RAIRO-Oper. Res. 47 (2013) 331–360
DOI: 10.1051/ro/2013042

RAIRO Operations Research
www.rairo-ro.org

COMPUTING AND PROVING WITH PIVOTS

Frédéric Meunier[1]

Abstract. A simple idea used in many combinatorial algorithms is the idea of *pivoting*. Originally, it comes from the method proposed by Gauss in the 19th century for solving systems of linear equations. This method had been extended in 1947 by Dantzig for the famous simplex algorithm used for solving linear programs. From since, a pivoting algorithm is a method exploring subsets of a ground set and going from one subset σ to a new one σ' by deleting an element inside σ and adding an element outside σ: $\sigma' = \sigma \setminus \{v\} \cup \{u\}$, with $v \in \sigma$ and $u \notin \sigma$. This simple principle combined with other ideas appears to be quite powerful for many problems. This present paper is a survey on algorithms in operations research and discrete mathematics using pivots. We give also examples where this principle allows not only to compute but also to prove some theorems in a constructive way. A formalisation is described, mainly based on ideas by Michael J. Todd.

Keywords. Combinatorial topology, complementarity problems, constructive proofs, pivoting algorithms.

Mathematics Subject Classification. 90C49, 05E45.

1. Introduction

1.1. Motivation

Pivoting is one of the oldest ideas used for an algorithm. The Gauss elimination method aims at solving linear systems of the form $A\boldsymbol{x} = \boldsymbol{b}$. This method uses *pivots*. The matrix A is progressively transformed into an upper triangular matrix with 1's on the diagonal. At any step of the algorithm, the pivot is the nonzero entry on the diagonal of the current matrix used to annul the entries below it.

Received June 21, 2013. Accepted July 1, 2013.

[1] Université Paris Est, CERMICS (ENPC), 77455 Marne-la-Vallée, France
frederic.meunier@enpc.fr

Article published by EDP Sciences

© EDP Sciences, ROADEF, SMAI 2013

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

A similar idea is used in the simplex algorithm solving linear programs. In the simplex algorithm, during the pivot operation, a column is added to a set called a *basis* and another column leaves the *basis*, leading to a fixed cardinality for the basis. In this framework, pivoting becomes a combinatorial operation. During the last decades, similar algorithms, maintaining a set of fixed size with elements entering and leaving, have been succesfully applied for other problems in mathematical programming. Their success relies on the simplicity of this idea, and also on some special structures of the problems that have been progressively revealed. Pivoting algorithms also appears naturally in some constructive proofs. One of the most famous example is Sperner's lemma providing a simple constructive proof of Brouwer's theorem and which, by a slight adaptation by Scarf, becomes a fully algorithmic proof of that same theorem with the help of pivot operations.

The purpose of our paper is to provide a survey of these various applications and to describe a simple framework inspired mainly by the work by Todd [63] and in which it is quite easy to describe the ideas of pivoting. This framework may help researchers to build their own pivoting algorithms for problems they meet. Some related open questions are also stated.

1.2. Some tools and notations

We give here the main definitions we need for the paper. Other definitions, less used, are introduced when needed.

1.2.1. Set, vectors, and support

For a positive integer n, we denote by $[n]$ the set $\{1, \dots, n\}$. Given vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2, \dots, \boldsymbol{a}_k$ in $\mathbb{R}^d$, their *conic hull* is the set of all vectors $\sum_{i=1}^k \lambda_i \boldsymbol{a}_i$ with $\lambda_i \geq 0$ for all i. Their *convex hull* is defined similarly, with the additional condition that $\sum_{i=1}^k \lambda_i = 1$. For a vector $\boldsymbol{x}$ in $\mathbb{R}^d$, the *support* of $\boldsymbol{x}$, denoted $\mathrm{supp}(\boldsymbol{x})$, is the set $\{i \in [d] : x_i \neq 0\}$.

1.2.2. Simplicial complexes

A useful notion in the context of algorithms using pivot operations is that of *(abstract) simplicial complex*. An abstract simplicial complex K is a collection of subsets of a given ground set V called its *vertex set* such that, for any $\sigma \in \mathsf{K}$ and $\tau \subseteq \sigma$, we have also $\tau \in \mathsf{K}$. It implies in particular that $\emptyset \in \mathsf{K}$. An element of K is called a *simplex* or a *face* of K. We decide to consider $\emptyset$ as a simplex, but not all textbooks make this choice.

The *dimension* of a simplex σ, denoted $\dim(\sigma)$, is its cardinality minus 1: $\dim(\sigma) = |\sigma| - 1$. A d-dimensional simplex is often simply called a *d-simplex*. The *dimension* of a simplicial complex K is the maximal dimension of its faces: $\max_{\sigma \in \mathsf{K}} \dim(\sigma)$.

An abstract simplicial complex all maximal faces (for inclusion) of which have same cardinality is said to be *pure*. A pure d-dimensional abstract simplicial complex M such that any face of dimension $d-1$ is contained in one or two faces

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

of dimension d is called a *pseudomanifold*. The *boundary* of the pseudomanifold, denoted $\partial\mathsf{M}$, is the collection of the $(d-1)$-dimensional faces contained in exactly one d-dimensional face.

A *geometric simplex* is the convex hull of affinely independent points. A collection of geometric simplices such that (i) the face of any of them is still in the collection and (ii) the intersection of any two geometric simplices is a face of both, is called a *geometric simplicial complex*. We allow a face to be empty. The union of all simplices of a geometric simplicial complex K is called its *polyhedron* and denoted $\|\mathsf{K}\|$.

The vertex sets of the geometric simplices of a geometric simplical complex form an abstract simplicial complex of same dimension. The geometric simplicial complex is said to be a *realization* of the abstract one. It is quite easy to show that any abstract simplicial complex has a geometric realization. Geometric and abstract simplicial complexes are thus close notions and the adjectives geometric and abstract are often omitted: the same simplicial complex can sometimes be considered as abstract and sometimes as geometric, the context being generally enough to decide which point of view is taken.

A *triangulation* of a topological space X is a geometric simplicial complex whose polyhedron is homeomorphic to X.

1.2.3. Graphs

A *graph* is a pair (V, E), where V is the *vertex set* – or *node set* – and E the *edge set* such that $E \subseteq \binom{V}{2}$ (set of unordered pairs of vertices). Using the traditional terminology, this definition means that we assume the graphs to be simple and without loop. An edge $e = uv \in E$ is *incident* to the vertices u and v. The number of edges incident to a vertex v is its *degree* and denoted $\deg(v)$. A path is a sequence

$$v_0, e_0, v_1, \ldots, e_{k-1}, v_k, \text{ with } e_i = v_i v_{i+1} \in E \text{ for all } i = 0, \ldots, k-1.$$

The quantity k is then its *length*. If all edges e_i are distinct, the path is said to be *simple*; if all vertices v_i are distinct, the path is said to be *elementary*.

A *directed graph* is a pair (V, A), where V is the *vertex set* – or *node set* – and A the *arc set* such that $A \subseteq V \times V$ (set of ordered pairs of vertices). Using the traditional terminology, this definition means that we assume the graphs to be simple. We allow *loop*, which are pairs of the form (v, v). An arc $a = (u, v) \in A$ is *incident* to the vertices u and v. A *directed path* is a sequence

$$v_0, a_0, v_1, \ldots, a_{k-1}, v_k, \text{ with } a_i = (v_i, v_{i+1}) \in A \text{ for all } i = 0, \ldots, k-1.$$

The quantity k is then its *length*. If all arcs a_i are distinct, the directed path is said to be *simple*; if all vertices v_i are distinct, the directed path is said to be *elementary*.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

1.3. Plan

The paper is structured as follows. Section 2 gives two main examples we use in the paper. We try to provide for each of them a full description, but we may omit some technical details. At this stage, we do not try to see the common structures underlying these two examples. Then, in Section 3, we describe combinatorial structures particularily suitable to formalize pivot operations. These structures are called *primoids* and *duoids*. In the same section, we reinterpret the examples in light of primoids and duoids and give other examples. Section 4 is devoted to special cases of pivoting algorithms, the so-called *complementary pivoting algorithms*, which appear when both a primoid and a duoid are present. Some of the previous examples fit into the framework of complementary pivoting algorithms. Complexity classes related to pivoting algorithm are given in Section 5. Section 6 gathers some generalizations and extensions of pivoting algorithms. We end with open questions (Sect. 7).

2. Two main examples using pivots

2.1. Simplex algorithm

A linear optimization problem consists in solving

$$\begin{aligned} \min\ & \boldsymbol{c}\cdot\boldsymbol{x} \\ \text{s.t. } & A\boldsymbol{x}=\boldsymbol{b} \\ & \boldsymbol{x}\in\mathbb{R}^n_+, \end{aligned} \tag{2.1}$$

where A is a $m\times n$ matrix of rank m. Linear optimization is one of the most important class of optimization problems. It is one of the core techniques in operations research and finds numerous applications in industry, logistics, supply-chain management, etc. It finds also many applications in other areas of mathematics, especially in combinatorics. For example of such applications, see the book by Chvátal [7] or the recent book by Matoušek and Gärtner [41].

One of the most popular algorithms to solve such a problem is the *simplex* algorithm designed in 1947 by Dantzig [11] and which works roughly as follows. We assume that $\boldsymbol{b}$ is *generic* with respect to A, *i.e.* any solution $\boldsymbol{x}$ of $A\boldsymbol{x}=\boldsymbol{b}$ has at least m nonzero components. It is not a restriction since one can slightly perturb $\boldsymbol{b}$ in such a way that if the problem (2.1) is feasible, the new problem obtained with the perturbed $\boldsymbol{b}$ is still feasible.

The simplex algorithm uses the notion of *feasible basis*. A feasible basis of the pair $(A,\boldsymbol{b})$ is an m-cardinality subset J of $[n]$ such that the $m\times m$ matrix $A_J:=(A_j)_{j\in J}$, where A_j is the jth column of A, is nonsingular and $A_J^{-1}\boldsymbol{b}\in\mathbb{R}^m_+$. A key property of linear optimization is that given a feasible basis J and $k\in[n]\setminus J$, there is at most one feasible basis in $J\cup\{k\}$ distinct from J. Given a feasible basis J and a $k\in[n]\setminus J$, a *pivot operation* consists in computing the other feasible basis in $J\cup\{k\}$ if there is one. If there is none, it implies that the polyhedron $\{\boldsymbol{x}\in\mathbb{R}^n_+ : A\boldsymbol{x}=\boldsymbol{b}\}$ is unbounded.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Under the generic assumption on $\boldsymbol{b}$, starting from a feasible basis and repeating pivot operations in a suitable way (more or less in a greedy way) lead eventually to the optimal solution of the problem if there is one. To prove this fact completely needs additional results from linear optimization, especially that there is always an optimal solution of the problem (2.1) of the form $A_J^{-1}\boldsymbol{b}$ with J a feasible basis. The missing details can be found in the books cited above.

2.2. Sperner lemma

Sperner's lemma is an important result in combinatorial topology. It was originally proposed by Sperner [60] to obtain a simple and constructive proof of Brouwer's fixed-point theorem stating that any continuous map from a finite-dimensional ball into itself has a fixed-point. Brouwer's fixed-point theorem has numerous applications in mathematics and economy. The relation between Sperner's lemma and Brouwer's theorem can be found for instance in the recent book by De Longueville [12]. The original proof by Sperner, even constructive, was not algorithmic. Motivated by concrete applications, Scarf, inspired by the Lemke algorithm, proposed later an algorithmic proof [58], which is actually a pivot-based one. It is an adaptation of his proof that is given below.

