COLORFUL LINEAR PROGRAMMING, NASH EQUILIBRIUM, AND PIVOTS

FRÉDÉRIC MEUNIER AND PAULINE SARRABEZOLLES

Abstract. The colorful Carathéodory theorem, proved by Bárány in 1982, states that given $d+1$ sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ in $\mathbb{R}^d$, such that each $\mathbf{S}_i$ contains $\mathbf{0}$ in its convex hull, there exists a set $T \subseteq \bigcup_{i=1}^{d+1} \mathbf{S}_i$ containing $\mathbf{0}$ in its convex hull and such that $|T \cap \mathbf{S}_i| \leq 1$ for all $i \in \{1, \ldots, d+1\}$. An intriguing question – still open – is whether such a set T, whose existence is ensured, can be found in polynomial time. In 1997, Bárány and Onn defined colorful linear programming as algorithmic questions related to the colorful Carathéodory theorem. The question we just mentioned comes under colorful linear programming.

We present new complexity results for colorful linear programming problems and propose a variant of the "Bárány-Onn" algorithm, which is an algorithm computing a set T whose existence is ensured by the colorful Carathéodory theorem. Our algorithm makes a clear connection with the simplex algorithm. Some combinatorial versions of the colorful Carathéodory theorem are also discussed from an algorithmic point of view. Finally, we show that computing a Nash equilibrium in a bimatrix game is polynomially reducible to a colorful linear programming problem. On our track, we found a new way to prove that a complementarity problem belongs to the PPAD class with the help of Sperner's lemma.

1. Introduction

1.1. **Context.** In 1982, Bárány proved a colorful generalization of the Carathéodory theorem, whose statement is the following.

Theorem 1 (Colorful Carathéodory theorem [2])**.** *Given $d+1$ sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ in $\mathbb{R}^d$, such that each $\mathbf{S}_i$ contains $\mathbf{0}$ in its convex hull, there exists a set $T \subseteq \bigcup_{i=1}^{d+1} \mathbf{S}_i$ containing $\mathbf{0}$ in its convex hull and such that $|T \cap \mathbf{S}_i| \leq 1$ for all $i \in \{1, \ldots, d+1\}$.*

A natural question raised by this theorem is whether such a *colorful* set T can be computed in polynomial time. The case with $\mathbf{S}_1 = \cdots = \mathbf{S}_{d+1}$, corresponding to the usual Carathéodory theorem, is known to be solvable in polynomial time, via linear programming. However, the complexity of the colorful version remains an open question.

In 1997, Bárány and Onn defined algorithmic and complexity problems related to the colorful Carathéodory theorem [4], giving birth to colorful linear programming. In their paper, the complexity question raised by the colorful Carathéodory theorem is referred as an "outstanding problem on the borderline of tractable and intractable problems". In addition to provide a theoretical challenge, the colorful Carathéodory theorem has several applications in discrete geometry (e.g. Tverberg partition, "first selection lemma", see [18]). Any efficient algorithm computing such a colorful set T would benefit these applications. We formally define the problem of finding a set T as in the colorful Carathéodory theorem. It is a *search*

Key words and phrases. bimatrix game, colorful linear programming, complexity, pivoting algorithms .

problem, and more specifically belongs to the *TFNP class*. A search problem is like a decision problem but a certificate is sought in addition to the 'yes' or 'no' answer. The class of search problems whose decision counterpart has always a 'yes' answer is called TFNP, where TFNP stands for "Total Function Non-deterministic Polynomial".

A set of points is said to be *positively dependent* if it is nonempty and contains $\mathbf{0}$ in its convex hull. Given a configuration of k sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_k$ in $\mathbb{R}^d$, a set $T \subseteq \bigcup_{i=1}^k \mathbf{S}_i$ such that $|T \cap \mathbf{S}_i| \leq 1$ for $i = 1, \ldots, k$ is said to be *colorful*.

Colorful Linear Programming (TFNP version)
Input. A configuration of $d+1$ positively dependent sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ in $\mathbb{Q}^d$.
Task. Find a positively dependent colorful set.

As we have already mentioned, the complexity status is still open. A more general problem, still in TFNP, has been recently proved to be PLS-complete by Mulzer and Stein [22]. The PLS class, where PLS stands for "Polynomial Local Search", is a subclass of the TFNP class and contains the problems for which local optimality can be verified in polynomial time [12].

The original proof of the colorful Carathéodory theorem by Bárány naturally provides an algorithm computing a solution to this problem. This algorithm, known as the Bárány-Onn algorithm, was analyzed in [4]. It is a pivot algorithm roughly relying on computing the closest facet of a simplex to $\mathbf{0}$. Although not polynomial, this algorithm is quite efficient, as proved by Deza et al. through an extensive computational study [8].

In addition to the TFNP version of Colorful Linear Programming, Bárány and Onn formulated the following problem, which is in a sense more general.

Colorful Linear Programming
Input. A configuration of k sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_k$ in $\mathbb{Q}^d$.
Task. Decide whether there exists a positively dependent colorful set for this configuration of points. If there is one, find it.

Bárány and Onn showed that the case of Colorful Linear Programming with $k = d$ is NP-complete even if each $\mathbf{S}_i$ is of size 2, proving that the general case is NP-complete as well. It contrasts with the TFNP version of Colorful Linear Programming. In this version, when each $\mathbf{S}_i$ is of size 2, we clearly have a polynomial special case: select one point in each $\mathbf{S}_i$, find the linear dependency, and change for the other point in $\mathbf{S}_i$ for those having a negative coefficient.

A slightly more general version of Colorful Linear Programming can be defined with conic hulls instead of convex hulls.

Colorful Linear Programming (conic version)
Input. A configuration of k sets of points $\mathbf{S}_1, \ldots, \mathbf{S}_k$ in $\mathbb{Q}^d$ and an additional point $\boldsymbol{p}$ in $\mathbb{Q}^d$.
Task. Decide whether there exists a colorful set T such that $\boldsymbol{p} \in \operatorname{cone}(T)$.

By an easy geometric argument, this problem coincides with Colorful Linear Programming when $\mathrm{conv}(\{\boldsymbol{p}\} \cup \bigcup_{i=1}^{k} \mathbf{S}_i)$ does not contain $\mathbf{0}$. Note that as usual for this kind of problems, there is a shift in the dimension when going to one version to the other.

