Complexity of the Gale String Problem for Equilibrium Computation in Games

Marta Maria Casetti

Thesis submitted to the Department of Mathematics
London School of Economics and Political Science
for the degree of Master of Philosophy

London, April 2015

Declaration

I certify that chapter 2 of this thesis I have presented for examination for the MPhil degree of the London School of Economics and Political Science is based on joint work with Julian Merschen and Bernhard von Stengel, published in [2]. The appendix was the result of previous study for the Master of Science degree I undertook at the London School of Economics and Political Science in 2008, see [3].

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

Abstract

This thesis presents a report on original research, published as joint work with Merschen and von Stengel in *Electronic Notes in Discrete Mathematics* [2]. Our result shows a polynomial time algorithm to solve two problems related to labeled Gale strings, a combinatorial structure consisting a string of labels and a bitstring satisfying certain conditions introduced by Gale in [7].

Gale strings can be used in the representation of a particular class of games that Savani and von Stengel [16] used as an example of exponential running time for the classical Lemke-Howson algorithm to find a Nash equilibrium of a bimatrix game [9]. It was conjectured that solving these games via the Lemke-Howson algorithm was complete in the class **PPAD** (Proof by Parity Argument, Directed version). A major motivation for the definition of this class by Papadimitriou [15] was, in turn, to capture the pivoting technique of many results related to the Nash equilibrium, including the Lemke-Howson algorithm.

Our result, on the contrary, sets apart this class of games as a case for which there is a polynomial-time algorithm to find a Nash equilibrium. Since Daskalakis, Goldberg and Papaditrimiou [5] and Chen and Deng [4] proved the **PPAD**-completeness of finding a Nash equilibrium in general normal-form games, we have a special class of games, unless **PPAD** = **P**.

Our proof exploits two results. The first one is the representation of the Nash equilibria of these games as Gale strings, as seen in Savani and von Stengel [16]. The second one is the polynomial-time solvability of the problem of finding a perfect matching in a graph, proven by Edmonds [6].

Merschen [12] and Végh and von Stengel [19] expanded our technique to prove further interesting results.

An appendix relates an amendment to the proof of the **PPAD**-completeness result by Daskalakis, Goldberg and Papaditrimiou [5].

Contents

Introduction

The topic of this thesis is a problem in the field of *algorithmic game theory*, that is, the study of game-theoretic problems from the point of view of computer science. In particular, we focus on the computational complexity of a particular class of games. These

General refs for comp compl [14]

General refs for geometry [21]

Chapter 1

Complexity, Games, Labels and Gale Strings

1.1 Some Complexity Classes

We start by recalling some standard definitions of computational complexity theory; we then move on to the more recent classes **TFNP** and **PPAD**, first introduced in [11] and [15] respectively. The latter, in particular, is a key concept in the study of the problems that are the focus of this thesis.

A *computational problem* is given by the combination of an *input* and a related *output*. A specific input gives an *instance* of the problem. Computational problems can be classified according to the form of their output. A *function problem* P returns for an intance x an output y that satisfies a given binary relation $R(x, y)$. In the case of a *decision problems*, y answers a "YES / NO" question. The *complement* of a decision problem P is the problem $\bar{P}$ that returns "NO" for each instance of P that returns "YES", and vice versa. *Search problems* are function problems that return either an output y such that $R(x, y)$, or "NO" if it's not possible to find any such y. If y is guaranteed to exist, the problem is called a *total function problem*.

An example of decision problem is: "(input) given a graph, (question) is it possible to find an Euler tour of the graph?" Its complement is "(input) given a graph, (question) is it possible that there isn't any Euler of the graph?" A search problem is: "(input) given a graph, (output) return one Euler tour of the graph, or "NO" if no such tour exists." A total function problem is: "(input) given an Euler graph, (output) return one of its Euler tours."

Computational problems are also classified according to their *computational complexity*

, given by the *reducibility* from each other.

deterministic Turing machines: here (we use only deterministic ones)

Let P_1 be a computational problem, such that its instance x is encoded by $|x|$ bits. P_1 *reduces to the problem* P_2 *in polynomial time*, denoted $P_1 \leq_P P_2$, if there exists a *polynomial-time reduction*, that is, a function $f : \{0,1\}^* \to \{0,1\}^*$ and a Turing machine $\mathcal{M}$ such that for all $x \in \{0,1\}^*$

1. $x \in P_1 \quad \Longleftrightarrow \quad f(x) \in P_2$;
2. $\mathcal{M}$ computes $f(x)$;
3. $\mathcal{M}$ stops after $p(|x|)$ steps, where p is a polynomial.

Intuitively, if P_1 is polynomial-time reducible to P_2, it takes polynomial time to "translate" P_1 to P_2, and then to "translate back" a solution of P_2 as a solution of P_1.

For any class C of decision problems, the class of all complements of the problems in C is the *complement class* co $-$ C. A problem P is *hard* for a class C if for every problem P_C in C there is a polynomial-time reduction to P; that is, if P is hard to solve at least as every problem in C. A C $-$ *hard* problem in C is *complete* for C.

The complexity class **P** contains all the *polynomially decidable problems*, that is, all problems P such that there exists a Turing machine $\mathcal{M}$ that outputs either "YES" or "NO" for all inputs $x \in \{0,1\}^*$ of P after $p(|x|)$ steps, where

p is a polynomial. Intuitively, a decision problem is in **P** if the answer to its question can be found in a number of steps that is polynomial in the input of the problem.

A problem P belongs to the class **NP**, *non-deterministic polynomial-time problems*, if there exists a Turing machine $\mathcal{M}$ and polynomials p_1, p_2 such that

1. for all $x \in P$ there exists a *certificate* $y \in \{0,1\}^*$ which satisfies $|y| \leq p_1(|x|)$;

2. $\mathcal{M}$ accepts the combined input xy, stopping after at most $p_2(|x| + |y|)$ steps;

3. for all $x \notin P$ there does not exist $y \in \{0,1\}^*$ such that $\mathcal{M}$ accepts the combined input xy.

This means that a decision problem is in **NP** if it takes polynomial time to verify whether the "certificate solution" y is, indeed, a correct answer to the question posed by the problem. The class #**P** is the class of all problems that output the number of possible certificates for a problem in **NP**.

check formal def of # P (?)

In [11], Megiddo and Papadimitriou introduce the classes **FNP**, *function non-deterministic polynomial*, and **TFNP**, *total function non-deterministic polynomial*. The former is defined as the class of binary relations $R(x, y)$ such that there is a polynomial-time algorithm that decides whether $R(x, y)$ holds for given x, y satisfying $|y| \leq p(|x|)$, where p is a polynomial. The latter is the class of all such problems for which y is guaranteed to exist. Intuitively, **FNP** and **TFNP** are similar to **NP**, but they allow for problems of (respectively) function and total function form.

In [11], Megiddo and Papadimitriou also prove that, unless $\mathbf{NP} = \mathbf{co} - \mathbf{NP}$, **TFNP** is a *semantic* class, that is, a class without complete problems. To circumvent this limitation of **TFNP**, Papadimitriou ([15]) focused on the problems for which the existence of a solution is proved by a "parity argument", introducing the classes **PPA** (*Proof by Parity Argument*) and **PPAD** (*Proof by Parity Argument, Directed version*).