Sperner's lemma itself has many applications, in game theory or in combinatorics. Cloutier, Nyman, and Su [8] show for instance how to use it to prove the celebrated envy-free cake division theorem by Neyman [50]. They propose also an extension of this theorem in the case when there are two cakes or more. Aharoni and Hall [2] prove a very general sufficient condition for the existence of a system of distinct representatives for a family of hypergraphs. Another nice application of the Sperner lemma is a theorem due to Monsky [46]: any dissection of a rectangle into triangles of same area needs an even number of triangles.

Let T be a triangulation of a d-dimensional sphere $\mathcal{S}^d$ and let $\lambda : V(\mathsf{T}) \to \{0, 1, \ldots, d\}$ be a map assigning integers – or *labels* – between 0 and d to the vertices of the triangulation. One of the multiple versions of Sperner's lemma is the following theorem, proposed by Scarf [56]. A simplex of T whose vertices get pairwise distinct labels is said to be *rainbow*.

Theorem 2.1 (Sperner's lemma)**.** *Let T be a triangulation of a d-dimensional sphere $\mathcal{S}^d$ and let $\lambda : V(\mathsf{T}) \to \{0, 1, \ldots, d\}$. If there is a rainbow d-simplex, there is at least another rainbow d-simplex.*

There are several proofs of this theorem. A quick one is given by elementary algebraic topology. We describe it, assuming basic knowledges in algebraic topology, which may be found in the book by Munkres [48] for instance. The map λ induces a simplicial map from T to the d-dimensional simplex $\triangle^d$. Assume for a contradiction that there is exactly one rainbow d-simplex, then we would have, at the level of mod 2 chains, $\lambda_\#(T) = \triangle^d$, where T is the sum of all d-simplices of T. This would lead a nonzero morphism for the dth homology groups, which would contradict the vanishing homology of $\triangle^d$.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Algorithmic proof of Theorem 2.1. Let σ_0 be a rainbow d-simplex in T. Denote by v the vertex of σ_0 labeled by 0. Since T is a triangulation of a manifold, the $(d-1)$-simplex $\tau_0 = \sigma_0 \setminus \{v\}$ is contained in exactly one d-simplex distinct from σ_0, *i.e.* there is exactly one vertex u not in σ_0 such that $\sigma_1 = \sigma_0 \setminus \{v\} \cup \{u\}$ is in T. Note that τ_0 is rainbow and misses label 0.

If σ_1 is a rainbow d-simplex, we are done. If not, σ_1 has exactly one facet τ_1 distinct from τ_0 that is rainbow and misses label 0. Again, τ_1 is contained in exactly one d-simplex distinct from σ_1. Call this new d-simplex σ_2. Repeating this process provides a sequence $\sigma_0, \sigma_1, \ldots, \sigma_k, \ldots$ of d-simplices such that the intersection $\sigma_k \cap \sigma_{k+1}$ of any consecutive d-simplices is a rainbow $(d-1)$-simplex missing label 0.

This algorithm does not cycle and ends at some other rainbow d-simplex for the following reason. Consider the graph whose nodes are the d-simplices and whose edges connect two d-simplices if their intersection is a rainbow $(d-1)$-simplex missing label 0. A rainbow d-simplex is a degree one node of this graph. A non-rainbow d-simplex having a rainbow facet missing label 0 is of degree two. Any other d-simplex is of degree zero. The algorithm generating the sequence $\sigma_0, \sigma_1, \ldots, \sigma_k, \ldots$ starts at a degree one node of this graph and follows a path in this graph as long it is possible to follow it. It necessarily ends at another degree one node because the graph is finite. This other degree one node is a rainbow d-simplex distinct from σ_0. □

An example of a path followed by the algorithm is given in Figure 1.

3. What is pivoting?

We describe now a combinatorial framework in which it is possible to express the most common algorithms using pivot operations in a generic way.

In this framework, there are two types of pivots: primal and dual. To define them properly, we need the notions of *primoid* and *duoid*, introduced by Todd [63]. This terminology does not seem to be used anymore, but they allow a simple description of the pivots. We slightly modify his definition to be able to deal with examples that would not fit otherwise into the framework.

3.1. Primal pivot

A *primoid* is a collection P of subsets of a given ground set $V(\mathsf{P})$, called the *vertex set*, satisfying the following properties.

(i) If $\omega \in \mathsf{P}$ and $\rho \supseteq \omega$, then $\rho \in \mathsf{P}$.
(ii) All minimal subsets for inclusion in P have same cardinality, called the *rank* of P and denoted $\mathrm{rk}(\mathsf{P})$.
(iii) A subset $\rho \in \mathsf{P}$ of cardinality $\mathrm{rk}(\mathsf{P}) + 1$ contains one or two subsets of cardinality $\mathrm{rk}(\mathsf{P})$ in P.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

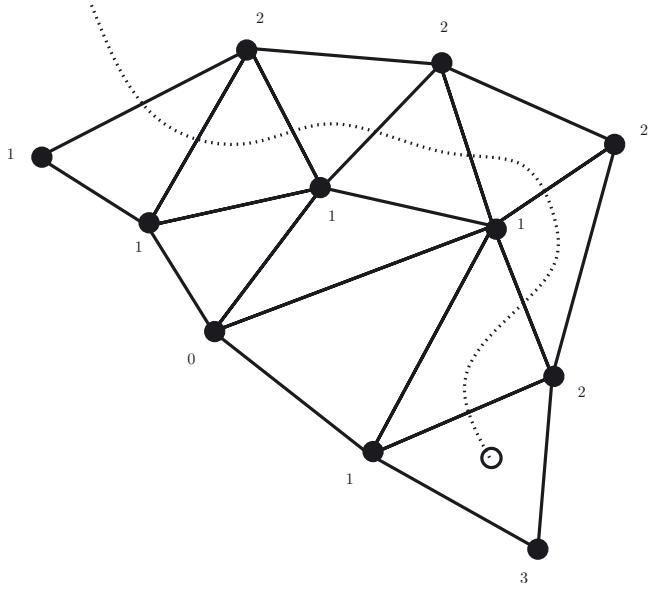


FIGURE 1. Sperner's lemma in dimension 2: a part of a path followed by the algorithm finding a rainbow simplex.

In his definition, Todd forbids the case when there is exactly one subset of cardinality $\mathrm{rk}(\mathsf{P})$ in ρ. We call such a subset $\rho \in \mathsf{P}$ of cardinality $\mathrm{rk}(\mathsf{P})+1$ containing exactly one subset of cardinality $\mathrm{rk}(\mathsf{P})$ a *boundary subset*.

Given a subset $\omega \in \mathsf{P}$ of cardinality $\mathrm{rk}(\mathsf{P})$ and an element $v \notin \omega$, the *pivot operation* consists in computing the other subset $\omega' \in \mathsf{P}$ of cardinality $\mathrm{rk}(\mathsf{P})$ contained in $\omega \cup \{v\}$, if it exists. According to the definition of a primoid, there is at most one such subset ω'. Such a pivot operation is said to be *primal*. Given $\omega \in \mathsf{P}$ with $|\omega| = \mathrm{rk}(\mathsf{P})$ and a vertex v not in ω, we can therefore define the function $\mathtt{Pivot}_{\mathsf{P}}(\omega, v)$ that computes $\omega' \in \mathsf{P} \setminus \{\omega\}$ such that $\omega' \subseteq \omega \cup \{v\}$ and $|\omega'| = |\omega|$, if it exists.

A pivoting algorithm is an algorithm repeating these pivot operations until there is no such subset ω' or until some special condition has been met (encoded as a function $f : \mathsf{P} \to \{YES, NO\}$). In addition to the pivot, the algorithm requires a way for selecting the vertex v. Given $\omega \in \mathsf{P}$ with $|\omega| = \mathrm{rk}(\mathsf{P})$, any function $\mathtt{EnteringVertex}_{\mathsf{P}}(\cdot, \omega)$ computing a vertex v not in ω fully defines an algorithm. To give more flexibility, we allow a first term ρ in the function, $\mathtt{EnteringVertex}_{\mathsf{P}}(\rho, \omega)$, with $\rho \supseteq \omega$ and $|\rho| = \mathrm{rk}(\mathsf{P}) + 1$. We can then choose the entering vertex in a more clever way.

Algorithm 1 is then the general algorithm using a primal pivot. As an input, it takes an implicit description of a primoid. It means that we are able to test

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

whether a given subset of its vertex set is an element of it or not. It can also mean that we have the function $\texttt{Pivot}_{\mathsf{P}}(\cdot,\cdot)$ at our disposal. We use a subscript f in $\texttt{EnteringVertex}_{\mathsf{P},f}$ to indicate that this function is in general designed in a way taking into account the function f.

input : An implicit description of a primoid P, an oracle $f : \mathsf{P} \to \{YES, NO\}$, and an initial subset ω^{ini} of cardinality $\text{rk}(\mathsf{P})$
output: A subset $\omega \in \mathsf{P}$ such that $f(\omega) = YES$ or contained in a boundary subset

$v \leftarrow \texttt{EnteringVertex}_{\mathsf{P},f}(\cdot, \omega^{\text{ini}})$;
$\omega^{\text{curr}} \leftarrow \omega^{\text{ini}}$;
$\rho^{\text{curr}} \leftarrow \omega^{\text{curr}} \cup \{v\}$;
while $f(\omega^{\text{curr}}) = NO$ *and* ρ^{curr} *is not boundary* **do**
  $\omega^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{P}}(\omega^{\text{curr}}, v)$;
  $v \leftarrow \texttt{EnteringVertex}_{\mathsf{P},f}(\rho^{\text{curr}}, \omega^{\text{curr}})$;
  $\rho^{\text{curr}} \leftarrow \omega^{\text{curr}} \cup \{v\}$;
end
return ω^{curr};

Algorithm 1: Pivoting algorithm: primal.

3.2. Dual pivot

A *duoid* is a collection D of subsets of a given ground set $V(\mathsf{D})$, called the *vertex set*, satisfying the following properties.

(i) If $\sigma \in \mathsf{D}$ and $\tau \subseteq \sigma$, then $\tau \in \mathsf{D}$.
(ii) All maximal subsets for inclusion in D have same cardinality, called the *rank* of D and denoted $\text{rk}(\mathsf{D})$.
(iii) A subset $\tau \in \mathsf{D}$ of cardinality $\text{rk}(\mathsf{D}) - 1$ is contained in one or two subsets of cardinality $\text{rk}(\mathsf{D})$ in D.

In his definition, Todd forbids the case when there is exactly one subset of cardinality $\text{rk}(\mathsf{D})$ containing τ. We call such a subset $\tau \in \mathsf{D}$ of cardinality $\text{rk}(\mathsf{D}) - 1$ contained in exactly one subset of cardinality $\text{rk}(\mathsf{D})$ a *boundary subset*. Note that the notions of duoid and pseudomanifold coincide. The first name is used rather of the latter when pivots are present or to emphasize the duality with a primoid. Indeed, a duoid is defined exactly as a primoid except that "$\supseteq$" is replaced by "$\subseteq$", "minimal" by "maximal", and '$\text{rk}(\mathsf{P}) + 1$' by "$\text{rk}(\mathsf{D}) - 1$". There is another way to see the duality between primal pivot and dual pivot, see Section 6.1.

