

## 1 Introduction

## 2 Complexity, Games, Polytopes and Gale Strings

### 2.1 Some Complexity Classes

### 2.2 Normal Form Games and Nash Equilibria

file: background-subsection

### 2.3 Some Geometrical Notation

### 2.4 Bimatrix Games, Labels and Polytopes

file: polytopes-subsection

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

a point in  $P$  has label  $k$  iff  $x_k = 0$  for  $k \in \{1, \dots, m\}$  or  $(B^\top x)_{k-m} = 0$  for  $k \in \{m+1, \dots, m+n\}$ ; analogously, a point in  $Q$  has label  $k$  if and only if either  $y_{k-m} = 0$  for  $k \in \{m+1, \dots, m+n\}$  or  $(Ay)_k = 0$  for  $k \in \{1, \dots, m\}$ .

**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*

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

**Theorem 1.** [15] *Let  $l : [n] \rightarrow [m]$ , and let  $(U, B)$  be the unit vector game where  $U = (e_{l(1)} \cdots e_{l(n)})$ . Consider the polytopes  $P^l$  and  $Q^l$  where*

$$P^l = \{x \in \mathbb{R}^m | x \geq \mathbf{0}, B^\top x \leq \mathbf{1}\} \quad (2)$$

$$Q^l = \{y \in \mathbb{R}^n | y \geq \mathbf{0}, \sum_{\substack{j \in N_i \\ i \in [m]}} y_j \leq 1\} \quad (3)$$

where  $N_i = \{j \in [n] | l(j) = i\}$  for  $i \in [m]$ .

Label every facet of  $P^l$  according to the inequality defining it, as follows:

- $x_i \geq 0$  has label  $i$ , for  $i \in [m]$
- $(B^\top x)_j \leq 1$  has label  $l(j)$ , for  $j \in [n]$

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  $(U, B)$

## 2.5 Cyclic Polytopes and Gale Strings

## 2.6 Labeling and the Problem ANOTHER GALE

file: gale-def-subsection

# 3 Algorithmic and Complexity Results

### 3.1 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 in its corresponding version on Gale strings (name: Lemke-Howson for Gale, where???).

We will also mention dual version on simplicial polytopes with labeled vertices (name: exchange algorithm, Edmonds - Sanità). (This case: index more problematic - see )

Sth about why (exp lemke paths; conjecture)

Consider a labeling  $l : [n] \rightarrow [m]$ , for a set  $X$  with  $|X| = n$ . Then  $x = (x_1, \dots, x_m) \in X^m$  is *almost completely labeled* if  $|\{j \in [n] \mid x_i = j \text{ for some } i \in [m]\}| = [m] \setminus \{k\}$  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  $l : [n] \rightarrow [m]$  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 a vertex we “leave behind” a facet, that has label  $k$ ; we call this *dropping label*  $k$ . We will then reach a vertex that lies on a new facet, that has label  $j$ ; we will call this *picking up label*  $j$ . We give the *Lemke-Howson algorithm* as follows:

reference!

The steps of the Lemke-Howson algorithm result in a *Lemke path* that connects two completely labeled vertices through *k-almost complementary*

---

**Algorithm 1:** Lemke-Howson algorithm

---

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

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

```
1 choose a label  $k \in [n]$ 
2 pivot from  $x$  to  $y$  dropping label  $k$ 
3 while  $y$  is not completely labeled do
4   | let  $j$  be the duplicate label of  $y$ 
5   | pivot from  $y$  to  $y'$  dropping label  $j$  on the facet shared with  $x$ 
6   | rename  $y = y'$ 
7 return  $y$ 
```

---

vertices and edges, that is, almost completely labeled vertices and edges where the only missing label is  $k$ . It remains to show that  $y \neq x$ . 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. This proves the following.