The formal definition of **PPA** and **PPAD**,

definition of PPA(D): one in Papadimitriou 1994 and one in DGP the second w END OF THE LINE, use that one.

as an example, SPERNER (look at Papadimitriou 1994); will reconnect to NASH in next subsection - BROUWER mention, maybe

1.2 Normal Form Games and Nash Equilibria

We now give the game-theoretic background that will be used in this thesis. A *game*, as first defined by von Neumann in [20], is a model of strategic interaction. A *finite normal form game* is $\Gamma = (P, S = \times_{p \in P} S_p, u = \times_{p \in P} u^p)$ where both the set of *players* P and the sets of *pure strategies* S_p (and therefore the set of *pure strategy profiles* S) are finite. We will use the notation $S_{-p} = \times_{q \neq p} S_p$. The purpose of each player $p \in P$ is to maximize her *payoff function* $u^p : S \to \mathbb{R}$. In the following pages, by "game" we will always mean "finite normal form game." If there are only two players, we will refer to player 1 using feminine pronouns and to player 2 using masculine ones; such games are called *bimatrix games* since they can be characterized by the $m \times n$ payoff matrices A and B, where a_{ij} and b_{ij} are the payoffs of respectively player 1 and of player 2 when the former plays her ith pure strategy and the latter plays his jth pure strategy. A bimatrix game is *zero-sum* if $B = -A$, and *symmetric* if $B = A^\top$.

A *mixed strategy* of player p is a probability distribution on S_P; it can be described as a point $x = (x_1^p, \dots, x_{|S_p|}^p)$ on the $(|S_p| - 1)$-dimensional *mixed strategy simplex* $\Delta_p = \{x \in \mathbb{R}^{|S_p|} | x \geq \mathbf{0},\ \mathbf{1}^\top x = 1\}$; the set of *mixed strategy profiles* will be the simplicial polytope $\Delta = \times_{p \in P} \Delta_p$. We extend the payoff functions to $u^p : \Delta \to \mathbb{R}$ by linearity.

A *Nash equilibrium* of a game is a strategy profile in which each player cannot improve her expected payoff by unilaterally changing her strategy. Such a strategy is called a *best response*; a strategy that is not a best response is called *dominated*. Formally: for $s \in S_{-p}$ let $x_s = \prod_{q \neq p} x^q_{s_q}$; then a Nash equilibrium is a strategy profile x such that for every $p \in P$ and every $\sigma, \tau \in S_p$

$$\sum_{s \in S_{-p}} u^p(\sigma, s) x_s > \sum_{s \in S_{-p}} u^p(\tau, s) x_s \quad \Rightarrow \quad x^p_\tau = 0 \tag{1.1}$$

Note that applying an affine transformation to all the payoffs does not change the Nash equilibria of the game. Note also that there might be more than one equilibrium. The existence of a Nash equilibrium is guaranteed by the fundamental theorem by Nash ([13]).

Theorem 1. (Nash [13]) *Every finite game in normal form has a Nash equilibrium.*

We give three classic examples of game: matching pennies, the prisoners' dilemma and coordination.

Example 1.1. In the non-symmetric zero-sum game *matching pennies*, both players have payoff zero unless they play the same strategy. In this case, player 2 (the *evader*) pays a sum to payoff 1 (the *pursuer*).

pursuer \ evader	up	down
up	-1 1	0 0
down	0 0	-1 1

At the unique equilibrium of the game, each player follows the uniform distribution over their strategies.

In the symmetric non zero-sum *prisoners' dilemma*, each player must decide whether to "help" the other one or to "betray" them. If both players help each other, they will get a small reward; if both betray, they will pay a small penalty; if one betrays and the other cooperate the former will get a large reward and the latter will pay a large penalty.

1 \ 2	betray	help
betray	10 -2	-8 -2
help	5 -8	5 10

The only equilibrium is the profile in which both players betray. Assume that player 1 helps: then she must switch to betrayal, since she would get 10 instead of 5 if player 2 helps and -2 instead of -8 if player 2 betrays. The same applies to player 2, so both players will betray. Note that the payoff matrices can be rewritten as $A = \left(\begin{smallmatrix}1 & 3\\ 0 & 2\end{smallmatrix}\right)$ and $B = A^\top$.

As an example of game with more than one equilibrium is *coordination*: both players drive on a mountain road; they lose if drive on the same side of the road and win if they avoid each other, regardless of which side they take.

1 \ 2	mountain	valley
mountain	1 0	1 0
valley	0 1	0 1

Both (mountain,road) and (road,mountain) are Nash equilibria.

Consider the problem n-NASH, as follows.

---

n-NASH

---

input : A n-player game Γ.

output: A Nash equilibrium of Γ.

By theorem 1, n-NASH is a total function problem; Megiddo and Papadimitriou ([11]) proved that it is in **TFNP**. Daskalakis, Goldberg and Papadimitriou [5] and Chen and Deng [4] have proven its **PPAD**-completeness, the former for $n \geq 3$ and the latter for $n \geq 2$.

Theorem 2. (Daskalakis, Goldberg and Papadimitriou [5]; Chen and Deng [4]) *For $n \geq 2$, the problem n-*Nash *is* **PPAD**-*complete.*

1.3 Bimatrix Games and Labels

In the rest of this thesis we will focus bimatrix games. We will assume that the payoff matrices (A, B) are non-negative, and that neither A nor $B^\top$ has a zero column, if necessary applying an affine transformation, which does not affect the equilibria of the game.

The Nash equilibria of bimatrix games can be analysed from a combinatorial point of view using *labels*. This method is due to Shapley [18], in a study building on ideas introduced in a paper by Lemke and Howson [9]. Let $n, m \in \mathbb{N}$ with $m \leq n$, and consider a set X with $|X| = n$. A *labeling* of X is a function $l : X \to [m]$. An m-uple $x = (x_1, \ldots, x_m) \in X^m$ is *completely labeled* if $\{i \in [m] \mid l(x_j) = i \text{ for some } j \in [m]\} = [m]$, that is, if each label $j \in [m]$ appears once and only once in $(l(x_1), \ldots, l(x_m))$.

Let (A, B) be bimatrix game, and let X and Y be the mixed strategy simplices of respectively player 1 and 2; that is

$$X = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0},\ \mathbf{1}^\top x = 1\}; \quad Y = \{y \in \mathbb{R}^n \mid y \geq \mathbf{0},\ \mathbf{1}^\top y = 1\}. \quad (1.2)$$

A *labeling* of the game is then given as follows:

1. the m pure strategies of player 1 are identified by $1, \ldots, m$;
2. the n pure strategies of player 2 are identified by $m+1, \ldots, m+n$;
3. each mixed strategy $x \in X$ of player 1 has
   - label i for each $i \in [m]$ such that $x_i = 0$, that is if in x player 1 does not play her ith pure strategy,

- label $m+j$ for each $j \in [n]$ such that the jth pure strategy of player 2 is a best response to x;

4. each mixed strategy $y \in Y$ of player 2 has

   - label $m+j$ for each $j \in [n]$ such that $y_j = 0$, that is if in y player 2 does not play his jth pure strategy,
   - label i for each $i \in [m]$ such that the ith pure strategy of player 1 is a best response to y.

The labeling of mixed strategy profiles can be used to characterize the Nash equilibria of the game.

Theorem 3. (Shapley [18]) *Let $(x, y) \in X \times Y$; then (x, y) is a Nash equilibrium of the bimatrix game (A, B) if and only if (x, y) is completely labeled.*

Proof. The mixed strategy $x \in X$ has label $m + j$ for some $j \in [n]$ if and only if the jth pure strategy of player 2 is a best response to x; this, in turn, is a necessary and sufficient condition for player 2 to play his jth strategy at an equilibrium against x. Therefore, at an equilibrium (x, y) all labels $m+1, \ldots, m+n$ will appear either as labels of x or of y. The analogous holds for the labels $i \in [n]$. □

An useful graphical representation of labels on the simplices X and Y. is done by labeling the outside of each simplex according to the player's own pure strategies that are *not* played, and by subdividing its interior in closed polyhedral sets corresponding to the other player's best response, called *best response regions*. We give an example of this construction.

Example 1.2. page 3–4 of Savani, von Stengel, Unit Vector Games. With graphics.