1.2. **Main contributions.** We show that Colorful Linear Programming is NP-complete even if $k-d$ is fixed and each $\mathbf{S}_i$ is of size 2. The complexity status of the case $k-d=1$ was a question of Bárány and Onn (Section 5 of [4]). As a by-product of this complexity result, we get a new proof of the coNP-completeness of deciding whether a polytope is the projection of another polytope, both being described by systems of linear inequalities. These results are stated and proved in Section 2.

Section 3 is devoted to a link we found between Colorful Linear Programming and Bimatrix, which is the problem of finding a Nash equilibrium in a bimatrix game. We exhibit a polynomial reduction of Bimatrix to Colorful Linear Programming. Bimatrix being PPAD-complete, it shows that any NP-complete problem is at least as hard as any PPAD-complete problem. It has already been noted that P=NP would imply P=PPAD, see [24]. We give here a concrete example of a reduction of a PPAD-complete problem to an NP-complete problem. We do not know whether such an example was already known. On our way we present a general method to prove that a complementarity problem belongs to the PPAD class, based on Sperner's lemma.

In Section 4, we give a new version of the Bárány-Onn algorithm, replacing the search of a closest facet by a classical reduced cost consideration. We get in this way an algorithm similar to the "Phase I" simplex method. Numerical performances of this approach are provided.

We end the paper with a study of special cases and analogues of colorful linear programming in combinatorics. In particular, two combinatorial and polynomial cases of the TFNP version of Colorful Linear Programming are presented.

2. Complexity of colorful linear programming

2.1. **Proof of NP-completeness.** For a fixed $q \in \mathbb{Z}$, we define $\mathrm{CLP}(q)$ to be the Colorful Linear Programming problem with the additional constraint that $k-d=q$.

Lemma 1. *If* $\mathrm{CLP}(q)$ *is* NP-*complete, then* $\mathrm{CLP}(q-1)$ *is also* NP-*complete.*

Proof. Let $\mathbf{S}_1, \ldots, \mathbf{S}_k$ in $\mathbb{R}^d$ be an instance with $k = d+q$. Define $d' = d+1$. Embedding this instance in $\mathbb{R}^{d'}$ by adding a d'th component equal to 0, we get an instance with $k = d'+q-1$, every solution of which provides a solution for the case $k = d+q$, and conversely. This latter case being NP-complete, we get the conclusion. □ □

Lemma 2. *If* $\mathrm{CLP}(q)$ *is* NP-*complete, then* $\mathrm{CLP}(q+1)$ *is also* NP-*complete.*

Proof. Let $\mathbf{S}_1, \ldots, \mathbf{S}_k$ in $\mathbb{R}^d$ be an instance with $k = d+q$. Define $d' = d+1$ and $k' = k+2$. Embed this instance in $\mathbb{R}^{d'}$ by adding a d'th component equal to 0. Add two sets $\mathbf{S}_{k+1}$ and $\mathbf{S}_{k+2}$ entirely located at coordinate $(0, \ldots, 0, 1)$. We have thus an instance with $k' = d'+q+1$, every solution of which provides a solution for the case $k = d+q$, and conversely. This latter case being NP-complete, we get the conclusion. □ □

We have the following theorem.

Theorem 2. CLP(q) *is* NP*-complete for any fixed* $q \in \mathbb{Z}$*.*

In particular, it is NP-complete for $q = 1$.

Proof of Theorem 2. CLP(0) is NP-complete according to Theorem 6.1 in [4]. Lemmas 1 and 2 allow to conclude. □ □

Polynomially checkable sufficient conditions ensuring the existence of a positively dependent colorful set exist: the condition of the colorful Carathéodory theorem is one of them. More general polynomially checkable sufficient conditions when $k = d + 1$ are given in [1, 10, 21]. However, the fact that CLP(1) is NP-complete implies that there are no polynomially checkable conditions that are simultaneously sufficient and necessary for a positively dependent colorful set to exist when $k = d + 1$, unless P=NP.

Remark 1. The instances built in the proof of Lemma 2 are not in general position, since $\mathbf{0}$ and the $\mathbf{S}_i$'s with $i \leq k$ are all in a same hyperplane. By *general position*, we mean no $d' + 1$ points in a same $(d' - 1)$-dimensional affine subspace. We could wonder whether the case $k = d + 1$ remains NP-complete when the points are in general position. The answer is yes, and we explain how to reduce the instance built in the proof of Lemma 2 to an instance in general position.

First, the sets $\mathbf{S}_{k+1}$ and $\mathbf{S}_{k+2}$ can be slightly perturbed without changing the conclusion. Second, we slightly move $\mathbf{0}$ into one of the halfspaces delimited by the hyperplane containing the $\mathbf{S}_i$'s for $i \leq k$. We choose the halfspace containing $\mathbf{S}_{k+1}$ and $\mathbf{S}_{k+2}$. This move must be sufficiently small so that $\mathbf{0}$ does not traverse another hyperplane generated by d' points in $\bigcup_{i=1}^{k+2} \mathbf{S}_i$. All coordinates being rational, Cramer's formula allows to compute a length of the displacement that ensures this condition. Third, we move each point of the $\bigcup_{i=1}^{k} \mathbf{S}_i$ independently along a line originating from $\mathbf{0}$.

2.2. **Projection and colorful linear programming.** Algorithmic questions related to projecting polytopes are usually identified as difficult questions. Tiwary [27] recently showed that given two polytopes $\mathcal{Q}$ and $\mathcal{Q}'$ described by systems of linear inequalities, deciding whether $\mathcal{Q}$ is a projection of $\mathcal{Q}'$ is coNP-complete. Note that it is in coNP since deciding whether a partial solution of a system of linear inequalities can be extended to a full solution is a linear programming problem. His proof of coNP-completeness uses a reduction of the problem of deciding whether a polytope described by its facets is contained in a polytope described by its vertices, which is a coNP-complete problem [9]. Colorful Linear Programming is another way to prove this result.