Given a subset $\sigma \in \mathsf{D}$ of cardinality $\text{rk}(\mathsf{D})$ and an element $v \in \sigma$, the *pivot operation* consists in computing the other subset $\sigma' \in \mathsf{D}$ of cardinality $\text{rk}(\mathsf{D})$ containing $\sigma \setminus \{v\}$, if it exists. According to the definition of a duoid, there is at most one such subset σ'. Such a pivot operation is said to be *dual*. Given $\sigma \in \mathsf{D}$ with $|\sigma| = \text{rk}(\mathsf{D})$ and a vertex v in σ, we can therefore define the function $\texttt{Pivot}_{\mathsf{D}}(\sigma, v)$ which computes $\sigma' \in \mathsf{D} \setminus \{\sigma\}$ such that $\sigma' \supseteq \sigma \setminus \{v\}$ and $|\sigma'| = |\sigma|$ if it exists.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

A pivoting algorithm is an algorithm repeating these pivot operations until there is no such subset σ' or until some special condition has been met (encoded as a function $f : \mathsf{D} \to \{YES, NO\}$). In addition to the pivot, the algorithm requires a way for selecting the vertex v. Given $\sigma \in \mathsf{D}$ with $|\sigma| = \mathrm{rk}(\mathsf{D})$, any function $\texttt{LeavingVertex}_{\mathsf{D}}(\cdot, \sigma)$ computing a vertex v in σ fully defines an algorithm. Again, we allow a first term $\tau \subseteq \sigma$ with $|\tau| = \mathrm{rk}(\mathsf{D}) - 1$ in the function, $\texttt{LeavingVertex}_{\mathsf{D}}(\tau, \sigma)$.

Algorithm 2 is the general algorithm using a dual pivot. We use a subscript f in $\texttt{LeavingVertex}_{\mathsf{P},f}$ to indicate that this function is in general designed in a way taking into account the function f.

input : An implicit description of a duoid D, an oracle $f : \mathsf{D} \to \{YES, NO\}$, and an initial subset σ^{ini} of cardinality $\mathrm{rk}(\mathsf{D})$
output: A subset $\sigma \in \mathsf{D}$ such that $f(\sigma) = YES$ or having a boundary subset

$v \leftarrow \texttt{LeavingVertex}_{\mathsf{D},f}(\cdot, \sigma^{\mathrm{ini}})$;
$\sigma^{\mathrm{curr}} \leftarrow \sigma^{\mathrm{ini}}$;
$\tau^{\mathrm{curr}} \leftarrow \sigma^{\mathrm{curr}} \setminus \{v\}$;
while $f(\sigma^{\mathrm{curr}}) = NO$ *and* τ^{curr} *is not boundary* **do**
  $\sigma^{\mathrm{curr}} \leftarrow \texttt{Pivot}_{\mathsf{D}}(\sigma^{\mathrm{curr}}, v)$;
  $v \leftarrow \texttt{LeavingVertex}_{\mathsf{D},f}(\tau^{\mathrm{curr}}, \sigma^{\mathrm{curr}})$;
  $\tau^{\mathrm{curr}} \leftarrow \sigma^{\mathrm{curr}} \setminus \{v\}$;
end
return σ^{curr};

Algorithm 2: Pivoting algorithm: dual.

3.3. Termination of the algorithms

In the form presented above, it is not possible to prove that in general such algorithms terminate. It depends on the functions $\texttt{EnteringVertex}_{\mathsf{P},f}(\cdot, \cdot)$ and $\texttt{LeavingVertex}_{\mathsf{D},f}(\cdot, \cdot)$. The best example for this is the simplex algorithm; its termination needs *ad hoc* arguments. In Section 4, a special version of these pivoting algorithms – the complementary pivoting algorithms – is presented. A nice feature is that complementary pivoting algorithms do not have this issue.

3.4. Back to the two main examples

We show now how the main examples of Section 2 fit into the general framework of primoids and duoids and what are exactly the pivot operations for them.

3.4.1. Simplex algorithm

The underlying algorithm uses a primal pivot. The primoid P has vertex set $V(\mathsf{P}) = [n]$. A subset $J \subseteq [n]$ is in P if A_J contains $\boldsymbol{b}$ in its conic hull – we still assume genericity. We call P the primoid of the feasible bases of $(A, \boldsymbol{b})$. Because

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

of genericity, if $J \in \mathsf{P}$ and $|J| = m$, the matrix A_J is nonsingular. The function f checks whether $\boldsymbol{c}^T_{[n]\setminus J} - \boldsymbol{c}^T_J A_J^{-1} A_{[n]\setminus J} \geq 0$ (*reduced costs*). Indeed, a classical result in linear programming ensures that an optimal solution has been found when the reduced costs are nonnegative. Note that f is only applied on elements of P of cardinality m in Algorithm 1, so we do not have to define it for the other elements of P. The function `EnteringVertex` is any "pivot rule" according to the tradtional terminology of the simplex algorithm. Usually, it selects the index of a negative component of the reduced costs. We refer to the survey by Terlaky and Zhang for an overview of pivot rules [61].

3.4.2. Sperner lemma

The underlying algorithm uses a dual pivot. The duoid D is the triangulation of the sphere. The function f checks whether σ^{curr} is a rainbow d-simplex distinct from σ^{ini}. If the current d-simplex σ^{curr} is not rainbow, there is exactly one label appearing twice. The function `LeavingVertex` selects the vertex in τ^{curr} with the label appearing twice. For the first call to `LeavingVertex`, the vertex selected in σ^{ini} is the one with label 0.

In Section 4, we will see that actually a primoid and a duoid are present simultaneously. This point of view will allow a more systematic interpretation of the proof given in Section 2.2.

3.5. Other examples

In this section, we describe four other examples fitting into our framework. All of these four examples are famous and important in their field.

3.5.1. Lemke algorithm

A *linear complementarity problem* consists in solving

$$\begin{aligned} &A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ &\boldsymbol{x} \cdot \boldsymbol{y} = 0 \\ &\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m_+, \end{aligned} \tag{3.1}$$

where A is any $m \times m$ matrix. This problem generalizes the problem of finding a Nash equilibrium in bimatrix games (see also Section 4.1 for another yet close formulation of this problem). Any quadratic minimization problem can also be formulated as a linear complementarity problem (using Karush–Kuhn–Tucker conditions). Solving such problems is therefore particularly important. The book by Cottle, Pang, and Stone [9] presents several techniques for solving this problems and some applications.

A way for solving it is the use of an algorithm designed by Lemke [38] in 1965. If all components of $\boldsymbol{b}$ are nonnegative, there is the obvious solution

$$\boldsymbol{x} = \boldsymbol{0} \text{ and } \boldsymbol{y} = \boldsymbol{b}.$$

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Otherwise, we use the intermediate problem, where $\boldsymbol{e}$ is the all one vector $(1,\dots,1)$ of $\mathbb{R}^m$,

$$\begin{array}{l} A\boldsymbol{x}+\boldsymbol{y}-\lambda\boldsymbol{e}=\boldsymbol{b} \\ \boldsymbol{x}\cdot\boldsymbol{y}=0 \\ \lambda\in\mathbb{R}_+ \\ \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^m_+. \end{array} \tag{3.2}$$

The linear complementarity problem consits then in finding a solution with $\lambda=0$. Note that Equation (3.2) has always an obvious solution when at least one of the components of $\boldsymbol{b}$ is nonpositive

$$\boldsymbol{x}=\boldsymbol{0},\ \lambda=-\min_{i=1,\dots,m} b_i,\text{ and }\boldsymbol{y}=\boldsymbol{b}+\lambda\boldsymbol{e}.$$

Note also that such a solution satisfies moreover $x_i=y_i=0$ for some i.

The Lemke algorithm uses the notion of feasible basis developped for linear optimization. Let A' be the $m\times(2m+1)$ matrix $[A,I_m,\boldsymbol{e}]$. For sake of simplicity, we make the same assumption as for the simplex algorithm described in Section 2.1: we assume that $\boldsymbol{b}$ is generic with respect to A', which means that any solution of $A'\boldsymbol{z}=\boldsymbol{b}$ has at least m nonzero components. This can be obtained by a slight perturbation of $\boldsymbol{b}$.

Given a feasible basis $J\subseteq[2m+1]$ for $(A',\boldsymbol{b})$, there is at most one feasible basis in $J\cup\{k\}$ distinct from J, whatever the index $k\notin J$ is. This is the well-known result in linear optimization already mentioned in the description of the simplex algorithm. The Lemke algorithm uses indices $k=i$ or $k=m+i$ where i is the unique index such that $x_i=y_i=0$, the uniqueness of the index being a consequence of the genericity assumption. In this way, computing the other feasible basis in $J\cup\{k\}$ – the pivot operation – if it exists, leads to a new solution of problem (3.2), again with a unique index i such that $x_i=y_i=0$. We repeat these pivot operations until there in no other feasible basis in $J\cup\{k\}$ or until we reach a solution with $2m+1\notin J$, in which case we have found a solution of (3.1).

The algorithm does not cycle when we repeat the pivot operations. This fact will be proven later in the paper, see Section 4. However, there is no insurance to finish with a solution of the original problem (3.1). We may reach an infinite ray in the polyhedron $\{\boldsymbol{z}\in\mathbb{R}^{2m+1}_+ : A'\boldsymbol{z}=\boldsymbol{b}\}$, a fact that happens when there is no other feasible basis in $J\cup\{k\}$.

We see now how the algorithm described above fits into the framework of this section. The algorithm uses a primal pivot, *i.e.* Algorithm 1. The primoid P has vertex set $V(\mathsf{P})=[2m+1]$. A subset $J\subseteq[2m+1]$ is in P if A'_J contains $\boldsymbol{b}$ in its conic hull. It is the primoid of the feasible bases of $(A',\boldsymbol{b})$. The function f checks whether $\lambda=0$ for the basic solution, *i.e.* $2m+1\notin J$. The first subset ω^{ini} is $\{m+1,\dots,2m+1\}\setminus\{\bar{i}\}$ where $\bar{i}=\arg\min_{i=1,\dots,m} b_i$. The function `EnteringVertex` selects i such that $x_i=y_i=0$, the entering vertex being i or $m+i$, depending on whether i is already in ω^{curr} or not. The first call to `EnteringVertex` selects $\bar{i}$. A subset $J\cup\{k\}$ containing no other feasible basis than J – leading to an infinite ray – is a boundary subset of P.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

3.5.2. Fan lemma

As Sperner's lemma, Fan's Lemma [20] is an important result in combinatorial topology. It is actually a combinatorial generalization of the Borsuk−Ulam theorem and a generalization of Sperner's lemma. It has an algorithmic proof based on a pivot argument in a similar spirit as for Sperner's lemma. The Borsuk−Ulam theorem, which states that there is no continuous map $\mathcal{S}^d \to \mathcal{S}^{d-1}$ commuting with the central symmetry, has numerous applications in mathematics, and even in combinatorics. As for the Brouwer theorem, constructive or algorithmic proofs for this theorem are motivated by these applications. Actually, a combinatorial version of the Borsuk−Ulam theorem was found before Fan proved his lemma. It is Tucker's lemma [65] – often named "Sperner's lemma for the Borsuk−Ulam theorem" – but Fan's lemma is more general and maybe slightly easier to prove. The book by Matoušek [40] shows the link between Tucker's lemma and the Borsuk−Ulam theorem and gives many examples of applications of this latter.

Let $\boldsymbol{e}_i$ be the unit vector of $\mathbb{R}^k$ having a 1 at position i and 0 elsewhere. The *k-dimensional cross-polytope* $\Diamond^k$ is the convex hull of the $2k$ vectors $\pm\boldsymbol{e}_i$. It can alternatively be defined as the k-dimensional ball for the L_1-norm:

$$\Diamond^k = \left\{ (x_1, \ldots, x_k) \in \mathbb{R}^k : \sum_{i=1}^{k} |x_i| \leq 1 \right\}.$$

Its boundary is denoted $\partial\Diamond^k$ and is homeomorphic to $\mathcal{S}^{k-1}$. The northern (resp. southern) hemisphere of $\partial\Diamond^k$ is the set of its points having a nonnegative (resp. nonpositive) last coordinate and is denoted $\partial\Diamond^{k+}$ (resp. $\partial\Diamond^{k-}$). Note that we can alternatively define the northern hemisphere by $\{(x_1, \ldots, x_k) \in \mathbb{R}^k : \sum_{i=1}^{k} |x_i| = 1$ and $x_k \geq 0\}$ and the southern hemisphere by $\{(x_1, \ldots, x_k) \in \mathbb{R}^k : \sum_{i=1}^{k} |x_i| = 1$ and $x_k \leq 0\}$. The equator – intersection of the northern and the southern hemispheres – is the set $\{(x_1, \ldots, x_{k-1}, 0) \in \mathbb{R}^k : \sum_{i=1}^{k-1} |x_i| = 1\}$. Note that it is homeomorphic to $\mathcal{S}^{k-2}$.