The point of view of best response regions can be translated to an equivalent construction of *best response polytopes*. We begin by noticing that the best-response regions can be obtained as projections on X and Y of the *best-response facets* of the polyhedra

$$\bar{P} = \{(x, v) \in X \times \mathbb{R} \mid B^\top x \leq \mathbf{1}v\}; \quad \bar{Q} = \{(y, u) \in Y \times \mathbb{R} \mid Ay \leq \mathbf{1}u\}. \quad (1.3)$$

These facets in $\bar{P}$ are defined as the points $(x, v) \in X \times \mathbb{R}$ such that $(B^\top x)_j = v$, which in turn represent the strategies $x \in X$ of player 1 that give exactly payoff v to player 2 when he plays strategy j; the projection of the facet defined by $(B^\top x)_j = v$ to X has then label j. Analogously, the facet of $\bar{Q}$ given by the points $(y, u) \in Y \times \mathbb{R}$ such that $(Ay)_i = u$ will project to the best-response region of Y with label i.

Example 1.3. cont of ex above, page 4–5, image on page 5 left

Given the assumptions on non-negativity of A and $B^\top$, we can give a change coordinates to x_i/v and y_j/u and replace $\bar{P}$ and $\bar{Q}$ with the *best-response polytopes*

$$P = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}; \quad Q = \{y \in \mathbb{R}^n \mid y \geq \mathbf{0},\ Ay \leq \mathbf{1}\}. \quad (1.4)$$

The polytope P is the intersection of $n + m$ half spaces, one for each inequality corresponding to either player 1 avoiding her i-th pure strategy, where $i \in [m]$, or to a best response of player 2 that gives non-zero probability to his j-th strategy, where $j \in [n]$. Formally, a point $x \in P$ has label k if and only if either $x_k = 0$ for $k \in [m]$ or $(B^\top x)_k = 0$ for $k \in [n]$. Analogously, a point in Q has label k if and only if either $y_k = 0$ for $k \in [n]$ or $(Ay)_k$ for $k \in [m]$. Then a point $(x, y) \in P \times Q$ is completely labeled if and only if it satisfies the *complementarity condition*

$$\begin{aligned} x_i = 0 \text{ or } (Ay)_i = 1 \text{ for all } i \in [m]; \\ y_j = 0 \text{ or } (B^\top x)_j \text{ for all } j \in [n]. \end{aligned} \quad (1.5)$$

Then either he point corresponding to (x, y) in $\bar{P} \times \bar{Q}$ is a Nash equilibrium, or $(x, y) = (\mathbf{0}, \mathbf{0})$; we will refer to the latter case as *artificial equilibrium.*

Example 1.4. ex in Savani, von Stengel, image on page 5 right

nondegeneracy.

in the following: equilibria are on vertices, not simply points

P is simple, so later P^{Δ} is simplicial.

"lexicographic" perurbation makes it non-degenerate (see AB).

ex pg 9 SvS-15

From now on, we assume nondegeneracy.

We note a useful property: any bimatrix game can be "symmetrized." The result is due to Gale, Kuhn and Tucker [8] for zero-sum games; its extension to non-zero-sum games is a folklore result.

Proposition 1. *Let (A, B) be a bimatrix game and (x, y) be one of its Nash equilibria. Then (z, z), where $z = (x, y)$, is a Nash equilibrium of the symmetric game $(C, C^{\top})$, where*

$$C = \begin{pmatrix} 0 & A \\ B^{\top} & 0 \end{pmatrix}. \tag{1.6}$$

McLennan and Tourky [10] have proven a converse of proposition 1 for bimatrix games of the form (I, B), called *imitiation games*.

Proposition 2. (McLennan and Tourky [10]) *The pair (x, x) is a symmetric Nash equilibrium of the symmetric bimatrix game $(C, C^{\top})$ if and only if there is some y such that (x, y) is a Nash equilibrium of the imitation game $(I, C^{\top})$.*

Example 1.5. ex Savani, von Stengel, pg 8

A generalization of imitation games is given by *unit vector games*. These are games of the form (U, B), where the columns of the matrix U are unit vectors. The correspondence between equilibria of a unit vector game and completely labeled points of a polytope is given by the following theorem, first proved in dual form by Balthasar [1] and given here as in Savani and von Stengel [17].

also in thm: vertices, if non-deg

Theorem 4. (Savani and von Stengel [17])*[17] Let* $l : [n] \to [m]$*, and let* (U, B) *be the unit vector game where* $U = (e_{l(1)} \ \cdots \ e_{l(n)})$*. Let* $N_i = \{j \in [n] \mid l(j) = i\}$ *for* $i \in [m]$*, and consider the polytopes* P^l *and* Q^l

$$P^l = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}; \tag{1.7}$$

$$Q^l = \{y \in \mathbb{R}^n \mid y \geq \mathbf{0},\ \sum_{\substack{j \in N_i \\ i \in [m]}} y_j \leq 1\}. \tag{1.8}$$

Let l_f *be the labeling of the facets of* P^l *defined as follows:*

$$x_i \geq 0 \text{ has label } i \text{ for } i \in [m];$$

$$(B^\top x)_j \leq 1 \text{ has label } l(j) \text{ for } j \in [n]. \tag{1.9}$$

Then $x \in P^l$ *is a completely labeled point of* $P^l \setminus \{\mathbf{0}\}$ *if and only if there is some* $y \in Q^l$ *such that, after scaling, the pair* (x, y) *is a Nash equilibrium of* (U, B) vertex

Proof. Let P, Q be the polytopes associated to the game (U, B) as before. check

Let $(x, y) \in P \times Q \setminus \{\mathbf{0}, \mathbf{0}\}$ be a Nash equilibrium of (U, B), therefore completely labeled in $[m + n]$. Then, if $x_i = 0$, then x has label $i \in m$. If $x_i > 0$ instead, then y has label i, therefore $(Uy)_i = 1$, therefore for some $j \in [n]$ we have $y_j > 0$ and $U_j = e_i$, so $l(j) = i$. Since $y_j > 0$ and (x, y) is completely labeled, $x \in P$ has label $m + j$, that is, $(B^\top x)_j = 1$, therefore $x \in P^l$ has label $l(j) = i$. Hence, x is a completely labeled point of P^l.

Conversely, let $x \in P^l \setminus \{\mathbf{0}\}$ be completely labeled. If $x_i > 0$, then there is $j \in [m]$ such that $(B^\top x) = j$ and $l(j) = i$, that is, $j \in N_i$. For all i such that $x_i > 0$, define y as follows: $y_h = 0$ for all $h \in N_i \setminus \{j\}$, $y_j = 1$. Then $(x, y) \in P \times Q$ is completely labeled. $\square$

Theorem 4 gives a correspondence between the completely labeled vertices of the polytope P^l and the equilibria of the unit vector game (U, B), with an "artificial" equilibrium corresponding to the vertex $\mathbf{0}$. We will now construct its dual version given in [1]. We traslate the polytope P^l as in 1.7 to $P =$

$\{x - \mathbf{1} \mid x \in P^l\}$, and multiply all payoffs in B by a constant, if necessary, so that $\mathbf{0}$ is in the interior of P. We have that

$$P = \{x + \mathbf{1} \geq \mathbf{0},\ (x + \mathbf{1})^\top B \leq \mathbf{1}\} =$$
$$= \{x \in \mathbb{R}^m \mid -x_i \leq 1 \text{ for } i \in [m],\ x^\top (b_j/(1 - \mathbf{1}^\top b_j)) \leq 1 \text{ for } j \in [n]\}.$$

The polar of P is then

$$P^\Delta = \operatorname{conv}(\{e_i \mid i \in [m]\} \ \cup\ \{\frac{b_j}{1 - \mathbf{1}^\top b_j}\}) \tag{1.10}$$

Since P is a simple polytope with $\mathbf{0}$ in its interior, P^Δ is simplicial, $P^{\Delta\Delta} = P$, and the facets of P^Δ correspond to the vertices of P and vice versa. We can then label the vertices of P^Δ as the corresponding facets in P^l; the completely labeled facets of P^Δ will then correspond to the completely labeled vertices of P^l. In particular, the facet corresponding to $\mathbf{0}$ will be

$$F_0 = \{x \in P^\Delta \mid -\mathbf{1}^\top x = 1\} = \operatorname{conv}\{e_i \mid i \in [m]\}. \tag{1.11}$$

Theorem 4 then translates as in the original version by Balthasar [1].

