, b), (b, c), $(c, a) \in E$.

Case 2: (a, b), $(b, c) \in E$, but $(c, a) \notin E$. The value of $\ell(3 \cdot a + c)$ is 2 if a is a source in G, and 1 otherwise. Similarly, the value of $\ell(3 \cdot c + a)$ is 2 if c is a sink in G, and 1 otherwise.

Case 3: $(a, b) \in E$, but (b, c) and (c, a) are not in E. The value of $\ell(4 \cdot c)$ is 2 if c is isolated in G, and otherwise 0. The value of $\ell(a + 3 \cdot c) = \ell(b + 3 \cdot c)$ is 1 if $\log_w^{\pi_c}(b) < \log_w^{\pi_c}(a) < \log_w^{\pi_c}(w')$, and otherwise 2. The value of $\ell(3 \cdot a + c)$ is 2 if a is a source in G, and 1 otherwise. The value of $\ell(3 \cdot b + c)$ is 2 if b is a sink in G, and 1 otherwise.

Case 4: (a, b), (b, c) and (c, a) are not in E. Let v be in $\{a, b, c\}$. We do not enumerate all the possible sub-cases, but only state the essential relations between the labels:

- (i) $\ell(3 \cdot v + \sigma(v)) = 1 \iff \ell(3 \cdot v + \sigma^{-1}(v)) = 1$, as both $3 \cdot v + \sigma^{-1}(v)$ and $3 \cdot v + \sigma(v)$ simultaneously fall in one of the cases (1), (4) and (5) in the definition of ℓ.
- (ii) $\ell(3 \cdot v + \sigma(v)) = 0 \iff \ell(2 \cdot v + 2 \cdot \sigma(v)) = 0$, as if $\ell(3 \cdot v + \sigma(v)) = 0$ or $\ell(2 \cdot v + 2 \cdot \sigma(v)) = 0$ then case (1) in the definition of ℓ must apply,
- (iii) $\ell(3 \cdot v + \sigma^{-1}(v)) = 0 \iff \ell(2 \cdot v + 2 \cdot \sigma^{-1}(v)) = 0$, for the same reasons as in *(ii)*.

These are the only possible cases, up to renaming the vertices a, b and c of the triangle (a, b, c).

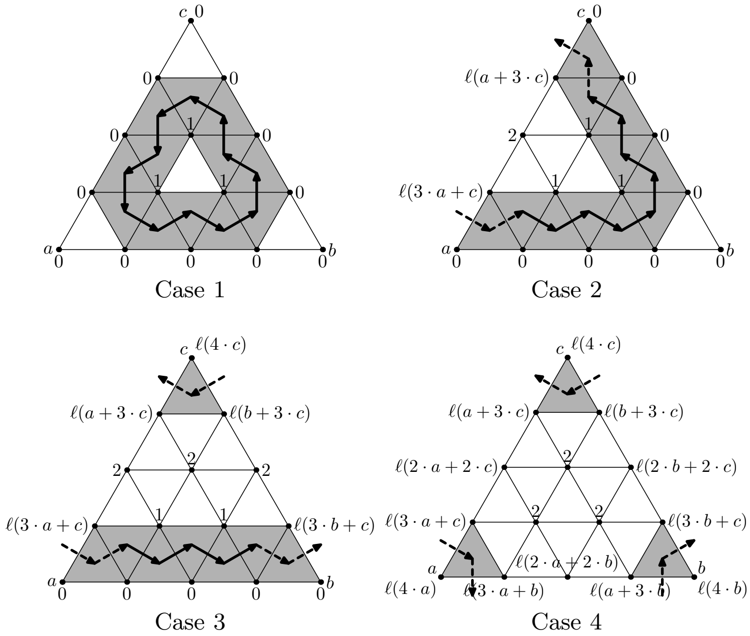

Fig. 2: The different possible cases in the labeling of a triangle (a, b, c) of $\mathcal{S}_m^{(4)}$.

We have to prove that this labeling scheme ℓ is correctly defined among different triangles. It is easy to check that it is correctly defined on $4 \cdot v$, where v is a vertex of V: if v is an isolated vertex in G, then in every face to which it belongs only the case (5) in the definition of ℓ applies, and therefore $\ell(4 \cdot v) = 2$. If v is not isolated, then case (2) in the definition of ℓ applies, and therefore $\ell(4 \cdot v) = 0$.