Take any instance $\mathbf{S}_1, \ldots, \mathbf{S}_d, \boldsymbol{p}$ of the conic version of Colorful Linear Programming, all points being in general position, with $\operatorname{conv}(\{\boldsymbol{p}\} \cup \bigcup_{i=1}^{d+1} \mathbf{S}_i)$ not containing $\mathbf{0}$. Because of Theorem 2 and Remark 1, the problem of deciding whether there is colorful solution is NP-complete. We show how to reduce this instance to an instance of a the aforementioned polytope projection problem. We define A_i to be the matrix with the columns being the vectors in $\mathbf{S}_i$, for $i = 1, \ldots, d$. We define then the following polytopes:

$$\mathcal{P} = \left\{ \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d) \in \mathbb{R}_+^{d^2} : \sum_{i=1}^{d} A_i \boldsymbol{x}_i = \boldsymbol{p} \right\}$$

and

$$\mathcal{P}_i = \{\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d) \in \mathcal{P} : \boldsymbol{x}_i = \boldsymbol{0}\}.$$

They are polytopes because of the assumption $\operatorname{conv}(\{\boldsymbol{p}\} \cup \bigcup_{i=1}^{d+1} \mathbf{S}_i)$ does not contain $\mathbf{0}$.

There exists a colorful solution to the Colorful Linear Programming problem we consider if and only if $\mathcal{P} \setminus \operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is nonempty. Indeed, if there exists a colorful solution to the Colorful Linear Programming problem, the latter set in nonempty: a colorful solution provides a point (a basis in the linear programming terminology) in $\mathcal{P}$ with each $\boldsymbol{x}_i$ being nonzero, because of the genericity assumption. Conversely, if the set is nonempty, there is a vertex of $\mathcal{P}$ not in $\operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$, and such a vertex has exactly d nonzero components, each corresponding to a column of a distinct A_i, and provides a solution to the Colorful Linear Programming problem we consider.

Deciding whether $\mathcal{P} \setminus \operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is nonempty is therefore NP-hard. We prove below that $\operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is a projection of some higher dimensional polytope $\mathcal{Q}'$. Hence, deciding whether $\mathcal{P} \setminus \operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is nonempty is equivalent to deciding whether $\mathcal{P}$ is the projection of $\mathcal{Q}'$.

The polytope $\operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is described by the solutions $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ satisfying the following system of linear equalities and inequalities:

$$\left\{\begin{array}{rcll} \displaystyle\sum_{j=1}^{d} A_i \boldsymbol{x}_{ij} - y_i \boldsymbol{p} & = & \mathbf{0} & \forall i \\ \displaystyle\sum_{i=1}^{d} \boldsymbol{x}_{ij} & = & \boldsymbol{x}_j & \forall j \\ \displaystyle\sum_{i=1}^{d} y_i & = & 1 & \\ \boldsymbol{x}_{ii} & = & \mathbf{0} & \forall i \\ y_i & \in & \mathbb{R}_+ & \forall i \\ \boldsymbol{x}_{ij} & \in & \mathbb{R}_+^d & \forall i, j. \end{array}\right.$$

Indeed, a point $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d) \in \operatorname{conv}(\bigcup_{i=1}^{d} \mathcal{P}_i)$ is such that $\boldsymbol{x} = \sum_{i=1}^{d} y_i \boldsymbol{x}'_i$, with $\sum_{i=1}^{d} y_i = 1$ and $\boldsymbol{x}'_i = (\boldsymbol{x}'_{i1}, \ldots, \boldsymbol{x}'_{id}) \in \mathcal{P}_i$ for each i. Defining $\boldsymbol{x}_{ij}$ to be $y_i \boldsymbol{x}'_{ij}$ shows that such an $\boldsymbol{x}$ satisfies the system. Conversely, a solution of the system induces a point $\boldsymbol{x}$ that can be written as $\sum_{i=1}^{d} y_i \boldsymbol{x}'_i$ with $\boldsymbol{x}'_i \in \mathcal{P}_i$ for all i. Indeed, define $\boldsymbol{x}'_{ij} = \frac{1}{y_i} \boldsymbol{x}_{ij}$ when $y_i \neq 0$, and $\boldsymbol{x}'_{ij} = \mathbf{0}$ otherwise. In this latter case, all the $\boldsymbol{x}_{ij}$'s are equal to $\mathbf{0}$ because of the assumption $\mathbf{0} \notin \operatorname{conv}(\{\boldsymbol{p}\} \cup \bigcup_{i=1}^{d+1} \boldsymbol{S}_i)$.

3. Links with Nash equilibria

3.1. **Another problem.** The link between Nash equilibria and Colorful Linear Programming relies on the study of another problem similar to Colorful Linear Programming. This problem was proposed by Meunier and Deza [21] as a byproduct of an existence theorem, the Octahedron lemma [3, 7], which by some features has a common flavor with the colorful Carathéodory theorem. The Octahedron lemma states that if each $\mathbf{S}_i$ of the configuration is of size 2 and if the points are in general position, the number of positively dependent colorful sets is even. The problem we call Finding Another Colorful Simplex is the following.

Finding Another Colorful Simplex
Input. A configuration of $d+1$ pairs of points $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ in $\mathbb{Q}^d$ and a positively dependent colorful set in this configuration.
Task. Find another positively dependent colorful set.

Another positively dependent colorful set exists for sure. Indeed, by a slight perturbation, we can assume that all points are in general position. If there were only one positively dependent colorful set, there would also be only one positively dependent colorful set in the perturbed configuration, which violates the evenness property stated by the Octahedron lemma. In their paper, Meunier and Deza question the complexity status of this problem. We solve the question by proving that it is actually a generalization of the problem of computing a Nash equilibrium in a bimatrix game.

3.2. Finding Another Colorful Simplex **is in** PPAD. In [21], it was noted that Finding Another Colorful Simplex is in PPA. The class PPA, also defined by Papadimitriou in 1994 [23], contains the class PPAD. PPA contains the problems that can be polynomially reduced to the problem of finding another degree 1 vertex in a graph whose vertices all have degree at most 2 and in which a degree 1 vertex is already given. The graph is supposed to be implicitly described by the neighborhood function, which, given a vertex, returns its neighbors in polynomial time. The PPAD class is the subclass of PPA for which the implicit graph is oriented and such that each vertex has an outdegree at most 1 and an indegree at most 1. The problem becomes then: given an *unbalanced* vertex, that is a vertex v such that $\deg^+(v) + \deg^-(v) = 1$, find another unbalanced vertex. See [23] for further precisions.