Theorem 5. (Balthasar [1]) *Let Q be a labeled m-dimensional simplicial polytope with $\mathbf{0}$ in its interior and vertices $e_1, \ldots, e_m, c_1, \ldots, c_n$ such that $F_0 = \operatorname{conv}\{e_i \mid i \in [m]\}$ is a facet of Q. Let (U, B) be a unit vector game, with $U = (e_{l(1)} \ \cdots \ e_{l(n)})$ for a labeling $l : [n] \to [m]$ and $B = (b_1 \ \cdots \ b_n)$ with $b_j = c_j/(1 + \mathbf{1}^\top c_j)$ for $j \in [n]$. Let l_v be a labeling of the vertices of Q as follows:*

$$l_v(-e_i) = i \textit{ for } i \in [m];$$
$$l_v(c_j) = l(j) \textit{ for } j \in [n]. \tag{1.12}$$

Then a facet $F \neq F_0$ of Q with normal vector v is completely labeled if and only if (x, y) is a Nash equilibrium of (U, B), where $x = (v + \mathbf{1})/(\mathbf{1}^\top(v + \mathbf{1}))$, and $x_i = 0$ if and only if $e_i \in F$ for $i \in [m]$. Any j so that c_j is a vertex of F represents a pure best reply to x; the mixed strategy y is the uniform distribution on the set of the pure best replies to x.

??? here to end thm

As in theorem 4 we have a correspondence between completely labeled vertices of P^l and equilibria of the unit vector game (U, B) with an "artificial" equilibrium corresponding to the vertex $\mathbf{0}$, in theorem 5 we have a correspondence between the completely labeled facets of the polytope Q and the equilibria of the unit vector game (U, B) with the "artificial" equilibrium corresponding to the facet F_0.

prove polynomial encoding - therefore poly reduction

edit from here

Consider now the problems

Another Completely Labeled Vertex

---

input : A simple m-dimensional polytope S with $m+n$ facets; a labeling $l_f : [m+n] \to [n]$; a facet F_0 of S, completely labeled by l_f.

output: A facet $F \neq F_0$ of S, completely labeled by l.

Another Completely Labeled Facet

---

input : A simplicial m-dimensional polytope S with $m+n$ vertices; a labeling $l_v : [m+n] \to [n]$; a facet F_0 of S, completely labeled by l_v.

output: A facet $F \neq F_0$ of S completely labeled by l_v.

Unit Vector Nash

---

input : A unit vector game Γ.

output: A Nash equilibrium of Γ.

Proposition 3. *The problem* Unit Vector Nash *is polynomial-time reducible to the problems* Another Completely Labeled Vertex *and its dual* Another Completely Labeled Facet.

1.4 Cyclic Polytopes and Gale Strings

In theorems 4 and 5 we have built a correspondence between labeled polytopes and unit vector games, where Nash equilibria correspond to completely labeled vertices or facets. We now focus on a particular kind of simplicial polytopes, called *cyclic polytopes*, that can be represented as a combinatorial structure, the *Gale strings*. We will first give the definition of cyclic polytope, then of Gale string, then the theorem by Gale [7] about their correspondence.

The *moment curve* in dimension d is defined as

$$\mu_d : \mathbb{R} \longrightarrow \mathbb{R}^d, \quad \mu_d : t \longmapsto (t, t^2, \ldots, t^d)^\top. \tag{1.13}$$

The *cyclic polytope* in dimension d with n vertices, where $n > d$ is

$$C_d(n) = \mathrm{conv}\{\mu_d(t_i) \text{ for } t_1 < \ldots < t_n\}. \tag{1.14}$$

Given $k \in \mathbb{N}$ and a set S, we can represent the function $f : [k] \to S$ as the string $s = s(1)s(2)\cdots s(k)$; we have a *bitstring* if $S = \{0, \mathbf{1}\}$. A maximal substring of consecutive $\mathbf{1}$'s in a bitstring is called a *run*; an *interior* run is bounded on both sides by 0's. We will use the notation $\mathbf{1}^k$ for a run of length k, and 0^k for a string of 0's of length k A *Gale string of length* n *and dimension* d, where $n > d$, is a bitstring $s \in G(d, n)$ satisfying the following conditions:

bitstring already def in section complexity?

1. exactly d bits in s are $\mathbf{1}$ and
2. (*Gale evenness condition*)

$$0\mathbf{1}^k 0 \text{ is a substring of } s \quad \Longrightarrow \quad k \text{ is even.} \tag{1.15}$$

In general, the Gale evenness conditions allows for Gale strings that start or end with an odd-length run; but if d is even then s can start with an odd run if and only if it ends with an odd run. We can then consider the Gale strings in $G(d, n)$ with even d as a the "loops" obtained by "glueing together" the extremes of the strings, so that all runs on the loops are even. Formally:

we can see the indices of a Gale string $s \in G(d,n)$ with d even as equivalence classes modulo n, identifying $s(i+n) = s(i)$. This also shows that the set of Gale strings of even dimension is invariant under a cyclic shift of the strings.

Example 1.6. As an example of d even, we have

$$G(4,6) = \{\mathbf{1111}00, \mathbf{111}00\mathbf{1}, \mathbf{11}00\mathbf{11}, \mathbf{1}00\mathbf{111}, 00\mathbf{1111},$$
$$0\mathbf{1111}0, \mathbf{11}0\mathbf{11}0, \mathbf{1}0\mathbf{11}0\mathbf{1}, 0\mathbf{11}0\mathbf{11}\}$$

The strings $\mathbf{1111}00$, $\mathbf{111}00\mathbf{1}$, $\mathbf{11}00\mathbf{11}$, $\mathbf{1}00\mathbf{111}$, $00\mathbf{1111}$ and $0\mathbf{1111}0$ are equivalent under a cyclic shift (if considering the strings as "loops", the $\mathbf{1}$'s are all consecutive), as are the strings $\mathbf{11}0\mathbf{11}0$, $\mathbf{1}0\mathbf{11}0\mathbf{1}$ and $0\mathbf{11}0\mathbf{11}$ (if considering the strings as "loops", the even runs of $\mathbf{1}$'s are two couples separated by a single 0).

As an example for d odd, we have

$$G(3,5) = \{\mathbf{111}00, \mathbf{1}0\mathbf{11}0, \mathbf{1}00\mathbf{11}, \mathbf{11}00\mathbf{1}, 0\mathbf{11}0\mathbf{1}, 00\mathbf{111}\}$$

Note how $0\mathbf{1}0\mathbf{11}$ is a cyclic shift of $\mathbf{1}0\mathbf{11}0$, but it is not a Gale string.

The relation between cyclic polytopes and Gale strings is given by the following theorem by Gale [7].