Theorem 3.1 (Fan's lemma)**.** *Let T be a triangulation of $\partial\Diamond^{d+1}$ that is symmetric with respect to the origin. Let $\lambda : V(\mathsf{T}) \to \{-m, -(m-1), \ldots, -1, +1, \ldots, +(m-1), +m\}$ be a map assigning signed labels to the vertices of the triangulation for some positive integer m, such that (i) $\lambda(-v) = -\lambda(v)$ for all $v \in V(\mathsf{T})$ and (ii) $\lambda(u) + \lambda(v) \neq 0$ for all adjacent vertices u and v of T.*

Then there is an odd number of simplices σ of T such that

$$\lambda(\sigma) = \{-j_0, +j_1, \ldots, (-1)^{d+1} j_d\}$$

for some integers $1 \leq j_0 < j_1 < \ldots < j_d \leq m$.

Tucker's lemma is the special case when $m = d + 1$. Sperner's lemma is a consequence of Fan's lemma, using a simple trick, see [68].

There is a proof of Theorem 3.1 using algebraic topology given by Živaljević [68], but it requires involved tools. This gives an additional interest to the following

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

constructive and pivot-based proof. Let us call *minus-alternating* a simplex having labels of the form $\{-j_0, +j_1, \ldots, (-1)^{d+1} j_d\}$ for some integers $j_0 < j_1 < \ldots < j_d$, and *plus-alternating* a simplex having labels of the form $\{+j_0, -j_1, \ldots, (-1)^d j_d\}$. An *alternating* simplex is minus-alternating or plus-alternating. The proof is a slight adaptation of the algorithmic proof proposed by Prescott and Su [54], itself being inspired by the algorithmic proof of Tucker's lemma by Freund and Todd [22].

Algorithmic proof of Theorem 3.1. The proof works by induction on d. For $d = 0$, the result is obvious. Assume $d \geq 1$. By induction, there is a minus-alternating $(d-1)$-simplex τ_0 on the equator of the cross-polytope.

Let σ_1 be the d-simplex in the northern hemisphere having τ_0 as a facet. If σ_1 is not alternating, one can easily check that σ_1 has exactly one other minus-alternating facet τ_1. The facet τ_1 is contained in exactly one other d-simplex σ_2 – pivot operation – and so on. Repeating this process provides a sequence $\sigma_1, \sigma_2, \ldots, \sigma_k, \ldots$ of d-simplices in the northern hemisphere such that the intersection $\sigma_k \cap \sigma_{k+1}$ of any consecutive d-simplices is a minus-alternating $(d-1)$-simplex.

The algorithm does not cycle and ends either at another minus-alternating $(d-1)$-simplex on the equator, or at an alternating d-simplex. The argument is similar as for Sperner's lemma. Consider the graph whose nodes are the d-simplices of $\partial \Diamond^{(d+1)+}$ and whose edges connect two d-simplices sharing a common minus-alternating facet. The nodes of this graph are of degree zero, one, or two. The degree one nodes are the d-simplices having a minus-alternating facet on the equator and the alternating d-simplices. The degree two nodes are the non-alternating d-simplices having a minus-alternating facet. Any other d-simplex is of degree zero. The algorithm generating the sequence $\sigma_1, \sigma_2, \ldots, \sigma_k, \ldots$ starts at a degree one node of this graph and follows a path in this graph as long it is possible to follow it. It ends necessarily at another degree one node, *i.e.* at a d-simplex having a minus-alternating facet on the equator or at an alternating d-simplex.

Since the number of minus-alternating $(d-1)$-simplices on the equator is odd by induction, the number of alternating d-simplices in the northern hemisphere is odd as well. Using the symmetry with respect to the origin, we get that the total number of minus-alternating simplices on the full boundary of $\Diamond^{d+1}$ is odd. □

In Section 4, we will see that as for Sperner's lemma a primoid and a duoid are present simultaneously, allowing a more systematic interpretation of the proof above.

An example of a path followed by the algorithm is given in Figure 2.

We see now how it fits into the framework of this section. The underlying algorithm uses a dual pivot, *i.e.* Algorithm 2. The duoid D is the triangulation induced by T on $\partial \Diamond^{(d+1)+}$. The function f checks whether σ^{curr} is an alternating d-simplex. The first subset σ^{ini} is the σ_1 in the proof. If the current d-simplex σ^{curr} is not alternating, there are two consecutive labels of same sign when they are sorted by increasing absolute values. The function `LeavingVertex` selects the vertex in τ^{curr} with one of these labels.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

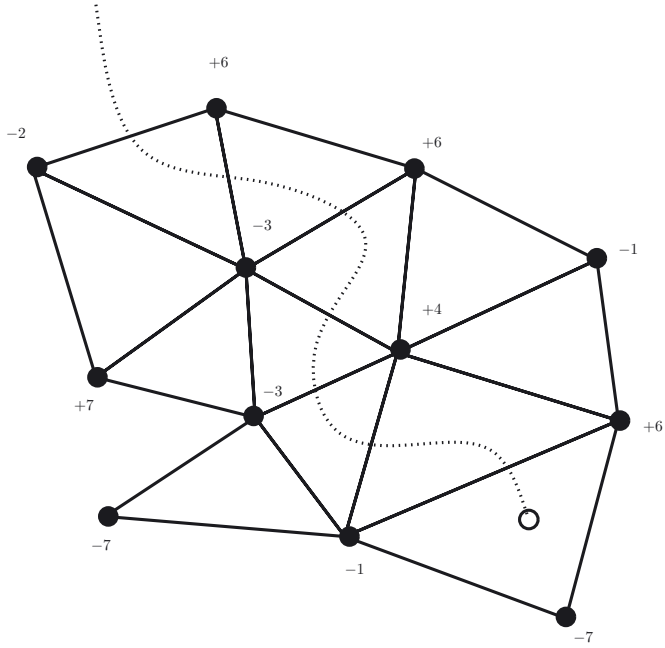


FIGURE 2. Fan's lemma in dimension 2: a part of a path followed by the algorithm finding another boundary alternating simplex or an alternating 2-simplex.

Remark 3.2. The following more general version of Theorem 3.1 in which the cross-polytope is replaced by any triangulated sphere is true, but requires additional work (see [42, 68]).

Theorem 3.3. *Let* T *be a triangulation of the* d*-dimensional sphere* $\mathcal{S}^d$ *that is symmetric with respect to the origin. Let* $\lambda : V(\mathsf{T}) \to \{-m, -(m-1), \ldots, -1, +1, \ldots, +(m-1), +m\}$ *be a map assigning signed labels to the vertices of the triangulation for some positive integer* m *such that (i)* $\lambda(-v) = -\lambda(v)$ *for all* $v \in V(\mathsf{T})$ *and (ii)* $\lambda(u) + \lambda(v) \neq 0$ *for all adjacent vertices* u *and* v *of* T.

Then there is an odd number of simplices σ *of* T *such that*

$$\lambda(\sigma) = \{-j_0, +j_1, \ldots, (-1)^{d+1} j_d\}$$

for some integers $1 \leq j_0 < j_1 < \ldots < j_d \leq m$.

3.5.3. Scarf lemma

In 1967, Scarf proved a combinatorial lemma in order to get his famous result on the core of an n-person game [57]. Actually, Scarf showed in a following paper [56] how to use his result to compute efficiently approximate fixed-point for continuous

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

maps from the ball into itself. It gives an alternative approach to what he proposed later with the help of Sperner's lemma, see Section 2.2.

In 1998, Aharoni and Holzman [3] found a surpising application of Scarf's lemma in combinatorics and provided a short and constructive proof of a conjecture by Berge and Duchet on kernels and perfect graphs [4] proved a short time before by Boros and Gurvich [5]. New applications of Scarf's lemma in combinatorics can be found in the paper of Aharoni and Fleiner [1].

Given a $m \times n$ real matrix C, a subset $J \subseteq [n]$ is said to be *C-dominating* if for any $j \in [n]$ there is a $i \in [m]$ such that $c_{ij} \leq c_{ik}$ for all $k \in J$.

Theorem 3.4 (Scarf's lemma)**.** *Let $m < n$ be positive integers and $\boldsymbol{b}$ be a vector in $\mathbb{R}^m_+$. Assume given two $m \times n$ matrices $B = (b_{ij})$ and $C = (c_{ij})$ satisfying the following three properties: (i) the first m columns of B form an $m \times m$ identity matrix, (ii) the set $\{\boldsymbol{x} \in \mathbb{R}^n_+ : B\boldsymbol{x} = \boldsymbol{b}\}$ is bounded, and (iii) $c_{ii} \leq c_{ik} \leq c_{ij}$ for any $i \in [m]$, $i \neq j \in [m]$, and $k \in [n] \setminus [m]$. Then there is a vector $\boldsymbol{x}$ in $\mathbb{R}^n_+$ such that $B\boldsymbol{x} = \boldsymbol{b}$ and such that* $\mathrm{supp}(\boldsymbol{x})$ *is C-dominating.*

The proof uses the following lemma, which we do not prove (details can be found for instance in [3]).

Lemma 3.5. *If in each row of C all entries are different, the set*

$$\mathsf{C} = \{J \subseteq [n] : J \text{ is } C\text{-dominating}\} \cup \{J \subseteq [m]\}$$

is a pseudomanifold without boundary, i.e. a duoid without boundary.

C is called the *domination complex* of C.

Proof of Theorem 3.4. First, by a perturbation argument, we can assume that any solution $\boldsymbol{x}$ of $B\boldsymbol{x} = \boldsymbol{b}$ has a support of cardinality $\geq m$ and that, in each row of C, all entries are different.

The proof uses a pivot-based algorithm on the domination complex C defined in the lemma above. We label each vertex $j \in [n]$ of C with a label $\lambda(j)$ equal to the jth column of B, which we denote B_j. Proving Theorem 3.4 consists in proving that there exists $\sigma \in \mathsf{C}$ such that $\lambda(\sigma)$ contains $\boldsymbol{b}$ in its conic hull.

A key remark is that, under the assumption above, if $\lambda(\sigma)$ contains $\boldsymbol{b}$ in its conic hull for $\sigma \subseteq [n] \setminus \{1\}$ with $|\sigma| = m$, then there is a unique m-subset $J \subseteq \sigma \cup \{1\}$ distinct of σ such that $\lambda(J)$ contains $\boldsymbol{b}$ in its conic hull. It is the same property used for the simplex algorithm. Note that J is not necessarily a simplex of C, but that $J \setminus \{1\}$ is included in σ.

The algorithm starts with the simplex $\sigma_0 = [m]$ of C. It selects then the facet τ_0 of σ_0 missing the label $B_1 = (1, 0, \ldots, 0)^T$. In other words, τ_0 is the $(m-2)$-simplex of C equal to $[m] \setminus \{1\}$. Note that $\lambda(\tau_0) \cup B_1$ contains $\boldsymbol{b}$ in its conic hull. According to Lemma 3.5, there is a unique $(m-1)$-simplex $\sigma_1 \in \mathsf{C}$ distinct from σ_0 containing τ_0. This is the pivot operation. $\lambda(\sigma_1) \cup B_1$ contains $\boldsymbol{b}$ in its conic hull. Thus, either $\lambda(\sigma_1)$ contains $\boldsymbol{b}$ in its conic hull and we are done, or there is a unique

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

facet τ_1 of σ_1 distinct from τ_0 such that $\lambda(\tau_1) \cup B_1$ contains $\boldsymbol{b}$ in its conic hull. In this latter case, τ_1 is contained in another $(m-1)$-simplex $\sigma_2 \in \mathsf{C}$. Again, either $\lambda(\sigma_2)$ contains $\boldsymbol{b}$ in its conic hull and we are done, or there is unique facet τ_2 of σ_2 distinct from τ_1 such that $\lambda(\tau_2) \cup B_1$ contains $\boldsymbol{b}$ in its conic hull. Repeating this process provides a sequence $\sigma_0, \ldots, \sigma_k, \ldots$ of $(m-1)$-simplices of C. In this sequence, the intersection $\tau_k = \sigma_k \cap \sigma_{k+1}$ of any consecutive $(m-1)$-simplices is such that $\lambda(\tau_k) \cup B_1$ contains $\boldsymbol{b}$ in its conic hull.