**proposition 2.** *The Lemke-Howson algorithm returns a solution to the problem ANOTHER COMPLETELY LABELED VERTEX.*

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 1, 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  $S$  as  $1, \dots, m+n, 1, \dots, m+n$ . 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 [16]. Let  $P$  and  $Q$  be the best response polytopes of  $(A, B)$  as in 1. 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.

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

*Example 3.1.*

(?) thm Savani '06 - pg 12 SvS-15: not only LH paths on  $P \times Q$  simple path, but projection to simple paths on both  $P$  and  $Q$ .

In the case of a unit vector game  $(U, B)$ , where  $U = (e_{l(1)} \cdots e_{l(n)})$  for a labeling  $l : [n] \rightarrow [m]$ , let as in 2; theorem 1 then shows that  $P^l$  encodes all the Nash equilibria of  $(U, B)$  as completely labeled vertices. We can apply the Lemke-Howson algorithm to both  $P^l$  and 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 in Savani and von Stengel [15], shows that there is no loss of generality in studying Lemke paths on  $P^l$ .

how?

**Theorem 2.** *Let  $(U, B)$  be a unit vector game, with  $U = (e_{l(1)} \cdots e_{l(n)})$  for a labeling  $l : [n] \rightarrow [m]$ ; 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; and let  $P^l = \{x \in \mathbb{R}^m | x \geq \mathbf{0}, B^\top x \leq \mathbf{1}\}$  as in 2. 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  $j \in [n]$ .*

if poly dual cyclic, when cons from pow gale strings we could 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  $12 \dots n$ .

Do we go straight for this, or? YES!

After all, in main we’re studying ANOTHER GALE in general, not nec starting from  $12 \dots n$ ; 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

Needed for why result interesting; it can be done at the end, with Morris paths

—

extend term “Lemke paths” to paths of every LH-style algorithms we see. ... In this case, Lemke paths “translate” as / correspond to...

### 3.2 The Complexity of GALE and ANOTHER GALE

file: main-result-subsection

## References

- [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] X. Chen, X. Deng (2006). “Settling the Complexity of 2-Player Nash Equilibrium.” *Proc. 47th FOCS*, pp. 261–272.
- [4] 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.
- [5] J. Edmonds (1965). “Paths, Trees, and Flowers.” *Canad. J. Math.* 17, pp. 449–467.
- [6] 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.
- [7] 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.
- [8] C. E. Lemke, J. T. Howson, Jr. (1964). “Equilibrium Points of Bimatrix Games.” *J. Soc. Indust. Appl. Mathematics* 12, pp. 413–423.
- [9] A. McLennan, R. Tourky (2010). “Imitation Games and Computation.” *Games and Economic Behavior* 70, pp. 4–11.
- [10] N. Megiddo, C. H. Papadimitriou (1991). “On Total Functions, Existence Theorems and Computational Complexity.” *Theoretical Computer Science* 81, pp. 317–324.
- [11] J. Merschen (2012). “Nash Equilibria, Gale Strings, and Perfect Matchings.” PhD Thesis, London School of Economics and Political Science.
- [12] J. F. Nash (1951). “Noncooperative games.” *Annals of Mathematics*, 54, pp. 289–295.
- [13] 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.
- [14] R. Savani, B. von Stengel (2006). “Hard-to-solve Bimatrix Games.” *Econometrica* 74, pp. 397–429.



better other article, “Exponentially many steps for finding a NE in a bimatrix game.” In the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2004.?

- [15] R. Savani, B. von Stengel (2015). “Unit Vector Games.” arXiv:1501.02243v1 [cs.GT]
  - [16] L. S. Shapley (1974). “A Note on the Lemke-Howson Algorithm.” *Mathematical Programming Study 1: Pivoting and Extensions*, pp. 175–189
  - [17] L. Végh, B. von Stengel “Oriented Euler Complexes and Signed Perfect Matchings.” arXiv:1210.4694v2 [cs.DM]
- textbooks - papad, ziegler, sth gth