Theorem 6. (Gale [7]) *For any positive integers d, n with $n > d$*

$$F \textit{ is a facet of } C_d(n)$$
$$\Longleftrightarrow$$
$$F = \operatorname{conv}\{\mu(t_j) \mid s(j) = 1 \textit{ for some } j \in [n] \textit{ and } s \in G(d,n)\}. \tag{1.16}$$

sketch of pf - see Ziegler - with drawing of moment curve + hyperplane

CP simplicial

CP does not depend on the choice of t_i's

Essentially, this holds because any set $S \subset [n]$ the moment curve defines a unique hyperplane which is crossed (and not just touched) by the moment curve; if the bitstring s that encodes F as $1(s)$ has a substring 01^k0

example of cyclic polytope + equivalent gale string (a simple one)

From this point forward, we will assume that d is even. We will also assume that the labeling $l : [n] \to [d]$ is such that $l(i) \neq l(i+1)$; this can be done without loss of generality, given the following consideration. Suppose that $l(i) = l(i+1)$ for some index i, and let s be a completely labeled Gale string for l. Then only one of $s(i)$ and $s(i+1)$ can be equal to $\mathbf{1}$ (note that it's possible that both are 0s). So $s(i)s(i+1)$ will never be a run of even length that "interferes" with the Gale Evenness Condition, so we can "simplify" by identifying the indices i and $i+1$.

Theorem 6 gives a correspondence between Gale strings and facets of cyclic polytopes; we have also seen that these polytopes are simplicial. On the other hand, theorem 5 gives a correspondence between completely labeled facets of a simplicial polytopes and Nash equilibria of unit vector games. To exploit these connections, we now give a definition of labeling for Gale strings that will allow us to study the Nash equilibria of a unit vector game for which the best response polytope is the dual of a cyclic polytope (recall that the polytope in theorem 4 is the best response polytope, whereas theorem 5 describes its dual version). This might seem a very specific case of bimatrix game; we will see in the next chapter that it leads to very interesting results. We therefore need a definition of "completely labeled" for Gale strings, and a labeling l_s for $G(d,n)$ such that $s \in G(d,n)$ is completely labeled if and only if the corresponding facet in $C_d(n)$ is completely labeled by l_v as given in theorem 5.

We say that $s \in G(d,n)$ is a *completely labeled Gale string* if for some labeling function $l_s : [n] \to [d]$ the set $\{i \in [n] \mid s(i) = \mathbf{1}\}$ is completely labeled by l_s. Since $s \in G(d,n)$ has exacty d bits equal to $\mathbf{1}$, this means that for each $j \in [d]$ there is exactly one $i \in [n]$ such that $s(i) = \mathbf{1}$ and $l_s(i) = j$. Note that it is not always possible to find a completely labeled Gale string.

Example 1.7. For $l = 121314$, there are no completely labeled Gale strings.

The labels $l(i) = 2, 3, 4$ appear only once in l, as $l(2), l(4), l(6)$ respectively; therefore we must have $s(2) = s(4) = s(6) = 1$. For every other $i \in [n]$ we have $l(i) = 1$, so we have $l(i) = 1$ for exactly one $i = 1, 3, 5$. The candidate strings are then **110101**, 0**11101**, 0**10111**; but none of these satisfies the Gale evenness condition.

Let (U, B), where $U = (e_{l(1),\dots,e_{l(d)}})$ for some labeling $l : [n] \to [d]$, be a unit vector game for which the dual of the best response polytope is a cyclic polytope $Q = \mathrm{conv}\{e_1, \dots, e_d, c_1, \dots, c_n\}$. Theorem 5 gives a labeling l_v of the $d + n$ vertices of Q as in 1.12:

$$l_v(-e_i) = i \text{ for } i \in [m];$$
$$l_v(c_j) = l(j) \text{ for } j \in [n].$$

Let the labeling $l_s : [d + n] \to [d]$ be defined as follows:

$$l_s(i) = i \text{ for } i \in [d];$$
$$l_s(d + j) = l(j) \text{ for } j \in [n]. \tag{1.17}$$

Then the Gale strings $s \in G(d, d + n)$ that are completely labeled for l_s correspond exactly to the completely labeled facets of Q, with the facet F_0 corresponding to the "trivial" completely labeled string $\mathbf{1}^d0$.

Example 1.8. Given the string of labels $l = 123432$, there are four associated completely labeled Gale strings: **1111**00, **11**0**11**0, **1**00**111** and **1**0**11**0**1**.

1	**2**	**3**	**4**	**3**	**2**
1	**1**	**1**	**1**	·	·
1	**1**	·	**1**	**1**	·
1	·	·	**1**	**1**	**1**
1	·	**1**	**1**	·	**1**

draw polytope

From a computational point of view, we can define the problem Another Gale and Unit Vector Cyclic Nash as follows:

Another Gale	
input :	A labeling $l : [n] \to [d]$, where d is even and $d < n$. A Gale string $s \in G(d, n)$, completely labeled by l.
output:	A Gale string $s' \in G(d, n)$, completely labeled by l, such that $s' \neq s$.

Unit Vector Cyclic Nash	
input :	A unit vector game Γ with dual cyclic best response polytope.
output:	A Nash equilibrium of Γ.

prove that encoding GS is poly

Then, by proposition 3, we have the following theorem.

Proposition 4. *The problem* Unit Vector Cyclic Nash *is polynomial-time reducible to the problem* Another Gale.

Chapter 2

Algorithmic and Complexity Results

2.1 Lemke Paths and the Lemke-Howson for Gale Algorithm

We have NEs $\Leftrightarrow$ completely labeled things (facets, vertices, GS) We give now different versions of fundam algorithm to deal with labeling looking for compl.label. - in particular in these cases
first in version on simple polytopes with labeled facets, (name: Lemke-Howson; Lemke-Howson 1964, Shapley 1974 beautiful exposition)
then dual case with labeled vertices (name???? exchange? cite from???)
then in special case Gale strings (name: Lemke-Howson for Gale, cite from???).
Mention general version - or leave it further results? (maybe better) (name: exchange algorithm, Edmonds - Sanità). (This case: index more problematic - see)

Consider a labeling $l : [n] \to [m]$, for a set X with $|X| = n$. Then $x = (x_1, \ldots, x_m) \in X^m$ is *almost completely labeled* if $|\{j \in [n] \mid x_i = j$ for some $i \in$

$[m]\}| = [m] \setminus \{k\}$ for exactly one $k \in [m]$. This mean that all labels appear once in x, except for the *missing label* k, and a *duplicate label* $\bar{k} \in [m]$ that appears twice.

now we see in poly with labeled facets, cl vertices

Let P be a simple polytope in dimension m with n facets. We define the operation of *pivoting on vertices* as moving from a vertex x of P to another vertex y such that there is an edge between x and y. Note that, since P is simple, there are exactly m possible choices for y.

Now let $l_f : [n] \to [m]$ be a labeling of the facets of P such that there is at least one completely labeled vertex x_0 of P. Note that if we pivot from vertex v we "leave behind" a facet F, that has label k; we call this *dropping label* k. We will then reach a vertex w that shares with v all facets except F (that contains v but not w) and another facet G (that contains w but not v) that has label j; we will call this *picking up label* j. We give the *Lemke-Howson algorithm* as in 1. reference!

Algorithm 1: Lemke-Howson algorithm

input : A simple m-polytope P with n facets. A labeling $l_f : [n] \to [m]$ of the facets of P. A vertex x_0 of P, completely labeled for l.

output: A completely labeled vertex $x \neq x_0$ of P.

```
choose a label k ∈ [n]
pivot from x_0 to x dropping label k
while x is not completely labeled do
    pivot from x to x' ≠ x_0 dropping the duplicate label j
    rename x_0 = x, x = x'
return x
```

The steps of the Lemke-Howson algorithm result in a *Lemke path* that connects two completely labeled vertices through k-*almost complementary* vertices and edges, that is, almost completely labeled vertices and edges where the only missing label is k. It remains to show that $y \neq x_0$. This comes from the fact that the Lemke paths are *simple paths*, that is, there are no "loops" where a vertex is visited more than once. This is not possible because at each vertex there are only two edges corresponding to the missing label k, since P is not degenerate; one is the edge that is traversed to get to the vertex, one is the one that is traversed to leave it in the next step.