Again, for similar reasons as for Sperner's lemma, this algorithm cannot cycle. The graph whose nodes are the $(m-1)$-simplices of C and whose edges connect two $(m-1)$-simplices that are adjacent by the pivot operation is made of paths and cycles. The node corresponding to the simplex $[m]$ is of degree one. Therefore, the algorithm ends necessarily with a simplex of C whose labels contain $\boldsymbol{b}$ in its conic hull, which is exactly what we want to prove. □

We see now how it fits into the framework of this section. The algorithm in the proof uses a dual pivot, *i.e.* Algorithm 2. The duoid D is the domination complex C. The function f checks whether the columns of B whose indices are in σ^{curr} contain $\boldsymbol{b}$ in their conic hull. The initial subset σ^{ini} is $[m]$. The function `LeavingVertex` selects the vertex j such that $\sigma^{\text{curr}} \setminus \{j\}$ is not τ^{curr} and such that the columns whose indices are in $\sigma^{\text{curr}} \setminus \{j\} \cup \{1\}$ contain $\boldsymbol{b}$ in their conic hull. The initial call to `LeavingVertex` selects the vertex 1.

In Section 4, we will see that as for Fan's lemma and Sperner's lemma a primoid and a duoid are present simultaneously allowing a more systematic interpretation of the proof above.

3.5.4. Hamiltonian cycles in cubic graphs

Now comes the sole example which is not directly topological or geometrical in nature. The following theorem is due to Smith, see [66], and the proposed proof to Thomason [62].

Theorem 3.6. *Let G be a cubic graph (i.e. all vertices are of degree three). If there is an Hamiltonian cycle in G, then there is another one.*

Proof. Let $e = uv$ be an edge of $G = (V, E)$ contained in the given Hamiltonian cycle C. We define a *lasso* to be a subset $F \subseteq E$ such that (V, F) is a connected graph whose vertices are all of degree two except exactly two vertices: one of degree one and the other of degree three. A key remark is that a lasso contains exactly two Hamiltonian paths of G: each of them obtained by deleting an edge incident to the degree 3 vertex.

Now, consider $C \setminus e$ to which we add the edge not in C incident to v. It gives a first lasso. This lasso contains a new Hamiltonian path P distinct of $C \setminus e$. Let w be the endpoint of P distinct of u. We can add to P the edge incident to w that was not in the first lasso, in order to get a new lasso, etc. This passage from a lasso to a new one is a pivot operation. See Figure 3. The algorithm cannot cycle. Consider the (abstract) graph whose nodes are the lassos and whose edges correspond to

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

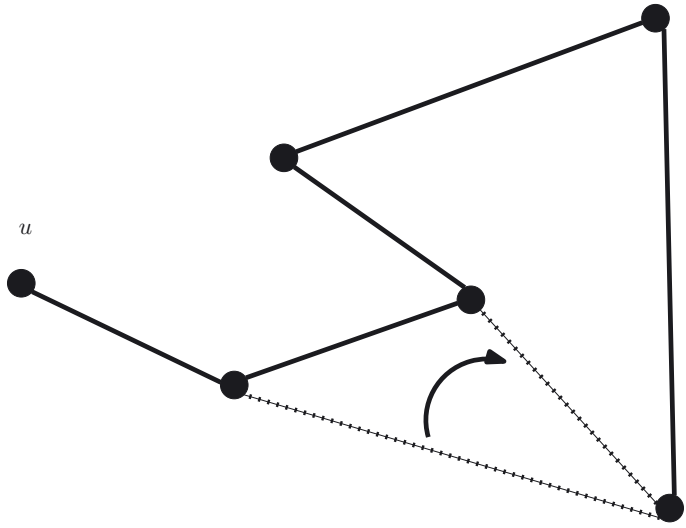


FIGURE 3. The pivot operation in the algorithm finding a second Hamiltonian path.

the pivot operation; this graph has all its nodes of degree one or two, the first lasso giving a node of degree one.

We reach necessarily an Hamiltonian path such that adding the edge according to the described procedure does not give a lasso, but gives instead an Hamiltonian cycle. □

We see now how it fits into the framework of this section. The algorithm in the proof uses a dual pivot, *i.e.* Algorithm 2. The duoid D is the set of all lassos with u of degree one, of all Hamiltonian cycles, and of all edge subsets of these sets. The function f checks whether σ^{curr} is an Hamiltonian cycle. σ^{ini} is the first lasso in the proof. The function `LeavingVertex` selects an edge incident to the degree three vertex and whose removal leads to an Hamiltonian path distinct from τ^{curr}. The initial call to `LeavingVertex` selects the arbitrarily chosen edge uv.

4. Complementary pivoting

4.1. Complementarity: problem and algorithm

For some problems, there are simultaneously a primoid and a duoid. Many of them are special cases of the *complementarity problem* defined below. As we will see in the next subsection, the linear complementarity problem or the existence of a rainbow d-simplex as in Sperner's lemma are some special cases.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Complementarity problem

Input. A primoid P and a duoid D having same vertex set and same rank; a subset σ in $\mathsf{P} \cap \mathsf{D}$

Task. Find $\sigma' \neq \sigma$ in $\mathsf{P} \cap \mathsf{D}$ if it exists.

This problem can be solved by a *complementary pivoting algorithm*, which is fully described in Algorithm 3.

input : Implicit descriptions of a primoid P and a duoid D having same vertex set V and same rank; an initial subset σ^{ini} in $\mathsf{P} \cap \mathsf{D}$
output: A subset σ in $\mathsf{P} \cap \mathsf{D}$ with $\sigma \neq \sigma^{\text{ini}}$ or a boundary subset of P or D.

$\sigma^{\text{prev}} \leftarrow \sigma^{\text{ini}}$;
pick $u_0 \in V$;
if $u_0 \in \sigma^{\text{ini}}$ **then**
  $\sigma^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{D}}(\sigma^{\text{ini}}, u_0)$;
  $\omega^{\text{curr}} \leftarrow \sigma^{\text{ini}}$;
else
  $\omega^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{P}}(\sigma^{\text{ini}}, u_0)$;
  $\sigma^{\text{curr}} \leftarrow \sigma^{\text{ini}}$;
end
while *($\sigma^{\text{curr}} \neq \omega^{\text{curr}}$), ($\sigma^{\text{curr}} \cup \omega^{\text{curr}}$) is not a boundary subset of* P*, and* ($\sigma^{\text{curr}} \cap \omega^{\text{curr}}$) *is not a boundary subset of* D **do**
  if $\sigma^{\text{prev}} = \sigma^{\text{curr}}$ **then**
    $\sigma^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{D}}(\sigma^{\text{curr}}, \sigma^{\text{curr}} \setminus \omega^{\text{curr}})$;
  else
    $\sigma^{\text{prev}} \leftarrow \sigma^{\text{curr}}$;
    $\omega^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{P}}(\omega^{\text{curr}}, \sigma^{\text{curr}} \setminus \omega^{\text{curr}})$;
  end
end
return σ^{curr};

Algorithm 3: Complementary pivoting algorithm.

Note that the algorithm works for any choice of u_0. Once P, D, and u_0 are fixed, the algorithm is fully described.

Todd proved that the algorithm always finishes. The way for proving consists in introducing what Todd calls the *fundamental graph* [63], which we denote by $G(\mathsf{P}, \mathsf{D}, u_0)$. The nodes are defined as the pairs $(\omega, \sigma) \in \mathsf{P} \times \mathsf{D}$ such that

(i) ω and σ are of cardinality $\text{rk}(\mathsf{P})$, and
(ii) $\omega \subseteq \sigma \cup \{u_0\}$.

The edges are defined as follows. There is an edge between the node (ω, σ) and the node (ω', σ') if any of the following conditions is satisfied

- $\omega = \omega' \neq \sigma$ and $\sigma' = \texttt{Pivot}_{\mathsf{D}}(\sigma, \sigma \setminus \omega)$.
- $\omega \neq \sigma = \sigma'$ and $\omega' = \texttt{Pivot}_{\mathsf{P}}(\omega, \sigma \setminus \omega)$.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

- $\omega = \omega' = \sigma$ and $\sigma' = \texttt{Pivot}_{\mathsf{D}}(\sigma, u_0)$.
- $\omega = \sigma = \sigma'$ and $\omega' = \texttt{Pivot}_{\mathsf{P}}(\omega, u_0)$.

The pivot operations correspond to the edges of the graph $G(\mathsf{P}, \mathsf{D}, u_0)$ (conditions (i) and (ii) above are still satisfied after such a pivot operation). The nodes of this graph are of degree one or two. The degree one nodes correspond to pairs (ω, σ) with $\omega = \sigma$ or to pairs (ω, σ) such that $\sigma \cup \omega$ or $\sigma \cap \omega$ is a boundary subset. The degree two nodes correspond to the other pairs (ω, σ). Starting from a node (ω, σ) with $\omega = \sigma$, the algorithm necessarily reaches another pair (ω', σ') with $\omega' = \sigma'$, or reaches a boundary subset. We have therefore the following theorem.

Theorem 4.1. *A complementary pivoting algorithm always terminates.*

The proof above gives actually another interesting result, based on the fact that the number of odd degree nodes (here of degree one) in a graph is even.

Theorem 4.2. *Let* P *and* D *be respectively a primoid and a duoid without boundary subsets. If they have same rank and share the same vertex set, then the cardinality of* $\mathsf{P} \cap \mathsf{D}$ *is even.*

Algorithm 3 can be seen as a special version of the algorithms of Section 3. If we see it as an algorithm with primal pivot, the `EnteringVertex` function is given with the function $\texttt{Pivot}_{\mathsf{D}}$. The part

$$\boxed{\sigma^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{D}}(\sigma^{\text{curr}}, \sigma^{\text{curr}} \setminus \omega^{\text{curr}})}$$

in Algorithm 3 plays the role of

$$\boxed{v \leftarrow \texttt{EnteringVertex}_{\mathsf{P},f}(\rho^{\text{curr}}, \omega^{\text{curr}})}$$

in Algorithm 1. Indeed, the vertex v used in $\texttt{Pivot}_{\mathsf{P}}$ is then $\sigma^{\text{curr}} \setminus \omega^{\text{curr}}$. The presence of the duoid D provides a natural way to select the vertex for the primal pivoting algorithm.

If we see the complementary pivoting algorithm as an algorithm with dual pivot, the `LeavingVertex` function is given with the function $\texttt{Pivot}_{\mathsf{P}}$. The part

$$\boxed{\omega^{\text{curr}} \leftarrow \texttt{Pivot}_{\mathsf{P}}(\omega^{\text{curr}}, \sigma^{\text{curr}} \setminus \omega^{\text{curr}})}$$

plays the role of

$$\boxed{v \leftarrow \texttt{LeavingVertex}_{\mathsf{D},f}(\tau^{\text{curr}}, \sigma^{\text{curr}})}$$

in Algorithm 2. Indeed, the vertex v used in $\texttt{Pivot}_{\mathsf{D}}$ is then $\sigma^{\text{curr}} \setminus \omega^{\text{curr}}$. The presence of the primoid P provides a natural way to select the vertex for the dual pivoting algorithm.