So, finally, proving that the labeling has been correctly defined amounts to proving that the label $\ell(z)$ of a vertex $z = i_a \cdot a + i_b \cdot b$, $0 < i_a, i_b < 4$ with $i_a + i_b = 4$, that we have defined is the same for the two triangles $(a, b, \pi_a^{-1}(b))$ and $(a, \pi_a(b), b)$. We study the different cases:

- $(i_a, i_b) = (3, 1)$ or $(1, 3)$: if (a, b) or (b, a) is in E, then case (1) in the definition of ℓ applies to z, and $\ell(z) = 0$. Otherwise, either case (4) applies and therefore $\ell(z) = 1$, or case (5) applies and therefore $\ell(z) = 2$.
- $(i_a, i_b) = (2, 2)$: if (a, b) or (b, a) is in E, then case (1) in the definition of ℓ applies to z, and $\ell(z) = 0$. Otherwise, case (5) applies.

Let (a', b', c') be a fully labeled triangle in the subdivision of a triangle (a, b, c) in $\mathcal{S}_m$. We prove that there exists a unique $v = v(a', b', c') \in \{a, b, c\}$ such that $(a', b', c') = (3 \cdot v + \sigma(v), 2 \cdot v + \sigma^{-1}(v) + \sigma(v), 3 \cdot v + \sigma^{-1}(v))$, and v is a source in G if (a', b', c') is a fully labeled triangle having indirect orientation, and a sink if (a', b', c') is a fully labeled triangle having direct orientation. Also, given (a', b', c'), one can efficiently retrieve $v(a', b', c')$. The proof is done for the different cases of Figure 2.

In Case 1 there is no such triangle (a', b', c').

Let us examine Case 2. The possible values for $\ell(3\cdot a+c)$ and $\ell(a+3\cdot c)$ are 1 and 2. Therefore, the only possibilities for (a', b', c') are $(b+3\cdot c, a+b+2\cdot c, a+3\cdot c)$ when $\ell(a+3\cdot c)=2$, and $(3\cdot a+c, 2\cdot a+b+c, 3\cdot a+b)$ when $\ell(3\cdot a+c)=2$. These values correspond respectively to the case when c is a sink and (a', b', c') is a fully labeled triangle having direct orientation, and to the case when a is a source and (a', b', c') is a fully labeled triangle having indirect orientation.

Let us turn to Case 3. In this case, we always have $\ell(a+3\cdot c)=\ell(b+3\cdot c)$. So, the only possibilities for (a', b', c') are $(3\cdot b+a, a+2\cdot b+c, 3\cdot b+c)$ when $\ell(3\cdot b+c)=2$, and $(3\cdot a+c, 2\cdot a+b+c, 3\cdot a+b)$ when $\ell(3\cdot a+c)=2$. These values correspond respectively to the case when b is a sink and (a', b', c') is a fully labeled triangle having direct orientation, and to the case when a is a source and (a', b', c') is a fully labeled triangle having indirect orientation.

We finish the case study by proving that in Case 4, there can be no fully labeled triangle (a', b', c'). All the triangles that have twice the label 2 can immediately be discarded. By symmetry between a, b and c, we can assume without loss of generality that a', b' and c' should be in $\{i_a\cdot a+i_b\cdot b+i_c\cdot c\in\mathbb{N}_4[\{a,b,c\}]\mid i_a\geq 2\}$. Assume that (a', b', c') is a fully labeled triangle. The possibilities are:

- $(a', b', c')=(3\cdot a+c, 4\cdot a, 3\cdot a+b)$: $\ell(4\cdot a)\in\{0,2\}$, so $\ell(3\cdot a+c)=1$ or $\ell(3\cdot a+b)=1$, and therefore relation *(i)* implies $\ell(3\cdot a+c)=\ell(3\cdot a+b)$, which is impossible,
- $(a', b', c')=(3\cdot a+c, 3\cdot a+b, 2\cdot a+b+c)$: similar to the previous case,
- $(a', b', c')=(2\cdot a+2\cdot c, 3\cdot a+c, 2\cdot a+b+c)$: $\ell(2\cdot a+2\cdot c)\in\{0,2\}$ and $\ell(2\cdot a+b+c)=2$, so $\ell(2\cdot a+2\cdot c)=0$ and therefore relation *(ii)* implies $\ell(3\cdot a+c)=0$, which is impossible,
- $(a', b', c')=(3\cdot a+b, 2\cdot a+2\cdot b, 2\cdot a+b+c)$: similar to the previous case, using relation *(iii)*.