Theorem 7. *The Lemke-Howson algorithm 1 returns a solution to the problem* Another Completely Labeled Vertex.

Furthermore, the number of completely labeled vertices in a simple polytope with labeled facets is even.

Proof. pf parity via Lemke paths

□

In the context of finding the Nash equilibrium of a bimatrix game (A, B), there are two equivalent implementations of the Lemke-Howson algorithm.

We can consider the game C as in proposition **??**, and the associated polytope $S = \{z \in \mathbb{R}^{m+n} \mid z \geq \mathbf{0},\ Cz \leq \mathbf{1}\}$, labeling the $2(m+n)$ inequalities defining the facets of of S as $1, \ldots, m+n, 1, \ldots, m+n$. Then applying the Lemke-Howson algorithm starting from vertex $\mathbf{0}$ returns a Nash equilibrium (z, z) of C and a corresponding $(x, y) = z$ a Nash equilibrium of (A, B).

We can also follow the "traditional" version of the Lemke-Howson algorithm; a very clear exposition of this can be found in Shapley [18]. Let P and Q be the best response polytopes of (A, B) as in 1.4. We then move alternately on P and Q, starting from the couple of vertices $(\mathbf{0}, \mathbf{0})$. Since we move in $\mathbb{R}^m$ and $\mathbb{R}^n$ instead of $\mathbb{R}^{m+n}$, this version is more practical to visualize, as shown in the following example.

Example 2.1. ex Savani - von Stengel, pag. 11; fig 8 are Schegel diagrams of BR polytopes.

Theorem 2.1 has a straightforward dual version. Let Q be a simplicial polytope in dimension m with n vertices. We *pivot on facets* by moving from facet F to facet G that shares an edge with F. Since P is simplicial, there are exactly m possible choices for G. Let $l_v : [n] \to [m]$ be a labeling of the vertices of P such that there is at least one completely labeled facet F_0. We *drop label* k and *pick up label* j when pivoting from a facet F to a facet G that shares with F all vertices except a vertex v with label k that belongs to F but not G, and another vertex w with label j that belongs to G but not F. The Lemke-Howson algorithm then becomes the theorem in 2.

reference!

Algorithm 2: Lemke-Howson algorithm on facets

input : A simplicial m-polytope Q with n vertices. A labeling $l_v : [n] \to [m]$ of the vertices of P. A vertex F_0 of Q, completely labeled for l.

output: A completely labeled facet $F \neq F_0$ of Q.

```
choose a label k ∈ [n]
pivot from F_0 to F dropping label k
while F is not completely labeled do
    pivot from F to F' ≠ F_0 dropping the duplicate label j
    rename F_0 = F, F = F'
return F
```

Considering the dual of Lemke paths on (almost) completely labeled facets, we get the dual result to theorem 2.1.

Theorem 8. *The algorithm 2 returns a solution to the problem* Another Completely Labeled Facet.

Furthermore, the number of completely labeled facets in a simplicial polytope with labeled vertices is even.

Example 2.2. example! octahedron? so we're ready for index?

To find a Nash equilibrium of a unit vector game (U, B), where $U = (e_{l(1)} \cdots e_{l(n)})$ for a labeling $l : [n] \to [m]$, we can apply theorem 4 and algorithm 1, or we can apply the dual theorem 5 and algorithm 2. The first case relies on the polytope $P^l = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}$ defined in **??**; theorem 4 shows that P^l encodes all the Nash equilibria of (U, B) as completely labeled vertices, with an "artificial" equilibrium corresponding to the vertex $\mathbf{0}$. The second case relies on the polytope Q, defined as the convex hull of vertices $-e_i$ for $i \in [m]$ and $c_j = \frac{b_j}{1 - \mathbf{1}^\top b_j}$ for $j \in [n]$; analogously, theorem 5 shows that Q encodes all the Nash equilibria of (U, B) as completely labeled facets, with the "artificial" equilibrium corresponding to the facet $F_0 = \operatorname{conv}(-e_1, \ldots, -e_m)$.

On the other hand, we can consider (U, B) as any bimatrix game, and apply algorithm 1 to the product of the best response polytopes P and Q. The projection of a Lemke path for a missing label $i \in [m]$ on $P \times Q$ to P defines a Lemke path in P^l. However, $P \times Q$ has $m + n$ labels, therefore there could be Lemke paths for a missing label $m + j$ with $j \in [n]$ on $P \times Q$ that get lost in the projection on P^l. The following theorem, proved by Savani and von Stengel in [17], shows that there is no loss of generality in studying Lemke paths on P^l; an analogous result holds for the dual case.

how?

Theorem 9. *Let (U, B) be a unit vector game, with $U = (e_{l(1)} \cdots e_{l(n)})$ for a labeling $l : [n] \to [m]$; let $P = \{x \in \mathbb{R}^m | x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}$ and $Q = \{y \in \mathbb{R}^n | y \geq \mathbf{0},\ Ay \leq \mathbf{1}\}$, as in 1.4; and let $P^l = \{x \in \mathbb{R}^m \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}$ as in* **??**. *Then the Lemke path on $P \times Q$ for the missing label k projects to a path on P that is the Lemke path on P^l for missing label k if $k \in [m]$, and for missing label $l(j)$ if $k = m + j$ with $j \in [n]$.*

As before, we now focus on the case of unit vector games where the simplicial polytope Q is cyclic; that is, the case that we can study from the point of view of Gale strings.

Consider a Gale string $s \in G(m, n)$ with d even, identifying the indices modulo n (that is, considering the string as a "loop"); let $s(k) = 1$ for an index $k \in [n]$. Then, by Gale evenness condition, there is an odd run of **1** 's in s either on the left or on the right of position k; let j be the first index after this run. A *pivoting on* s is then defined as setting $s(k) = 0$ and $s(j) = 1$. Given a labeling $l_s : [n] \to [m]$, pivoting as above becomes *dropping label* k and *picking up label* j. The *Lemke-Howson for Gale algorithm* is defined as in 3.

Algorithm 3: Lemke-Howson for Gale algorithm

input : A labeling $l_s : [n] \to [d]$ such that there is a completely labeled Gale string $s_0 \in G(d, n)$.

output: A completely labeled Gale string $s \in G(d, n)$ such that $s \neq s_0$.

```
choose a label k ∈ [d]
pivot on s_0 dropping k, obtaining s
while s is not completely labeled do
    pivot from s to s' ≠ s_0 dropping the duplicate label j
    rename s_0 = s, s = s'
return s
```

Considering the Lemke paths defined by algorithm 3 we can prove that the Lemke-Howson for Gale algorithm works and a parity result, analogously to theorems and .