The message of this final remark is the following: in the case when we have a complementarity problem, there are canonical `LeavingVertex` and `EnteringVertex` functions. The `LeavingVertex` and `EnteringVertex` functions do not need to be defined.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

4.2. Examples

4.2.1. Lemke algorithm

The complementarity problem was originally introduced as an abstract generalization of the linear complementarity problem by authors like Cottle and Dantzig [10], Eaves [15], or Kuhn [35]. The linear complementarity problem is a special case of the complementarity problem. Indeed, equation (3.2) used when $\boldsymbol{b}$ has at least one negative component – otherwise there is an obvious solution to the linear complementarity problem – can be written

$$\begin{pmatrix} A \\ \mathbf{0}^T \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} I_m \\ \mathbf{0}^T \end{pmatrix} \boldsymbol{y} - \lambda \begin{pmatrix} \boldsymbol{e} \\ 1 \end{pmatrix} - \mu \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ \beta \end{pmatrix} \qquad \begin{aligned} \boldsymbol{x} \cdot \boldsymbol{y} &= 0 \\ \lambda, \mu &\in \mathbb{R}_+ \\ \boldsymbol{x}, \boldsymbol{y} &\in \mathbb{R}^m_+, \end{aligned} \tag{4.1}$$

where β is an arbitrary number smaller than the smallest component of $\boldsymbol{b}$. Again, we look for a solution with $\lambda = 0$. Note that equation (4.1) has always an obvious solution when at least one of the components of $\boldsymbol{b}$ is nonpositive

$$\boldsymbol{x} = \mathbf{0},\ \lambda = -\beta,\ \mu = 0, \text{ and } \boldsymbol{y} = \boldsymbol{b} - \beta \boldsymbol{e}. \tag{4.2}$$

Define A'' to be the $(m+1) \times (2m+2)$ matrix

$$A'' = \begin{pmatrix} A & I_m & -\boldsymbol{e} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0}^T & -1 & -1 \end{pmatrix}$$

and $\boldsymbol{b}'' \in \mathbb{R}^{m+1}$ by

$$\boldsymbol{b}'' = \begin{pmatrix} \boldsymbol{b} \\ \beta \end{pmatrix}.$$

Similarly as for the description of the Lemke algorithm in Section 3.5.1, we assume $\boldsymbol{b}''$ to be generic with respect to A''.

Let P be the primoid of the feasible bases of $(A'', \boldsymbol{b}'')$ and let D be defined as

$$\mathsf{D} = \{L : |L \cap \{i, i+m\}| \leq 1 \text{ for all } i \in [m] \text{ and } |L \cap \{2m+1, 2m+2\}| \leq 1\}.$$

Note that D is homeomorphic to $\partial \Diamond^{m+1}$ and that P is slightly different of that defined in Section 3.5.1 for Lemke algorithm.

The solutions of the linear complementarity problem correspond to subsets in $\mathsf{P} \cap \mathsf{D}$ containing the element $2m+2$. The initial subset σ^{ini} provided by Equation (4.2) is $\{m+1, \ldots, 2m+1\}$. The original algorithm described in Section 3.5.1 is exactly Algorithm 3 with $u_0 = 2m+2$ provided that $|\beta|$ is sufficiently large. If Algorithm 3 reaches a boundary subset in P or a subset in $\mathsf{P} \cap \mathsf{D}$ not containing the element $2m+2$, it means that the original algorithm has reached an infinite ray. With a too small $|\beta|$, it can be checked that Algorithm 3 may reach a subset in $\mathsf{P} \cap \mathsf{D}$ not containing $2m+2$, whereas the original version of the algorithm does not reach an infinite ray.

Note that we get with Theorem 4.1 a full proof of the termination of the Lemke algorithm.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

4.2.2. Sperner lemma

The problem of finding another rainbow d-simplex can be formulated as a complementarity problem with the same duoid D as in Section 3 – the triangulation of the d-sphere. Let V be the vertex set. The suitable primoid P is the collection of all subsets ω of V such that $\lambda(\omega) = \{0, \ldots, d\}$. The vertex u_0 used for the algorithm is any vertex labeled with 0.

4.2.3. Fan lemma

The problem of finding an alternating d-simplex in the framework of Fan's lemma, given a minus-alternating $(d-1)$-simplex τ on the equator, can be formulated as a complementarity problem. We explain how this formulation works. Let T^+ be the triangulation of $\partial\Diamond^{(d+1)+}$ induced by T. We define V to be the vertex set of T^+, plus a dummy vertex u, with label $\lambda(u) = (-1)^{(d+1)}(m+1)$. We define

$$\mathsf{D} = \mathsf{T}^+ \cup \{\sigma \cup \{u\} : \sigma \in \partial\mathsf{T}^+\}$$

and

$$\mathsf{P} = \{\omega \subseteq V : \omega \text{ contains an alternating set of cardinality } d+1\},$$

where an *alternating set* J is a set such that $J = \{v_0, v_1, \ldots, v_d\}$ with $|\lambda(v_i)| < |\lambda(v_{i+1})|$ and $\lambda(v_i)\lambda(v_{i+1}) < 0$ for all $i = 0, \ldots, d-1$. In this framework, σ^{ini} is the subset $\tau \cup \{u\}$. The vertex u_0 used in the complementarity algorithm is the dummy vertex u.

Note that as in the proof given in Section 3.5.2, the algorithm ends either on an alternating d-simplex of T, or on an alternating d-simplex of the form $\sigma \cup \{u\}$ with $\sigma \in \partial\mathsf{T}^+$. The algorithm proposed by Prescott and Su [54] has not this drawback: it necessarily ends with an alternating d-simplex of T. Their algorithm works with variable dimensions and knows how to explore the boundaries of the hemispheres. A complementarity formulation for Fan's lemma with a similar property would be interesting.

4.2.4. Scarf lemma

The problem of finding an $\boldsymbol{x}$ as in Theorem 3.4 can be formulated as a complementarity problem with $\mathsf{D} = \mathsf{C}$ and P the primoid of the feasible bases of $(B, \boldsymbol{b})$. The vertex u_0 is the integer 1.

4.2.5. Colorful linear programming

Let $S_1, S_2, \ldots, S_{d+1}$ be $d+1$ sets of points in $\mathbb{R}^d$, each of them of cardinality two. We assume that the points are in general position. A *colorful simplex* is a set $S \subseteq \bigcup_{i=1}^{d+1} S_i$ such that $|S \cap S_i| \leq 1$ for $i = 1, \ldots, d+1$. Meunier and Deza [45], reformulating a result of [14] (the "Octahedron Lemma"), note that the number of colorful simplices containing the origin in its convex hull is even. They describe a pivoting algorithm finding another colorful simplex containing the origin, provided

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

there is already one. This result and the algorithm are special cases of the results of the present section: the collection of all colorful simplices forms a duoid, while the collection of all subsets of $\bigcup_{i=1}^{d+1} S_i$ containing the origin in their convex hulls forms a primoid. Therefore, the problem of finding another colorful simplex containing the origin in its convex hull is a complementarity problem.

5. Complexity aspects

5.1. A suitable complexity class

Suitable complexity classes for pivoting algorithm are the PPA class and the PPAD class, defined by Papadimitriou in 1994 [53]. Recall that a functional problem is a decision problem for which a certificate has to be computed in addition to the answer "Yes".

The *PPA class* is the class of functional problems that can be polynomially encoded as a problem of the following form. We are given a polynomial-time computable function N defined on a set of objects $\mathcal{O}$ such that for each $o \in \mathcal{O}$ we have (i) $N(o) \subseteq \mathcal{O}$, (ii) $|N(o)| \in \{0, 1, 2\}$, and (iii) if $o' \in N(o)$ then $o \in N(o')$. The function N describes then implicitely a path in an abstract graph. A problem in the PPA class can be polynomially encoded as the problem: given an object o of degree one, *i.e.* such that $|N(o)| = 1$, find another object of degree one.

The *PPAD class* is the class of functional problems that can be polynomially encoded as a problem of the following form. We are given two polynomial-time computable functions N^-, N^+ defined on a set of objects $\mathcal{O}$ such that for each $o \in \mathcal{O}$ we have (i) $N^-(o), N^+(o) \subseteq \mathcal{O}$, (ii) $|N^-(o)|, |N^+(o)| \in \{0, 1\}$, and (iii) $o' \in N^+(o)$ if and only if $o \in N^-(o')$. The functions N^-, N^+ describe then implicitely a directed path in an abstract graph. A problem in the PPAD class can be polynomially encoded as the problem: given an object o such that $|N^-(o)| + |N^+(o)| = 1$, find another such object. For obvious reasons, the PPAD class is also called the END OF LINE class.

Note that we have PPAD⊆PPA. Moreover, there are complete problems for these classes, see [53] for PPAD-complete problems and [27] for the only PPA-complete problem known so far.

Several problems involving primoids or duoids fall in the PPA class. We have explained for each example how to see the underlying algorithm in terms of a graph with degree one and two nodes. The same encoding proves their belonging to the PPA class. Actually, except the problem of finding another Hamiltonian cycle, they belong all to the PPAD class. In an informal way, a PPA problem can be identified as a problem in PPAD when the following occurs. Given the initial object o^{ini} and the final object o^{fin} being the other endpoint of the path in the abstract graph, we are able to say, for any object o' on the o^{ini}-o^{fin} path, which object in $N(o')$ is in the direction of o^{fin}. For the problem solved by Scarf's algorithm, see Section 3.5.3, the proof of its PPADness is not easy and can be found in [31].

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Among the problems cited above, the linear complementarity problem and the problem solved by Scarf's algorithm are PPAD-complete. Given an implicit description of the triangulation and the labeling, the problem of finding a rainbow simplex given a first one as in Sperner's lemma, or an alternating simplex, as in Fan's lemma, are PPAD-complete [6, 52, 53] even in the case of the 2-dimensional sphere. The complexity of finding a second Hamiltonian path given a first one in a cubic graph is not known.

5.2. Length of the path

Another way for dealing with the complexity of problems solved by pivoting algorithms consists in studying the number of pivot operations the algorithm will proceed. Such a study has been particularly intense for the simplex algorithm, see, *e.g.* Klee and Minty [32], Jeroslow [30], Friedmann [23]. Up to now, all rules for selecting the entering variable in the simplex algorithm lead to exponential number of pivots for some instances. Whether there exists a pivot rule leading to a polynomial time simplex algorithm is an open question.

Morris [47] have found a infinite family of polytopes for which the number of pivots performed by the Lemke algorithm is exponential. More recently, a family of bimatrix games instances leading also to an exponential number of pivots has been found by Savani and von Stengel [55].

For the second Hamiltonian cycle problem, Krawczyk [33] exhibits a graph for which Thomason algorithm needs an exponential number of pivots.

6. Generalizations

6.1. Duality and complements

We have seen that primoids and duoids are 'dual' notions. There is another way to link these notions. We have indeed

Given a duoid D *with vertex set* V*, the set* $\{\omega \subseteq V : V \setminus \omega \in \mathsf{D}\}$ *is a primoid.*
Given a primoid P *with vertex set* V*, the set* $\{\sigma \subseteq V : V \setminus \sigma \in \mathsf{P}\}$ *is a duoid.*

It allows to write the problems considered in the paper only in terms of primoids, or only in terms of duoids. The actual trend is to use only duoids since they have a more natural geometric interpretation: as it has already be noted, they are equivalent to pseudomanifold. The complementarity problem stated in Section 4.1 can be reformulated as follows. Let D_1 and D_2 be two duoids sharing the same vertex set V such that $\mathrm{rk}(\mathsf{D}_1) + \mathrm{rk}(\mathsf{D}_2) = |V|$. Assume that we are given a pair $(\sigma_1, \sigma_2) \in \mathsf{D}_1 \times \mathsf{D}_2$ such that σ_1, σ_2 is a partition of V. Find another pair in $\mathsf{D}_1 \times \mathsf{D}_2$ forming a partition of V, if it exists. Note that we can extend this problem for more than two duoids. We describe such a generalization in Section 6.2.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

The complementarity problem stated in Section 4.1 can also be reformulated in terms of primoids. Let P_1 and P_2 be two primoids sharing the same vertex set V such that $\mathrm{rk}(\mathsf{P}_1)+\mathrm{rk}(\mathsf{P}_2) = |V|$. Assume that we are given a pair $(\omega_1, \omega_2) \in \mathsf{P}_1 \times \mathsf{P}_2$ such that ω_1, ω_2 is a partition of V. Find another pair in $\mathsf{P}_1 \times \mathsf{P}_2$ forming a partition of V, if it exists. The formulation of the complementary problem with two primoids finds its interest for instance in the context of bimatrix games, see the following remark.