Our next step is showing that the map $(a', b', c')\mapsto v(a', b', c')$ is a bijection between fully labeled triangles (a', b', c') having indirect orientation and sources of G. It is onto, as if v is a source in G, v' is the successor of v and $v''=\pi_v^{-1}(v')$ then $v=v(a', b', c')$, where $(a', b', c')=(3\cdot v+v'', 2\cdot v+v'+v'', 3\cdot v+v')$. The case study also shows that if (a', b', c') is a fully labeled triangle of $\mathcal{S}_m^{(4)}$ having indirect orientation and $v=v(a', b', c')$ then $(a', b', c')=(3\cdot v+v'', 2\cdot v+v'+v'', 3\cdot v+v')$, where v' is the successor of v in G and $v''=\pi_v^{-1}(v')$. Therefore the map is injective as well. A similar bijection exists between fully labeled triangles (a', b', c') having direct orientation and sinks of G.

Let $(a_0, b_0, c_0)=(0^k, 1^k, \pi_{0^k}^{-1}(1^k))$. The triangle T, which is part of the input for **OSPS**, is $(3\cdot a_0+c, 2\cdot a_0+b_0+c_0, 3\cdot a_0+b_0)$. We conclude that if we can find a fully labeled triangle $(a', b', c')\neq T$ then we can efficiently retrieve a source or sink v of G with $v=v(a', b', c')$ different from 0^k.

The problem **SOPS** is the hand-made analogue of **OSPS** for PPADS, and therefore it is naturally complete in the class. Indeed, both for the easiness and the hardness results the proofs used for the completeness of **OSPS** in PPAD can be applied. The only substantial remark to be made is that the bijections defined between fully labeled triangles and nodes of degree one bijectively map fully labeled triangles having direct orientations onto sinks. Therefore, we obtain the following result.

Theorem 3. *The problem* **SOSPS** *is PPADS-complete.*

Acknowledgments

The last two authors acknowledge the hospitality of the Isaac Newton Institute, Cambridge, where their work was completed.

References

1. P. Beame, S. Cook, J. Edmonds, R. Impagliazzo and T. Pitassi. The relative complexity of NP search problems. *J. Comput. System Sci.*, 57(1):3–19, 1998.
2. X. Chen and X. Deng. Settling the Complexity of 2-Player Nash-Equilibrium. *ECCC Report* 140, 2005.
3. L. Goddyn, R. Bruce Richter and Jozef Širáň. Triangular embeddings of complete graphs from graceful labellings of paths. Preprint, 2004.
4. K. Fan. Simplicial maps from an orientable n-pseudomanifold into S^m with the octahedral triangulation. *J. Combinatorial Theory*, 2:588–602, 1967.
5. K. Friedl, G. Ivanyos, M. Santha and Y. F. Verhoeven. On the complexity of Sperner's Lemma. Isaac Newton Institute Preprint Series NI05002, 2005.
6. M. Grigni. A Sperner lemma complete for PPA. *Inform. Process. Lett.*, 77(5-6):255–259, 2001.
7. D. Johnson, C. Papadimitriou and M. Yannakakis. How easy is local search? *J. Comput. System Sci.*, 37(1):79–100, 1988.
8. V. P. Korzhik and H.-J. Voss, On the Number of Nonisomorphic Orientable Regular Embeddings of Complete Graphs, *J. Combinatorial Theory Series B* **81***:58-76 (2001).*
9. N. Megiddo and C. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoret. Comput. Sci.*, 81:317–324, 1991.
10. B. Mohar and C. Thomassen. *Graphs on surfaces.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2001.
11. C. Papadimitriou. On graph-theoretic lemmata and complexity classes. *In Proc. of 31st FOCS*, pp. 794–801, 1990.
12. C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.*, 48(3):498–532, 1994.
13. G. Ringel, *Map Color Theorem*, Springer-Verlag, New York, 1974.
14. E. Sperner. Neuer Beweis für die Invarianz der Dimensionzahl und des Gebietes. *Abh. Math. Sem. Hamburg Univ.* 6:265–272, 1928.
15. A. Thomason. Hamilton cycles and uniquely edge colourable graphs. *Ann. Discrete Math.* 3: 259–268, 1978.