Theorem 10. *The Lemke-Howson for Gale algorithm 3 returns a solution to the problem* Another Gale*.*

Furthermore, the number of completely labeled Gale strings $s \in G(d, n)$*, where* d *is even, is even.*

Example 2.3. example

In 1.17 we have given a labeling $l_s : [n + d] \to [d]$ to study the Nash equilibria (including the "artificial" equilibrium) of the unit vector game (U, B), where $U = (e_{l(1)} \cdots e_{l(n)})$ and $l : [n] \to [d]$, as Gale strings $s \in G(n + d, d)$ that are completely labeled by l_s, with the "artificial" equilibrium corresponding to the string 1^d0^n.

check!
thanks to dual of theorem SvS-15 9, when doing labeling we can take the str of labels $l(n+j)\cdots l(n+m)$ instead of $l(1)\cdots l(n+m)$, that is, we could cut the "artificial" first labels 12...n.
After all, in main we're studying ANOTHER GALE in general, not nec starting from 12...n; and we're interested in finding *one* eq that's not the one we started from (and is at other end of LPath, since index and so on), *not all equilibria*; but the eq we started from is not nec the artificial one - actually, if we go with this we can take any NE to start looking for another, and we're sure to find a "non-artificial" one. Note: if we were looking for all NE, LH doesn't work anyway - see ex by Wilson in Shapley, where "disconnected" paths between equilibria.

complexity considerations: PPAD complexity of GALE (also of ANOTHER CL VERTEX/FACET, analogous pf; for 2-NASH we have completeness by DGP+CD)
Now: it's PPAD; but is it complete? (Also, PPAD is relatively new, results welcome...)
Interesting case: Morris paths (Morris 1994), translates in terms of games by SvS 2006. Exp running time!
Could it be used to show a completeness result? Next section: we show that on the other hand it takes polynomial (!) time to find a result, so a completeness result would mean that P=PPAD - very unlikely! (One can dream, though...)
—-
Advantage of two dual cyclic polytopes as in SvS 06 over only one (and without "prefix" in labels for gale string) as seen in SvS 15 is: good example of exp on supports - see SvS 15

2.2 The Complexity of Gale and Another Gale

We will now give our main result: Another Gale can be solved in polynomial time. Therefore, it takes polynomial time to find a Nash Equilibrium of a bimatrix game for which the best response polytope is cyclic.

Our proof will be based on a simple graph construction, based on the idea of *perfect matching* and a theorem by Edmonds ([6]).

A perfect matching for a graph $G = (V, E)$ is a set $M \subseteq E$ of pairwise non-adjacent edges so that every vertex $v \in V$ is incident to exactly one edge in M.

Theorem 11 ([6]). *The problem* Perfect Matching, *defined as*

Perfect Matching	
input	***:*** *A graph* $G = (V, E)$.
question	***:*** *Is there a perfect matching for* G*?*

is solvable in polynomial time.

We will first consider the accessory problem Gale, and we will show that it is solvable in polynomial time by using theorem 11.

Gale	
input	**:** A labeling $l : [n] \to [d]$, where d is even and $d < n$.
question	**:** Does there exists a completely labeled Gale string s in $G(d, n)$ associated with l?

Theorem 12. *The problem* Gale *is solvable in polynomial time.*

Proof. We give a reduction of Gale to Perfect Matching.

In the following, we will consider every Gale string as a "loop," as seen in section **??**, so $n + 1 = 1$.

Given the labeling $l : [n] \to [d]$, let $V = [d]$, let $E = \{(l(i), l(i + 1))$ for $i \in [n]$ for every i such that $l(j) \neq l(i + 1)\}$, and consider the multigraph $G = (V, E)$.

Let $s \in G(d, n)$ be a completely labeled Gale string. Then every run of s splits uniquely into $d/2$ pairs $(i, i + 1)$ such that the labels $l(i)$ satisfy the condition $l(i) \neq l(i + 1)$, and all the labels $l(i) \in [d]$ occur. Then the labels will correspond to all the vertices of G, and the pairs will correspond to the edges of a perfect matching for G.

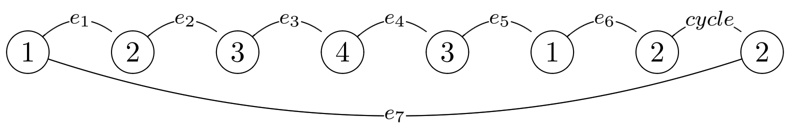
Conversely, let $l : [n] \to [d]$ be a labeling, and let M be a perfect matching for G as above. We can construct a string s such that $s(i) = s(i+1)$ for every $(l(i), l(i + 1)) \in M$ and $s(i) = 0$ otherwise. Since M is a matching, all the

$(l(i), l(i+1)) \in M$ are disjoint, so, considering s as a "loop," every run is of even length. Furthermore, since M is a perfect matching, every vertex $v \in [d]$ is the endpoint of an edge $(l(i), l(i+1))$, so s is completely labeled.

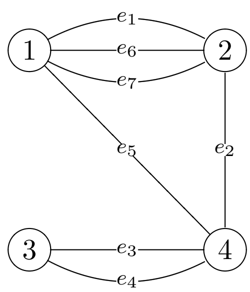
We have a reduction from Gale to the problem Perfect Matching, which is polynomial-time solvable by theorem 11. Finding a Gale string for a given labeling, or deciding that there isn't one, can therefore be done in polynomial time. $\square$

We give two examples of the construction used in theorem 12.

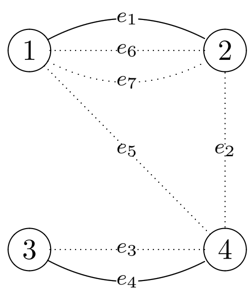
Example 2.4. Let $l = 12343122$ be a string of labels. Then the egdes e_i of the graph G obtained from the construction in the proof of theorem 12 will be as follow:



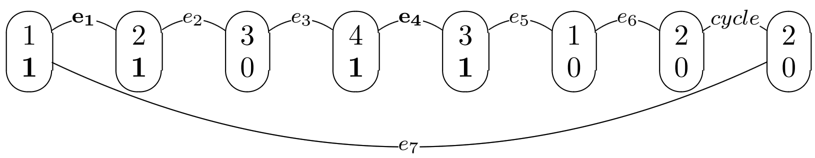
Given the vertices $v \in [4]$, the graph G will be:



A perfect matching for G is given by $M = \{e_1, e_4\}$.

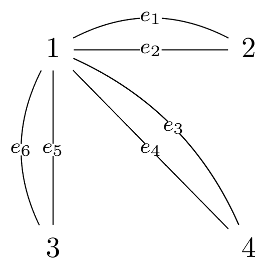


In turn, this corresponds to the completely labeled Gale string 11011000.



A perfect matching for a graph, and therefore a Gale string for a labeling, is not always possible, as shown in the next example.

Example 2.5. Let us consider the labeling $l = 121314$. The associated graph G will be



Since there aren't any disjoint edges, it's not possible to find a perfect matching for G. Analogously, we have seen in example **??** that there isn't any possible completely labeled Gale string for the labeling l.

We finally extend the proof of theorem 12 to show that Another Gale is polynomial-time solvable.

Theorem 13. *The problem* Another Gale *is solvable in polynomial time.*

Proof. Let $l : [n] \to [d]$ be a labeling, and let $s \in G(d, n)$ be a completely labeled Gale string for l. Let $G = (V, E)$ be the graph constructed from l as in the proof of theorem 12, and let M be its perfect matching for G corresponding to s.

If there is an edge $e = (l(i), l(i + 1)) \in M$ and there is an edge $e' \neq e$ in G such that $e' = (l(i), l(i + 1))$ (recall that G can be a multigraph), we simply consider the matching $M' = M \setminus \{e\} \cup \{e'\}$. Let s' be the completely labeled Gale string corresponding to M'; the 1's corresponding to the labels

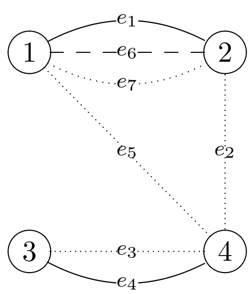
$l(i), l(i+1)$, that in s were in the position given by the edge e, for s' are in the position given by $e' \neq e$. Therefore, we have a completely labeled Gale string that is different from the one in the input of the problem.

We now assume that all the edges in every perfect matching M for G don't have a parallel edge. Note that this condition is only on the edges in the matching; G can still be a multigraph.