Remark 6.1 (Nash equilibrium in bimatrix games and Lemke–Howson algorithm)**.** Before stating and proving the main result in this context, we review some basic notions in game theory. The reader may consult the book by Osborne and Rubinstein [51] for more complete explanations of these notions and an introduction to game theory.

Let $A = (a_{ij})$ and $B = (b_{ij})$ be two $m \times n$ matrices with real coefficients. They form the input of a *bimatrix game*: a first player selects an index i in $[m]$, while a second player selects an index j in $[n]$. The payoff of first player is a_{ij} while the payoff of the second player is b_{ij}. They both try to maximize their payoff.

A *Nash equilibrium* is a pair (i^*, j^*) such that

$$a_{i'j^*} \leq a_{i^*j^*} \text{ for all } i' \in [m] \quad \text{and} \quad b_{i^*j'} \leq b_{i^*j^*} \text{ for all } j' \in [n].$$

It models a situation in which no player is incitated to modify his choice. Nash equilibrium does not necessarily exist. However, with the weaker notion of *mixed Nash equilibirum* we explain now, existence is always ensured.

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

$$\boldsymbol{y}'^T A \boldsymbol{z}^* \leq \boldsymbol{y}^{*T} A \boldsymbol{z}^* \text{ for all } \boldsymbol{y}' \in \Delta^m \quad \text{and} \quad \boldsymbol{y}^{*T} B \boldsymbol{z}' \leq \boldsymbol{y}^{*T} B \boldsymbol{z}^* \text{ for all } \boldsymbol{z}' \in \Delta^n. \tag{6.1}$$

It models the situation when the players choose a probability distribution on their choice set and then let chance decides.

In such a situation, there is always a mixed Nash equilibrium. It is a corollary of the famous Nash theorem [49]. A proof of this special case finds a natural framework with the complementarity problem formulated in terms of primoids. Let P_A be the primoid of the bases of $([A, I_m], (1, \ldots, 1)^T)$:

$$\mathsf{P}_A = \{\omega \subseteq [n+m] : \exists \boldsymbol{x} \in \mathbb{R}^{n+m}_+ \text{ s.t. } \omega \supseteq \mathrm{supp}(\boldsymbol{x}) \text{ and } [A, I_m]\boldsymbol{x} = (1, \ldots, 1)^T\},$$

and P_B be the primoid of the bases of $([I_n, B^T], (1, \ldots, 1)^T)$:

$$\mathsf{P}_B = \{\omega \subseteq [n+m] : \exists \boldsymbol{x} \in \mathbb{R}^{n+m}_+ \text{ s.t. } \omega \supseteq \mathrm{supp}(\boldsymbol{x}) \text{ and } [I_n, B^T]\boldsymbol{x} = (1, \ldots, 1)^T\}.$$

We can assume w.l.o.g. that $(1, \ldots, 1)^T$ is generic with respect to the matrices A and B. Moreover, the game does not change if the coefficients of any matrices are translated by a same value; we can therefore assume that both A and B have nonnegative coefficients.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Note that $\omega_A = \{n+1, \ldots, n+m\} \in \mathsf{P}_A$ and $\omega_B = \{1, \ldots, n\} \in \mathsf{P}_B$ form a partition of $[n+m]$. Since both primoids are without boundary – a consequence of the nonnegativity of A and B – Theorem 4.2 shows that there exists a pair (ω_A^*, ω_B^*) such that $\omega_A^* \in \mathsf{P}_A \setminus \{\omega_A\}$ and $\omega_B^* \in \mathsf{P}_B \setminus \{\omega_B\}$, forming a partition of $[n+m]$. Therefore there exist $\boldsymbol{y}, \bar{\boldsymbol{z}} \in \mathbb{R}_+^m$ and $\bar{\boldsymbol{y}}, \boldsymbol{z} \in \mathbb{R}_+^n$ such that

$$A\boldsymbol{z} + \bar{\boldsymbol{z}} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \text{and} \quad \bar{\boldsymbol{y}}^T + \boldsymbol{y}^T B = (1, \ldots, 1)$$

and

$$\operatorname{supp}(\boldsymbol{y}) \cap \operatorname{supp}(\bar{\boldsymbol{z}}) = \emptyset \text{ and } \operatorname{supp}(\bar{\boldsymbol{y}}) \cap \operatorname{supp}(\boldsymbol{z}) = \emptyset.$$

Since $\omega_A^* \neq \omega_A$ and $\omega_B^* \neq \omega_B$, the vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ are nonzero. It is then routine to check that $\boldsymbol{y}^* = \boldsymbol{y}/||\boldsymbol{y}||_1$ and $\boldsymbol{z}^* = \boldsymbol{z}/||\boldsymbol{z}||_1$ provide a mixed Nash equilibrium, *i.e.* they satisfy Equation (6.1).

Algorithm 3 in this context was first designed by Lemke and Howson in 1964, and is now called the *Lemke–Howson algorithm* [39].

6.2. Euler complexes and semi-duoids

The parity plays a special role in the proof of the results of Section 4. A natural way for generalizing duoid without boundary consists in replacing the condition of being contained in two maximal subsets by the condition of being contained in an even number of maximal subsets. Consider a pure simplicial complex D of dimension d such that each d-subset of its vertex set is contained in an even number of d-simplices. Todd [63] uses the name *semi-duoids* for these structures. They are now more commonly called *Euler complexes* or *oiks* according to a terminology popularized by Edmonds [18].

Remark 6.2. Using terminology from combinatorial topology, one can alternatively define oiks, or semi-duoids, of rank d as cycles of some $(d-1)$-dimensional simplicial complex. Cycle is to be understood as the kernel of the usual boundary operator ∂ acting at the level of chains.

Semi-primoids are generalization of primoids. They are obtained by replacing the condition of containing zero or two subsets by the condition of containing an even number of subsets. Similarly, they can be alternatively defined as the cocycles of some $(d-1)$-dimensional simplicial complex, where d is the rank of the semi-primoids. Here again, the terminology is the one of combinatorial topology: cocycle is to be understood as the kernel of the usual coboundary operator δ acting at the level of cochains.

Edmonds and Sanità in 2008 [19] defined the following problem, which generalizes the complementarity problem by its use of semi-duoids instead of duoids and by the number of duoids. Let $\mathsf{D}_1, \ldots, \mathsf{D}_\ell$ be ℓ oiks sharing a same vertex set V. A *room partition* of $(\mathsf{D}_1, \ldots, \mathsf{D}_\ell)$ is an ℓ-tuple $(\sigma_1, \ldots, \sigma_\ell)$ such that $\sigma_i \in \mathsf{D}_i$ for $i = 1, \ldots, \ell$ and $\sigma_1, \ldots, \sigma_\ell$ form a partition of V.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

Generalized complementarity problem

Input. Oiks $\mathsf{D}_1, \dots, \mathsf{D}_\ell$ sharing the same vertex set V and such that $\sum_{i=1}^{\ell} \mathrm{rk}(\mathsf{D}_i) = |V|$; a room partition $(\sigma_1, \dots, \sigma_\ell)$ of $(\mathsf{D}_1, \dots, \mathsf{D}_\ell)$.

Task. Find a room partition distinct from $(\sigma_1, \dots, \sigma_\ell)$.

This problem has always a solution. It is a consequence of the following theorem by Edmonds and Sanità proved in the same paper. Note that it generalizes Theorem 4.2.

Theorem 6.3. *Let $\mathsf{D}_1, \dots, \mathsf{D}_\ell$ be oiks sharing the same vertex set V. If $\sum_{i=1}^{\ell} \mathrm{rk}(\mathsf{D}_i) = |V|$, then the number of room partitions is even.*

The proof is very similar to the one of Theorem 4.2. It also consists in building a graph. In general, this graph is not a path, even for $\ell = 2$. However, its odd degree vertices are the room partitions. Algorithms are still possible [63] – any algorithm for exploring a graph can be used – but they require to save the already visited edges, which makes them less natural as the one proposed for the complementarity problem.

6.3. Orientations in primoids, duoids, and oiks

Simplicial complexes can be *oriented*, see the book by Munkres [48] for instance. It is also possible to define a notion of orientation for primoids and duoids. It has be done in 1976 by Todd [64], and extended to oiks in 2012 by Végh and von Stengel [67]. Such approaches can be used for proving that some problems belong to the PPAD class. It can also be useful to deal with the indices of equilibria in game theory, see [59].

6.4. Polytopes instead of simplices

There is also a literature where the duoids, which are simplical complexes, are replaced by other kinds of cell complexes.

6.4.1. Piecewise linear manifolds

The main example of such approaches is the method developped by Eaves [16] and Eaves and Saigal [17], inspired by the pionneer works by Scarf. They lead to an active research on what is often known under the name "path-following" method, see for the example the works by Garcia [24], Garcia and Zangwill [25,26], van der Laan and Talman [36,37], Herings and van den Elzen [29].

Eaves and Saigal extend the pivoting algorithm for manifolds subdivided into polyhedral cells. Such a complex M is called a *PL manifold* (PL stands for "piecewise linear"). Denoting $\|\mathsf{M}\|$ the union of all cells in M, we assume given a map $H : \|\mathsf{M}\| \to \mathbb{R}^{d-1}$, where d is the dimension of M. This map is supposed to be affine on each cell σ of M. Assuming nondegeneracy, a cell σ is said to be *fully*

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

labeled if $\sigma \cap H^{-1}(\mathbf{0})$ is nonempty. The usual purpose in this framework is: given a fully-labeled boundary cell of M, find another fully-labeled boundary cell. The algorithm produces a polygonal path from a fully-labeled boundary cell to another fully-labeled boundary cell. This path is contained in $H^{-1}(\mathbf{0})$. The operation consisting in going from a cell to an adjacent cell along this path is the *pivot operation*.

We show now how this approach fits into the framework presented in the present survey. If M is a d-dimensional pseudomanifold, *i.e.* cells are simplices, this problem is a complementarity problem: add a dummy vertex u completing all boundary simplices of M in order to get a pseudomanifold $\overline{\mathsf{M}}$ without boundary; denote by V the set of all vertices of $\overline{\mathsf{M}}$; define a map $\overline{H} : \|\overline{\mathsf{M}}\| \to \mathbb{R}^d$, affine on each cells of the completed pseudomanifold, such that $\overline{H}(\boldsymbol{x}) = (H(\boldsymbol{x}), 1)$ for $\boldsymbol{x} \in \|M\|$ and $\overline{H}(u) = (0, \ldots, 0, -1)$; define $\mathsf{D} = \overline{\mathsf{M}}$ and $\mathsf{P} = \{\sigma \subseteq V : \mathbf{0} \in \overline{H}(\sigma)\}$. The problem of finding a fully-labeled cell is a complementarity problem. We can then use Algorithm 3. Taking u_0 as the dummy vertex u, one can see that the pivot operations coincide.