We prove in this subsection that Finding Another Colorful Simplex is in PPAD. We proceed by showing that the existence of the other positively dependent colorful set is a consequence of Sperner's lemma [26]. Our method for proving that Finding Another Colorful Simplex belongs to PPAD is adaptable for other complementarity problems, among them Bimatrix. We believe that our method is new. It avoids the use, as in [6, 13, 23, 29], of *oriented primoids* or *oriented duoids* defined by Todd [28].

One of the multiple versions of Sperner's lemma is the following theorem, proposed by Scarf [25], which involves a triangulation of a sphere, whose vertices are labeled. A simplex whose vertices get pairwise distinct labels is said to be *fully-labeled*.

Theorem 3 (Sperner's lemma)**.** *Let* T *be a triangulation of an* n*-dimensional sphere* $\mathcal{S}^n$ *and let* V *be its vertex set. Assume that the elements of* V *are labeled according to a map* $\lambda : V \to E$*, where* E *is some finite set. If* E *is of cardinality* $n+1$*, then there is an even number of fully-labeled* n*-simplices.*

We state now the main proposition of this subsection.

Proposition 1. Finding Another Colorful Simplex *is in* PPAD.

Proof. By a perturbation argument, we can assume the points to be in general position, see [19] for instance for a description of such a polynomial-time computable perturbation. The proof consists then in proving the existence of another positively dependent colorful set via a polynomial reduction to Sperner's lemma.

We define a simplicial complex K with vertex set $\bigcup_{i=1}^{d+1} \mathbf{S}_i$:

$$\mathsf{K} = \{\sigma \subseteq \bigcup_{i=1}^{d+1} \mathbf{S}_i : \bigcup_{i=1}^{d+1} \mathbf{S}_i \setminus \sigma \text{ is positively dependent}\}.$$

Since any superset of a positively dependent set is a positively dependent set, K is a simplicial complex. The points being in general position, the dimension of K is $2(d+1)-(d+1)-1 = d$. Actually, K is a triangulation of $\mathcal{S}^d$. It can be seen using Gale transform and Corollary 5.6.3 (iii) of [18].

Now, for v a vertex of K, define $\lambda(v)$ to be its color, i.e. the index i such that $v \in \mathbf{S}_i$. Any fully-labeled simplex σ of K is such that $\bigcup_{i=1}^{d+1} \mathbf{S}_i \setminus \sigma$ is a positively dependent colorful set and conversely. There is thus an explicit one-to-one correspondence between the fully-labeled simplices of K and the positively dependent colorful set. Applying Theorem 3 (Sperner's lemma) with $\mathsf{T} = \mathsf{K}$, $n = d$, and $E = \{1, \ldots, d+1\}$ shows that there is an even number of fully-labeled simplices in K, and hence, an even number of positively dependent colorful sets. Since there is a proof of Sperner's lemma using an oriented path-following argument [20, 25] and since the triangulation here can easily be encoded by a Turing machine computing the neighbors of any simplex in the triangulation in polynomial time, FINDING ANOTHER COLORFUL SIMPLEX is in PPAD. □ □

3.3. **Reduction of** BIMATRIX**.** A bimatrix game involves two $m \times n$ matrices with real coefficients $A = (a_{ij})$ and $B = (b_{ij})$. There are two players. The first player chooses a probability distribution on $\{1, \ldots, m\}$, the second a probability distribution on $\{1, \ldots, n\}$. Once these probability distributions have been chosen, a pair $(\bar{i}, \bar{j})$ is drawn at random according to these distributions. The first player gets a payoff equal to $a_{(\bar{i},\bar{j})}$ and the second a payoff equal to $b_{(\bar{i},\bar{j})}$. A Nash equilibrium is a choice of distributions in such a way that if a player changes his distribution, he will not get in average a strictly better payoff.

Let Δ^k be the set of vectors $\boldsymbol{x} \in \mathbb{R}_+^k$ such that $\sum_{i=1}^k x_i = 1$. Formally, a *Nash equilibrium* is a pair $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ with $\boldsymbol{y}^* \in \Delta^m$ and $\boldsymbol{z}^* \in \Delta^n$ such that

$$\text{(1)} \qquad \boldsymbol{y}'^T A \boldsymbol{z}^* \leq \boldsymbol{y}^{*T} A \boldsymbol{z}^* \text{ for all } \boldsymbol{y}' \in \Delta^m \quad \text{and} \quad \boldsymbol{y}^{*T} B \boldsymbol{z}' \leq \boldsymbol{y}^{*T} B \boldsymbol{z}^* \text{ for all } \boldsymbol{z}' \in \Delta^n.$$

It is well-known that if the matrices have rational coefficients, there is a Nash equilibrium with rational coefficients, which are not too large with respect to the input. BIMATRIX is the following problem: given A and B with rational coefficients, find a Nash equilibrium. Papadimitriou showed in 1994 that BIMATRIX is in PPAD [23]. Later, Chen, Deng, and Teng [6] proved its PPAD-completeness.

A combinatorial approach to these equilibria consists in studying the *complementary* solutions of the two systems

$$\text{(2)} \qquad [A, I_m]\boldsymbol{x} = (1, \ldots, 1)^T \text{ and } \boldsymbol{x} \in \mathbb{R}_+^{n+m}$$

and

$$[I_n, B^T]\boldsymbol{x} = (1,\ldots,1)^T \text{ and } \boldsymbol{x} \in \mathbb{R}_+^{n+m}. \tag{3}$$

By *complementary solutions*, we mean a solution $\boldsymbol{x}_A$ of (2) and a solution $\boldsymbol{x}_B$ of (3) such that $\boldsymbol{x}_A \cdot \boldsymbol{x}_B = 0$. Indeed, complementary solutions with $\mathrm{supp}(\boldsymbol{x}_A) \neq \{n+1,\ldots,n+m\}$ or $\mathrm{supp}(\boldsymbol{x}_B) \neq \{1,\ldots,n\}$ give a Nash equilibrium. This point of view goes back to Lemke and Howson [16]. A complete proof within this framework can be found in Remark 6.1 of [20].

We derive the difficulty of Finding Another Colorful Simplex from the complexity of Bimatrix.