Theorem **??** guarantees the existence of a completely labeled Gale string $s' \neq s$; since the two strings are different, the perfect matching $M' \neq M$ corresponding to one of these s' does not use at least one edge $e \in M$. There are $d/2$ possible graphs $G'_i = (V, E'_i)$, where $E'_i = E \setminus \{e_i\}$ for each $e_i \in M$; since $V(G) = V(G')$ and $E(G) \subset E(G')$, every perfect matching for G' is a perfect matching for G as well. The existence of s' implies that there is at least one graph G' with a perfect matching $M' \neq M$. With a brute force approach, the time to find this G' and the corresponding M' will be given by the time to find a perfect matching multiplied by a factor $O(d)$. Therefore, searching for a completely labeled Gale string $s' \neq s$ takes again polynomial time. $\square$

We give two examples of the construction of theorem 13.

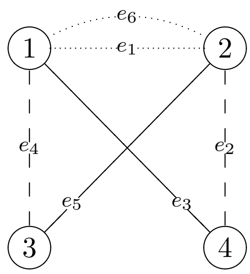
Example 2.6. We consider the labeling the string of labels $l = 1234312$. We have found in example 2.4 the completely labeled Gale string 1101100, corresponding to the perfect matching $M = \{e_1, e_4\}$ in the graph G.



If instead of e_1 we take the parallel edge e_6, the resulting matching is still perfect.

A case in which all the edges in every perfect matching don't have a parallel edge is the following; note that G is a multigraph.

Example 2.7. We consider the labeling $l = 123142$. There are only two possible perfect matchings for the corresponding graph: $M = \{e_2, e_4\}$, that corresponds to the completely labeled Gale string $s = 011110$, and $M' = \{e_3, e_5\}$, that corresponds to $s' = 001111$.



Chapter 3

Further results

A Note on the PPAD Completeness of NASH

better title

appendix/ppad-msc

Polytopes

better title

We denote the transpose of a matrix A as $A^\top$. We consider vectors $u, v \in \mathbb{R}^d$ as column vectors, so $u^\top v$ is their scalar product. A vector in $\mathbb{R}^d$ for which all components are 0's will be denoted as $\mathbf{0}$; similarly, a vectors for which all components are 1's will be denoted as $\mathbf{1}$. The *unit vector* e_i is the vector that has i-th component $e_{ii} = 1$ and $e_{ij} = 0$ for all other components. When writing an inequality of the form $u \geq v$ (and analogous), we mean that it holds for every component; that is, $u_i \geq v_i$ for all $i \in [d]$.

An *affine combination* of points in an Euclidean space $z_1, \ldots, z_n$ is

$$\sum_{i=1}^{n} \lambda_i z_i \quad \text{where } \lambda_i \in \mathbb{R} \text{ such that } \sum_{i=1}^{n} \lambda_i = 1$$

The points $z_1, \ldots, z_n$ are *affinely independent* if none of them is an affine combination of the others.

A *convex combination* of points $z_1, \ldots, z_n$ is an affine combination where $\lambda_i \geq 0$ for all $i \in [n]$. Note that such λ_i's can be seen as a probability distribution over the z_i's.

A set of point Z is *convex* if it is closed under forming convex combinations, that is, if $\bar{z} = \sum_{i=1}^{n} \lambda_i z_i$, where $z_i \in Z$, $\lambda_i \geq 0$ and $\sum_{i=1}^{n} \lambda_i = 1$, then $\bar{z} \in Z$. A convex set has *dimension* d if it has exactly $d + 1$ affinely independent points.

convex hull (needed for def cyclic poly);

pow hyperplanes;

polyhedra, polytopes

simplex

simple and simplicial polytopes

polar: $Q = \{x \in \mathbb{R}^d \mid x^\top c_i \leq 1,\ i \in [k]\}$

with $c_i \in \mathbb{R}^d$. Then the polar (Ziegler, 1995) of Q is given by

$Q^\Delta = \mathrm{conv}\{c_i, i \in [k]\}$

from here: notes - copy-paste

A (d-dimensional) *simplicial polytope* P is the convex hull of a set of at least $d + 1$ points v in $\mathbb{R}^d$ in general position, that is, no $d + 1$ of them are on a common hyperplane.

If a point v cannot be omitted from these points without changing P then v is called a *vertex* of P. A *facet* of P is the convex hull $\mathrm{conv}\, F$ of a set F of d vertices of P that lie on a hyperplane $\{x \in \mathbb{R}^d \mid a^T x = a_0\}$ so that $a^T u < a_0$ for all other vertices u of P; the vector a (unique up to a scalar multiple) is called the *normal vector* of the facet. We often identify the facet with its set of vertices F.

Acknowledgements

appendix/acknowledgments

Index of Symbols

appendix/symbols

Bibliography

[1] A. V. Balthasar (2009). “Geometry and equilibria in bimatrix games.” PhD Thesis, London School of Economics and Political Science.

[2] M. M. Casetti, J. Merschen, B. von Stengel (2010). “Finding Gale Strings.” *Electronic Notes in Discrete Mathematics* 36, pp. 1065–1082.

[3] M. M. Casetti (2008). “PPAD Completeness of Equilibrium Computation.” MSc Thesis, London School of Economics and Political Science.

[4] X. Chen, X. Deng (2006). “Settling the Complexity of 2-Player Nash Equilibrium.” *Proc. 47th FOCS*, pp. 261–272.

[5] C. Daskalakis, P. W. Goldberg, C. H. Papadimitriou (2006). “The Complexity of Computing a Nash Equilibrium.” *SIAM Journal on Computing*, 39(1), pp. 195–259.

[6] J. Edmonds (1965). “Paths, Trees, and Flowers.” *Canad. J. Math.* 17, pp. 449–467.

[7] D. Gale (1963). “Neighborly and Cyclic Polytopes.” *Convexity, Proc. Symposia in Pure Math.*, Vol. 7, ed. V. Klee, American Math. Soc., Providence, Rhode Island, pp. 225–232.

[8] D. Gale, H. W. Kuhn, A. W. Tucker (1950). “On Symmetric Games.” *Contributions to the Theory of Games* I, eds. H. W. Kuhn and A. W. Tucker, *Annals of Mathematics Studies* 24, Princeton University Press, Princeton, pp. 81–87.

[9] C. E. Lemke, J. T. Howson, Jr. (1964). “Equilibrium Points of Bimatrix Games.” *J. Soc. Indust. Appl. Mathematics* 12, pp. 413–423.

[10] A. McLennan, R. Tourky (2010). “Imitation Games and Computation.” *Games and Economic Behavior* 70, pp. 4–11.

[11] N. Megiddo, C. H. Papadimitriou (1991). "On Total Functions, Existence Theorems and Computational Complexity." *Theoretical Computer Science* 81, pp. 317–324.

[12] J. Merschen (2012). "Nash Equilibria, Gale Strings, and Perfect Matchings." PhD Thesis, London School of Economics and Political Science.

[13] J. F. Nash (1951). "Noncooperative games." *Annals of Mathematics*, 54, pp. 289-295.

[14] C. H. Papadimitriou (1994). *Computational Complexity.* Addison-Wesley, Reading, MA.

[15] C. H. Papadimitriou (1994). "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence." *J. Comput. System Sci.* 48, pp. 498–532.

[16] R. Savani, B. von Stengel (2006). "Hard-to-solve Bimatrix Games." *Econometrica* 74, pp. 397–429.

[17] R. Savani, B. von Stengel (2015). "Unit Vector Games." arXiv:1501.02243v1 [cs.GT]

[18] L. S. Shapley (1974). "A Note on the Lemke-Howson Algorithm." *Mathematical Programming Study 1: Pivoting and Extensions*, pp. 175–189

[19] L. Végh, B. von Stengel "Oriented Euler Complexes and Signed Perfect Matchings." arXiv:1210.4694v2 [cs.DM]

[20] J. von Neumann, (1928). "Zur Theorie der Gesellschaftspiele." *Mathematische Annalen*, 100, pp. 295–320.

[21] G. M. Ziegler (1995) *Lectures on Polytopes.* Springer, New York.