6.4.2. Cubical complexes

There are also papers focusing on cubical complexes. Kuhn [34] and Wolsey [69] study versions of Sperner's lemma where the simplices are replaced by cubes. The pivot operation consists in going from a cube to another cube sharing a common facet. One of the versions studied by Wolsey is a theorem by Fan [21] involving *cubical vertex maps*, which are counterpart of simplicial maps for cubical complexes. It is worth noting since the labeling used in Sperner's lemma, Fan's lemma, and Scarf's lemma can be interpreted as simplicial maps. Meunier [44] gives an algorithmic and pivot-based proof of another theorem by Fan, in the same paper [21], which is a cubical version of Theorem 3.1. The labeling used in this version is also a cubical vertex map. Again, the pivot operation consists in going from a cube to another cube sharing a common facet.

7. Open questions

A first open question, already outlined, is whether it is possible to find a complementarity formulation of Fan's theorem that would allow to design a complementary algorithm finding an alternating d-simplex, see Section 4.2.3.

Fan's lemma uses signed integers. The signs are $-$ and $+$. Generalizations involving more than two signs have been proposed by De Longueville and Živaljević [13] for Tucker's lemma, and Lange *et al.* [28] and Meunier [43] for the full version of Fan's lemma. The signs are seen as elements of $\mathbb{Z}_p$. In their paper, De Longueville and Živaljević raise the question of a constructive proof for their $\mathbb{Z}_p$-generalization of Tucker's lemma. A straightforward adaptation of the technics presented in the present survey would lead to a fundamental graph similar to the one of Section 4, but with some nodes of degree p, and other of degree one or two. Again, as for the generalized complementarity problems with oiks, an exploration of the graph

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

would require to store the already visited edges. An algorithm without this drawback remains to be designed for these problems.

Finally, whether the second Hamiltonian cycle problem can be formulated as a complementarity problem is another open question.

Acknowledgements. The author thanks Thomas Pradeau and Pauline Sarrabezolles for their useful remarks.

References

[1] R. Aharoni and T. Fleiner, On a lemma of Scarf. *J. Comb. Theor. Ser. B* **87** (2003) 72–80.
[2] R. Aharoni and P. Haxell, Hall's theorem for hypergraphs. *J. Graph Theory* **35** (2000) 83–88.
[3] R. Aharoni and R. Holzman, Fractional kernels in digraphs. *J. Comb. Theor. Ser. B* **73** (1998) 1–6.
[4] C. Berge and P. Duchet, *Séminaire MSH*. Paris (1983).
[5] E. Boros and V. Gurvich, Perfect graphs are kernel solvable. *Discrete Math.* **159** (1996) 35–55.
[6] X. Cheng and X. Deng, On the complexity of 2d discrete fixed point problem. *Theor. Comput. Sci.* **410** (2009) 448–4456.
[7] V. Chvátal, *Linear programming*. W.H. Freeman; 1st edn. (1983).
[8] J. Cloutier, K.L. Nyman and F.E. Su, Two-player envy-free multi-cake division. *Math. Soc. Sci.* **59** (2010) 26–37.
[9] R. Cottle, J. Pang and R. Stone, *The linear complementarity problem*. Academic Press, Boston (1992).
[10] R.W. Cottle and G.B. Dantzig, A generalization of the linear complementary problem. *J. Comb. Theor. Ser. B* **8** (1970) 79–90.
[11] G.B. Dantzig, Maximization of a linear function of variables subject to linear inequalities, in *Activity analysis of production and allocation*, edited by T.C. Koopmans. Wiley and Chapman-Hall (1947) 339–347.
[12] M. De Longueville, *A course in topological combinatorics*. Springer (2012).
[13] M. De Longueville and R. Živaljevic, The Borsuk-Ulam-property, Tucker-property and constructive proofs in combinatorics. *J. Comb. Theor. Ser. A* **113** (2006) 839–850.
[14] Antoine Deza, Sui Huang, Tamon Stephen and Tamás Terlaky, The colourful feasibility problem. *Discrete Appl. Math.* **156** (2008) 2166–2177.
[15] B.C. Eaves, *The linear complementary problem in mathematical programming*. Tech. report, Department of Operations Research, Standford University, Standford, California (1969).
[16] B.C. Eaves, Homotopies for the computation of fixed points. *Math. Programm.* **3** (1972) 1–22.
[17] B.C. Eaves and R. Saigal, Homotopies for computation of fixed points on unbounded regions. *Math. Programm.* **3** (1972) 225–237.
[18] J. Edmonds, *Euler complexes*, Research trends in combinatorial optimization. Springer (2009) 65–68.
[19] J. Edmonds and L. Sanità, On finding another room-partitioning of the vertices, *Electron. Notes in Discrete Math.* **36** (2010) 1257–1264.
[20] K. Fan, A generalization of Tucker's combinatorial lemma with topological applications. *Ann. Math.* **56** (1952) 128–140.
[21] K. Fan, Combinatorial properties of certain simplicial and cubical vertex maps. *Arch. Mathematiks* **11** (1960) 368–377.
[22] R.M. Freud and J. Todd, A constructive proof of Tucker's combinatorial lemma. *J. Comb. Theor. Ser. A* **30** (1981) 321–325.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

[23] O. Friedmann, A subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games, in *Proc. of the 15th Conference on Integer Programming and Combinatorial Optimization, IPCO'11*. New York, NY, USA (2011).
[24] C.B. Garcia, A fixed point theorem including the last theorem of Poincaré. *Math. Programm.* **8** (1975) 227–239.
[25] C.B. Garcia and W.I. Zangwill, An approach to homotopy and degree theory. *Math. Oper. Res.* **4** (1979) 390–405.
[26] C.B. Garcia and W.I. Zangwill, *Pathways to solutions, fixed points and equilibria*. Prentice-Hall, Englewood Cliffs (1981).
[27] M. Grigni, A Sperner lemma complete for PPA. *Inform. Process. Lett.* **77** (1995) 255–259.
[28] B. Hanke, R. Sanyal, C. Schultz and G. Ziegler, Combinatorial Stokes formulas via minimal resolutions. *J. Comb. Theor. Ser. A* **116** (2009) 404–420.
[29] P.J.-J. Herings and A. van den Elzen, Computation of the Nash equilibrium selected by the tracing procedure in n-person games. *Games and Economic Behavior* **38** (2002) 89–117.
[30] R. Jeroslow, The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Math.* **4** (1973) 367–377.
[31] S. Kintali, L.J. Poplawski, R. Rajaraman, R. Sundaram and S.-H. Teng, Reducibility among fractional stability problems. *IEEE Symposium Found. Comput. Sci. FOCS* (2009).
[32] V. Klee and G.J. Minty, How good is the simplex method?, Inequalities III, in *Proc. of Third Sympos. (New York), Univ. California, CA, 1969*. Academic Press (1972) 159–175.
[33] A. Krawczyk, The complexity of finding a second Hamiltonian cycle in cubic graphs. *J. Comput. System Sci.* **58** (1999) 641–647.
[34] H.W. Kuhn, Some combinatorial lemmas in topology. *IBM J.* **4** (1960) 518–524.
[35] H.W. Kuhn, Approximate search for fixed points, in *Computing methods in optimization problems 2*. Academic Press, New York (1969).
[36] G. van der Laan and A.J.J. Talman, A restart algorithm for computing fixed points without an extra dimension. *Math. Programm.* **17** (1979) 74–84.
[37] G. van der Laan and A.J.J. Talman, A restart algorithm without an artificial level for computing fixed points on unbounded regions, in *Functional differential equations and approximation of fixed points*, edited by H.O. Peitgen and M.O. Walther. Springer-Verlag, Berlin (1979) 247–256.
[38] C.E. Lemke, Bimatrix equilibrium points and equilibrium programming. *Manage. Sci.* **11** (1965) 681–689.
[39] C.E. Lemke and J.T. Howson, Equilibrium points of bimatrix games. *J. Soc. Industr. Appl. Math.* **12** (1964) 413–423.
[40] J. Matoušek, *Using the Borsuk-Ulam theorem*. Springer (2003).
[41] J. Matoušek and B. Gärtner, *Understanding and using linear programming*. Springer (2006).
[42] F. Meunier, *Configurations équilibrées*, Ph.D. thesis, Université Joseph Fourier. Grenoble (2006).
[43] F. Meunier, *A Z_q-Fan formula*. Tech. report, Laboratoire Leibniz, INPG, Grenoble (2006).
[44] F. Meunier, Discrete splittings of the necklace. *Math. Oper. Res.* **33** (2008) 678–688.
[45] Frédéric Meunier and Antoine Deza, A further generalization of the colourful Carathéodory theorem, Discrete Geometry and Optimization. *Fields Institute Communications* **69** (2013).
[46] P. Monsky, On dividing a square into triangles. *Am. Math. Monthly* **77** (1970) 161–164.
[47] D.M. Morris, Lemke paths on simple polytopes. *Math. Oper. Res.* **19** (1994) 780–789.
[48] J.M. Munkres, *Elements of algebraic topology*. Perseus Books (1995).
[49] J.F. Nash, Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. USA* **36** (1950) 48–49.
[50] J. Neyman, Un théorème d'existence. *Compt. R. Math. Acad. Sci. de Paris* **222** (1946) 843–845.
[51] M.J. Osborne and A. Rubinstein, *A course in game theory*. MIT Press (1994).
[52] D. Pálvölgyi, 2D-TUCKER is PPAD-complete. WINE, *Lect. Note Comput. Sci.* **5929** (2009) 569–574.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press

[53] C. Papadimitriou, On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.* **48** (1994) 498–532.
[54] T. Prescott and F.E. Su, A constructive proof of Ky Fan's generalization of Tucker's lemma. *J. Combi. Theor. Ser. A* **111** (2005) 257–265.
[55] R. Savani and B. von Stengel, Hard-to-solve bimatrix games. *Econometrica* **74** (2006) 397–429.
[56] H. Scarf, The approximation of fixed points of a continuous mapping. *SIAM J. Appl. Math.* **15** (1967) 1328–1343.
[57] H. Scarf, The core of an n person game. *Econometrica* **35** (1967) 50–69.
[58] H. Scarf, The computation of equilibrium prices: an exposition. in *Handbook of mathematical economics*, vol II, edited by K. Arrow and A. Kirman (1982).
[59] L.S. Shapley, A note on the Lemke-Howson algorithm. *Math. Programm. Study* **1** (1974) 175–189.
[60] E. Sperner, Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes, *Abh. Math. Sem. Univ. Hambourg* **6** (1928) 265–272.
[61] T. Terlaky and S. Zhang, Pivot rules for linear programming: A survey on recent theoretical developments. *Annal. Operat. Res.* **46** (1993) 203–233.
[62] A.G. Thomason, Hamiltonian cycles and uniquely edge colourable graphs. *Annal. Discrete Math.* **3** (1978) 259–268.
[63] M.J. Todd, A generalized complementary pivoting algorithm. *Math. Programm.* **6** (1974) 243–263.
[64] M.J. Todd, Orientations in complementary pivot algorithms. *Math. Oper. Res.* **1** (1976) 54–66.
[65] A.W. Tucker, Some topological properties of disk and sphere, in *Proc. of the First Canadian Mathematical Congress*, University of Toronto Press (1946).
[66] W.T. Tutte, On Hamiltonian circuits. *J. London Math. Soc.* **21** (1946) 98–101.
[67] L.A. Végh and B. von Stengel, Oriented Euler complexes and signed perfect matchings. *Tech. report* (2012).
[68] R. Živaljević, Oriented matroids and Ky Fan's theorem. *Combinatorica* **30** (2010) 471–484.
[69] L.A. Wolsey, Cubical Sperner lemmas as applications of generalized complementary pivoting. *J. Comb. Theor. Ser. A* **23** (1977) 78–87.

https://doi.org/10.1051/ro/2013042 Published online by Cambridge University Press