Proposition 2. Finding Another Colorful Simplex *is* PPAD*-complete.*

Proof. We prove that the following version of Finding Another Colorful Simplex with cones is PPAD-complete. This version is equivalent to Finding Another Colorful Simplex.

Finding Another Colorful Cone
Input. A configuration of $d+1$ pairs of points $\mathbf{S}_1,\ldots,\mathbf{S}_{d+1}$ in $\mathbb{Q}^{d+1}$, an additional point $\boldsymbol{p}$ in $\mathbb{Q}^{d+1}$ such that $\mathrm{conv}(\{\boldsymbol{p}\} \cup \bigcup_{i=1}^{d+1} \mathbf{S}_i)$ does not contain $\mathbf{0}$, and a colorful set T such that $\boldsymbol{p} \in \mathrm{cone}(T)$.
Task. Find another colorful set T' such that $\boldsymbol{p} \in \mathrm{cone}(T')$.

The proof uses a reduction of Bimatrix to Finding Another Colorful Cone. Consider an instance of Bimatrix. First note that we can assume that all coefficients of A and B are positive. Indeed, adding a same constant to all entries of the matrices does not change the game. Build the $(m+n) \times (2(m+n))$ matrix

$$M = \begin{pmatrix} A & I_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n & B^T \end{pmatrix}.$$

We denote by M_i the ith column of M. Note that the vector $\boldsymbol{u} = (1,\ldots,1) \in \mathbb{R}^{n+m}$ is in the conic hull of $T = \{M_{n+1},\ldots,M_{n+m},M_{n+m+1},\ldots,M_{2n+m}\}$. Indeed, the corresponding submatrix is the identity matrix.

Let $\mathbf{S}_i$ be the pair $\{M_i, M_{m+n+i}\}$ for $i = 1,\ldots,m+n$. Since all coefficients of A and B are positive, $\mathbf{0}$ is not in the convex hull of the columns of M and $\boldsymbol{u}$. A polynomial time algorithm solving Finding Another Colorful Simplex with T as input set would find another colorful set T' such that $\boldsymbol{u} \in \mathrm{cone}(T')$. The decomposition of $\boldsymbol{u}$ on the points in T' gives a vector $\boldsymbol{x}$ such that $M\boldsymbol{x} = \boldsymbol{u}$, $x_i x_{m+n+i} = 0$ for $i = 1,\ldots,m+n$, and $\mathrm{supp}(\boldsymbol{x}) \neq \{n+1,\ldots,2n+m\}$. Such an $\boldsymbol{x}$ can be written $(\boldsymbol{x}_A, \boldsymbol{x}_B)$ with $\boldsymbol{x}_A, \boldsymbol{x}_B \in \mathbb{R}_+^{m+n}$ satisfying $\boldsymbol{x}_A \cdot \boldsymbol{x}_B = 0$ and such that $\mathrm{supp}(\boldsymbol{x}_A) \neq \{n+1,\ldots,n+m\}$ or $\mathrm{supp}(\boldsymbol{x}_B) \neq \{1,\ldots,n\}$. In other words, it would find a Nash equilibrium. Bimatrix being PPAD-complete, Proposition 1 implies therefore that Finding Another Colorful Simplex is PPAD-complete. □ □

This proof shows that Finding Another Colorful Simplex is more general than computing complementary solutions of Equations (2) and (3). In [21], a pivoting algorithm for solving Finding Another Colorful Simplex is proposed. It reduces to the classical pivoting algorithm due to Lemke and Howson [16] used for computing such complementary solutions.

Remark 2 (Complexity of Sperner's lemma). In the proof of Proposition 1, we reduce Finding Another Colorful Simplex to the following Sperner-type problem; let T be a triangulation of the d-dimensional sphere involving $2(d+1)$ vertices and let $\lambda : V(\mathsf{T}) \to \{1, \ldots, d+1\}$ be a labeling; assume given a fully-labeled simplex; find another fully-labeled simplex. Proposition 2 shows that this problem is actually PPAD-complete, even if each label appears exactly twice. Sperner-type problems have already been proved to be PPAD-complete [5, 23], but these latter problems are in fixed dimension, with an exponential number of vertices, and with a labeling given by an oracle, while the Sperner-type problem we introduce has an explicit description of the vertices and of the labeling. Note that the number of vertices is small. A computational problem with similar features has been proposed in a paper by Király and Pap [14], but it involves a polyhedral version of Sperner's lemma distinct from the classical Sperner's lemma. Remark 4 is Section 5 will exhibit some polynomial cases of the Sperner-type problem we introduce here.

3.4. **Reduction of** Bimatrix **to** Colorful Linear Programming**.** Let $\mathcal{A}$ be an algorithm solving Colorful Linear Programming. Note that we refer here to the decision problem. In this subsection, we describe an algorithm solving Finding Another Colorful Simplex, and therefore Bimatrix because of the reduction described in Section 3.3, by calling exactly $d+1$ times $\mathcal{A}$. We get this way a polynomial reduction of Bimatrix to Colorful Linear Programming. We are probably not aware of the good references, but we were not able to find another such problem in the literature giving a concrete illustration of the fact that NP-complete problems are harder than PPAD problems.

We describe now the algorithm for Finding Another Colorful Simplex. The input is given by the $d+1$ pairs of points $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ and the positively dependent colorful set T. The algorithm selects successively a point in each of the $\mathbf{S}_i$'s. Each iteration consists in testing with the help of $\mathcal{A}$ which point of $\mathbf{S}_i$ is in a positively dependent colorful set compatible with the already selected points and in selecting such a point, with the priority given to $\mathbf{S}_i \setminus T$. A typical iteration is

> Define $\mathbf{S}_i' := \mathbf{S}_i \setminus T$; apply $\mathcal{A}$ to $\mathbf{S}_1', \ldots, \mathbf{S}_i', \mathbf{S}_{i+1}, \ldots, \mathbf{S}_{d+1}$; if the answer is 'no', define instead $\mathbf{S}_i' := \mathbf{S}_i \cap T$.

At the end, the algorithm outputs $\bigcup_{i=1}^{d+1} \mathbf{S}_i'$.

Since we know that there is another positively dependent colorful set, the answer will be 'yes' for at least one i. The returned colorful simplex is therefore a positively dependent colorful set distinct from T. This algorithm returns another positively dependent colorful set after calling $d+1$ times $\mathcal{A}$.

Remark 3. The same approach shows that the TFNP version of Colorful Linear Programming is polynomially reducible to the general version of Colorful Linear Programming.

4. Simplexification of Bárány-Onn algorithm

4.1. **Algorithm.** Recall that the colorful Carathéodory theorem states that when each of the $\mathbf{S}_i$'s is positively dependent and $k = d+1$, there exists a positively dependent colorful set. The pivoting algorithm proposed by Bárány and Onn for finding a positively dependent colorful set under these conditions goes roughly as follows. The input is the sets $\mathbf{S}_1, \ldots, \mathbf{S}_{d+1}$ of points in $\mathbb{Q}^d$, each of cardinality $d+1$ and positively dependent. All points are moreover assumed to be in general position.

Bárány-Onn algorithm

- Choose a first colorful set T_1 of size $d+1$ and let $i := 0$.
- Repeat:
  - Let $i := i+1$.
  - If $\mathbf{0} \in \operatorname{conv}(T_i)$, stop and output T_i.
  - Otherwise, find $F_i \subseteq T_i$ of cardinality d such that $\operatorname{aff}(F_i)$ separates $T_i \setminus F_i$ from $\mathbf{0}$; choose in the half-space containing $\mathbf{0}$ a point $\boldsymbol{t}$ of the same color as the singleton $T_i \setminus F_i$; define $T_{i+1} := F_i \cup \{\boldsymbol{t}\}$.

Since each $\operatorname{conv}(\mathbf{S}_i)$ contains $\mathbf{0}$, there is always a point of each color in the half-space delimited by $\operatorname{aff}(F_i)$ and containing $\mathbf{0}$. It explains that a point $\boldsymbol{t}$ as in the algorithm can always be found as long as the algorithm has not terminated.

The technical step is the way of finding the subset F_i and requires a distance computation or a projection [4], or the computation of the intersection of a fixed ray and $\operatorname{conv}(T_i)$ [21]. Deza et al. [8] proceed to an extensive computational study of algorithms solving this problem, with many computational experiments. In addition to some heuristics, "multi-update" versions are also proposed, but they do not avoid this kind of operations.

We propose to modify the approach as follows. We add a dummy point $\boldsymbol{v}$ and define the following optimization problem.

$$\begin{array}{rl} \min & z \\ \text{s.t.} & A\boldsymbol{\lambda} + z\bar{\boldsymbol{v}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\ & \boldsymbol{\lambda} \geq \mathbf{0},\ z \geq 0, \end{array}$$

where $\bar{\boldsymbol{v}} = (\boldsymbol{v}, 1)$ and where A is the $(d+1) \times (d+1)^2$ matrix whose columns are the points of $\bigcup_{i=1}^{d+1} \mathbf{S}_i$ with an additional 1 on the $(d+1)$th row. This optimization problem simply looks for an expression of $\mathbf{0}$ as a convex combination of the points in $\{\boldsymbol{v}\} \cup \bigcup_{i=1}^{d+1} \mathbf{S}_i$ with a minimal weight on $\boldsymbol{v}$. Especially, if $\mathbf{0} \in \operatorname{conv}(\bigcup_i \mathbf{S}_i)$, the optimal value is 0. The idea consists in seeking an optimal basis, with the terminology of the linear programming, which in addition is required to be colorful. The colorful Carathéodory theorem ensures that such a basis exists.

Now, choose a first *transversal* F_1, which is a colorful set of cardinality d. Choose the dummy point $\boldsymbol{v}$ so that $F_1 \cup \{\boldsymbol{v}\}$ contains $\mathbf{0}$ in the interior of its convex hull. Note that $F_1 \cup \{\boldsymbol{v}\}$ is a feasible basis. The algorithm proceeds with simplex pivots, going from feasible colorful basis to feasible colorful basis, until an optimal colorful basis is found. We start with $i := 0$. We repeat then

– Let $i := i + 1$.
– Choose a point $\boldsymbol{t}$ of the missing color in F_i with negative reduced cost. The reduced costs are computed according to the current basis $F_i \cup \{\boldsymbol{v}\}$.
– Proceed to a simplex pivot operation with $\boldsymbol{t}$ entering the current basis.
– If $\boldsymbol{v}$ leaves the basis, stop and output $F_i \cup \{\boldsymbol{t}\}$ (it is an optimal colorful basis).
– Otherwise, define F_{i+1} to be the new basis minus $\boldsymbol{v}$.

This algorithm eventually finds a positively dependent colorful set because of the following lemma. The remaining arguments are exactly the same as above: as long as a positively dependent colorful set has not been found, there is a point of the missing color in the half-space delimited by $\operatorname{aff}(F_i)$ and containing $\mathbf{0}$.

Lemma 3. *The points in the half-space delimited by* $\operatorname{aff}(F_i)$ *and containing* $\mathbf{0}$ *are precisely the points with a negative reduced cost.*

Proof. Let $F_i = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d\}$ and let $\boldsymbol{t}$ be any other point in $\left(\bigcup_{j=1}^{d+1} \mathbf{S}_j\right) \setminus F_i$. Consider $x_1, \ldots, x_d, r, s \in \mathbb{R}$ such that

$$r\boldsymbol{t} + s\boldsymbol{v} + \sum_{i=1}^{d} x_i \boldsymbol{u}_i = \mathbf{0}, \tag{4}$$

with $r > 0$ and $r + s + \sum_{i=1}^{d} x_i = 0$. We have $s \neq 0$ by genericity assumption. The reduced cost of $\boldsymbol{t}$ is exactly s/r. Therefore, proving the lemma amounts to prove that s is negative exactly when $\boldsymbol{t}$ is in the half-space delimited by $\operatorname{aff}(F_i)$ and containing $\mathbf{0}$.

To see this, note that Equation (4) can be rewritten

$$r(\boldsymbol{t} - \boldsymbol{u}_1) + s(\boldsymbol{v} - \boldsymbol{u}_1) + \sum_{i=2}^{d} x_i(\boldsymbol{u}_i - \boldsymbol{u}_1) = \mathbf{0}. \tag{5}$$

Now, take the unit vector $\boldsymbol{n}$ orthogonal to $\operatorname{aff}(F_i)$ and take the scalar product of Equation (5) and $\boldsymbol{n}$. It gives

$$r\boldsymbol{n} \cdot (\boldsymbol{t} - \boldsymbol{u}_1) + s\boldsymbol{n} \cdot (\boldsymbol{v} - \boldsymbol{u}_1) = 0$$

and the conclusion follows since $\boldsymbol{v}$ and $\mathbf{0}$ are in the same half-space delimited by $\operatorname{aff}(F_i)$. □

□

This approach is reminiscent of the "Phase I" simplex method, which computes a first feasible basis by solving an auxiliary linear program whose optimal value is 0 on such a basis.

4.2. **Numerical results.** We implemented our algorithm in C++. The tests are performed on a PC Intel® Core™ i3-2310M, with two 64-bit CPUs, clocked at 2.1 GHz, with 4 GB RAM. The instances are provided by five random generators, implemented by Huang in MATLAB. All the generators provide instances of $(d+1)^2$ points in general position on the unit sphere, partitioned into $d+1$ colors and such that the origin $\mathbf{0}$ is in the convex hull of each color. Descriptions of the generators can be found in [11]. At each iteration, we choose the entering point $\boldsymbol{t}$ that has the most negative reduced cost.

Table 1 presents the computational results on 50 instances by dimension and by generators. The columns "time" give the average execution time of the algorithm in milliseconds. The columns "# pivots" give the average number of pivots. The entry corresponding to the "tube" case in dimension 384 is empty, since we faced cycling behavior for some instances (we felt that adding anti-cycling pivot rules was not imperative for our experiments).

	Random		Tube		Highdensity		Lowdensity		Middensity	
Dimension	time	# pivots	time	# pivots	time	# pivots	time	# pivots	time	# pivots
3	0.0135	1.94	0.0123	2.02	0.0342	1.62	0.0138	2.32	0.0170	1.70
6	0.0180	3.38	0.0195	3.42	0.0474	1.98	0.0213	6.50	0.0164	2.88
12	0.0406	6.56	0.0396	7.68	0.0609	1.84	0.0591	19.00	0.0371	4.88
24	0.1433	13.76	0.1574	19.66	0.0871	1.94	0.2958	51.06	0.1123	9.62
48	0.9612	31.86	1.1684	43.88	1.1006	1.94	2.7946	133.70	0.7725	19.44
96	8.5069	76.42	11.2441	108.10	3.1116	1.92	28.5813	349.44	6.1306	39.46
192	8.1017	186.62	250.1050	284.96	21.6753	1.86	263.1400	831.26	50.2998	93.88
384	1111.5020	476.50			441.1310	2.00	5987.8880	2032.6	846.9148	279.12

Table 1. Average solution time and number of pivots for the simplex-like algorithm

We compare these results with those of the Bárány-Onn algorithm presented in the paper by Deza et al. [8] using the same generators. In general, our number of pivots is slightly larger than what they get. Regarding the time per iteration, it is hard to draw a conclusion since their implementation was done in MATLAB and since they used a different machine (a server with eight 64-bit CPUs, clocked at 2.6 GHz, with 64 GB RAM). However, the time per iteration of our algorithm is of the order of a thousand times smaller.

5. Special cases and analogues of colorful linear programming in combinatorics

We start this section with two combinatorial corollaries of the colorful Carathéodory theorem. For each of them, we provide a direct proof. In both cases, we show that the colorful set can be computed in polynomial time, and get thus special polynomial cases of the TFNP version of Colorful Linear Programming.

Proposition 3. *Let $D=(V,A)$ be a directed graph with n vertices. Let $C_1,\ldots,C_n$ be pairwise arc-disjoint circuits. Then there exists a circuit C sharing at most one arc with each of these C_i. Moreover, such a circuit can be computed in polynomial time.*

The existence of the colorful circuit as a consequence of the colorful Carathéodory theorem has already been noted and is attributed to Frank and Lovász [2].

Proof. We consider the bipartite graph with vertex classes V and $\{1,\ldots,n\}$, and in which edge vi exists if the vertex v belongs to C_i. If each $X\subseteq V$ touches at least $|X|$ distinct colors, Hall's marriage theorem ensures the existence of a perfect matching in the bipartite graph. We can thus select for each vertex $v\in V$ an arc a in $\delta^-(v)$ belonging to a distinct C_i. The subgraph induced by these arcs contains a circuit C as required.

Otherwise, there is a subset $X\subseteq V$ with a neighborhood in the bipartite graph of cardinality at most $|X|-1$. One can remove X from D and apply induction. Note that the existence of such an X can be decided in polynomial time by a classical maximum matching algorithm, which provides also the set X itself if it exists. □ □

The existence statement of the next proposition is a consequence of the conic version of the colorful Carathéodory theorem. We provide a direct proof based on a greedy algorithm.

Proposition 4. *Let $D = (V, A)$ be a directed graph with n vertices. Let s and t be two vertices, and $P_1, \ldots, P_{n-1}$ be pairwise arc-disjoint s-t paths. Then there exists an s-t path P sharing at most one arc with each P_i. Moreover, such a path can be computed in polynomial time.*

Proof. We build progressively an arborescence rooted at s.

We start with $X = \{s\}$. At each step, X is the set of vertices reachable from s in the partial arborescence. At step i, if X does not contain t, choose an arc a of P_i belonging to $\delta^+(X)$ and add to X the endpoint of a not yet in X. This arc exists since by direct induction X is of cardinality i at step i and the s-t path P_i leaves X. □ □

We end the paper by a brief survey of matroidal counterparts of colorful linear programming. The next proposition is common knowledge in combinatorics. It is a matroidal version of the colorful Carathéodory theorem (with an additional algorithmic result).

Proposition 5. *Let M be a matroid of rank d. Assume that the elements of M are colored in d colors. If there exists a monochromatic basis in each color, then there exists a colorful basis and this latter can be found by a greedy algorithm.*

A matroidal version of the Octahedron lemma stated in Section 3.1 also exists. It is due to Magnanti [17].

Proposition 6. *Let M be a matroid of rank d with no loops. Assume that the elements of M are colored in d colors and that the number of elements colored in each color is at least two. If there is a colorful basis, then there is another colorful basis and this latter can then be found in polynomial time.*

The proof by Magnanti is based on the matroid intersection algorithm due to Lawler [15]. The same algorithm shows that the matroidal version of Colorful Linear Programming, namely deciding whether there is a colorful basis in a matroid whose elements are colored, is polynomial.

Remark 4 (Back to Sperner's lemma). Remark 2 of Section 3.3 shows that even a very special case of Sperner's lemma already leads to a PPAD-complete problem. The matroidal counterpart of the Octahedron lemma implies that the problem becomes polynomial when the triangulation is the boundary of the *cross-polytope*. The cross-polytope is the convex hull of the vectors of the standard orthonormal basis and their negatives.

Proposition 7. *Let T be the boundary of the $(d+1)$-dimensional cross-polytope and let $\lambda : V(\mathsf{T}) \to \{1, \ldots, d+1\}$ be any labeling. Assume given a fully-labeled simplex. Another fully-labeled simplex can be computed in polynomial time.*

Proof. If a vertex has a label that appears only once on $V(\mathsf{T})$, we remove it and its antipodal, and work on the boundary of a cross-polytope with a dimension smaller by one. Solving this new problem leads to a solution for the original problem. We repeat this process until each label appears exactly twice. Now, note that the simplices of the boundary of a cross-polytope form the independents of a matroid (it is a partition matroid). Considering the labels as colors, the conclusion follows then from Proposition 6. □ □

With a similar proof (omitted), we also have the following proposition.

Proposition 8. *Let* T *be the boundary of the* $(d+1)$*-dimensional cross-polytope and let* $\lambda : V(\mathsf{T}) \to \{1, \ldots, d+1\}$ *be any labeling. Deciding whether there is a fully-labeled simplex can be done in polynomial time. Moreover, if there is such a fully-labeled simplex, it can be found in polynomial time as well.*

References

References

[1] Arocha, J. L., Bárány, I., Bracho, J., Fabila, R., Montejano, L., 2009. Very colorful theorems. Discrete and Computational Geometry 42, 142–154.
[2] Bárány, I., 1982. A generalization of Carathéodory's theorem. Discrete Mathematics 40, 141–152.
[3] Bárány, I., Matoušek, J., 2007. Quadratically many colorful simplices. SIAM Journal on Discrete Mathematics 21, 191–198.
[4] Bárány, I., Onn, S., 1997. Colourful linear programming and its relatives. Mathematics of Operations Research 22, 550–567.
[5] Chen, X., Deng, X., 2009. On the complexity of 2D discrete fixed point problem. Theoretical Computer Science 410, 4448–4456.
[6] Chen, X., Deng, X., Ten, S.-H., 2009. Settling the complexity of computing two-player Nash equilibria. Journal of the ACM 56, Article No. 14.
[7] Deza, A., Huang, S., Stephen, T., Terlaky, T., 2006. Colourful simplicial depth. Discrete and Computational Geometry 35, 597–604.
[8] Deza, A., Huang, S., Stephen, T., Terlaky, T., 2008. The colourful feasibility problem. Discrete Applied Mathematics 156, 2166–2177.
[9] Freund, R. M., Orlin, J. B., 1985. On the complexity of four polyhedral set containment problem. Mathematical Programming 33, 139–145.
[10] Holmsen, A. F., Pach, J., Tverberg, H., 2008. Points surrounding the origin. Combinatorica 28, 633–644.
[11] Huang, S., 2006. MATLAB code for colourful linear programming. URL `http://www.math.sfu.ca/~tamon/Software/CLP`
[12] Johnson, D. S., Papadimitriou, C. H., Yannakakis, M., 1988. How easy is local search. Journal of Computer and System Sciences 37, 79–100.
[13] Kintali, S., Poplawski, L. J., Rajaraman, R., Sundaram, R., Teng, S.-H., 2009. Reducibility among fractional stability problems. In: Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on. IEEE, pp. 283–292.
[14] Király, T., Pap, J., 2013. PPAD-completeness of polyhedral versions of Sperner's lemma. Discrete Mathematics 313, 1594–1599.
[15] Lawler, E. L., 1975. Matroid intersection algorithms. Mathematical Programming 9, 31–56.

[16] Lemke, C. E., Howson, Jr, J. T., 1964. Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 413–423.

[17] Magnanti, T. L., 1974. Complementary bases of a matroid. Discrete Mathematics 8, 355–361.

[18] Matoušek, J., 2002. Lectures on Discrete Geometry. Springer.

[19] Megiddo, N., Chandrasekaran, R., 1989. On the ε-perturbation method for avoiding degeneracy. Operations Research Letters 8, 305–308.

[20] Meunier, F., 2013. Computing and proving with pivots. RAIRO-Operations Research 47, 331–360.

[21] Meunier, F., Deza, A., 2013. A further generalization of the colourful Carathéodory theorem. Discrete Geometry and Optimization, Fields Institute Communications 69, 179–190.

[22] Mulzer, W., Stein, Y., March 2014. Complexity of finding nearest colorful polytopes. In: Proceedings of EuroCG.

[23] Papadimitriou, C. H., 1994. On the complexity of the parity argument and other inefficient proofs of existence. Journal of Computer and System Sciences 48, 498–532.

[24] Papadimitriou, C. H., 2007. The complexity of finding Nash equilibria. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. V. (Eds.), Algorithmic Game Theory. Cambridge University Press, pp. 29–51.

[25] Scarf, H., 1967. The approximation of fixed points of a continuous mapping. SIAM Journal on Applied Mathematics 15, 1328–1343.

[26] Sperner, E., 1928. Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes. Abh. Math. Sem. Univ. Hambourg 6, 265–272.

[27] Tiwary, H. R., 2012. Complexity of computing the projection of polytopes, in arXiv under the title “On Computing the Shadows and Slices of Polytopes”.

[28] Todd, M. J., 1976. Orientations in complementary pivot algorithms. Mathematics of Operations Research 1, 54–66.

[29] Végh, L. A., von Stengel, B., 2014. Oriented Euler complexes and signed perfect matchings. Mathematical Programming, Series B, 1–26.

Université Paris Est, CERMICS (ENPC), 6-8 avenue Blaise Pascal, Cité Descartes, F-77455 Marne-la-Vallée

E-mail address: frederic.meunier@enpc.fr, pauline.sarrabezolles@enpc.fr