1 Introduction

2 Complexity, Games, Polytopes and Gale Strings

2.1 The Complexity Classes P and PPAD

2.2 Normal Form Games and Nash Equilibria

file: background

2.3 Some Geometrical Notation

2.4 Bimatrix Games, Labels and Polytopes

Theorem 1. *Let $l : [n] \to [m]$, 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*

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

$$Q^l = \{y \in \mathbb{R}^n | y \geq \mathbf{0}, \ \sum_{\substack{j \in N_i \\ i \in [m]}} y_j \leq 1\} \tag{2}$$

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)

file: polytopes

2.5 Cyclic Polytopes and Gale Strings

We have now built a correspondence between labeled best response polytopes and bimatrix games. We now consider a special case of games, where the polytope in theorem 1 is the dual of a particular kind of polytope that can be represented as a combinatorial structure. We first give the definition of these particular polytopes, called *cyclic polytopes*, then we define the combinatorial structure that we will use to study them, the *Gale strings*.

A *cyclic polytope* P in dimension d with n vertices is the convex hull of n distinct points $\mu_d(t_j)$ on the *moment curve* $\mu_d : \ t \mapsto (t, t^2, \ldots, t^d)^\top$ for $j \in [n]$ such that $t_1 < t_2 < \cdots < t_n$.

example(s) with graphics! There's one in Ziegler

A *bitstring* is a string of labels that are either 0s or 1s. Formally: given an integer k and a set S, we can represent the function $f_s : [k] \to S$ as the string $s = s(1)s(2)\cdots s(k)$; we have a bitstring in the case where $S = \{0, 1\}$. A maximal substring of consecutive 1's in a bitstring is called a *run*.

A *Gale string of length n and dimension d* is a bitstring of length n, denoted as $s \in G(d, n)$, such that

1. exactly d bits in s are 1 and
2. (*Gale evenness condition*)

$$01^k0 \text{ is a substring of } s \quad \Longrightarrow \quad k \text{ is even.} \tag{3}$$

The Gale evenness condition characterises Gale strings in $G(d, n)$ as the bitstrings of length n with exactly d elements equal to 1, such that *interior* runs (that is, runs bounded on both sides by 0s) must be of even length. In general, this condition allows Gale strings to start or end with an odd-length run; but when d is even this means that s starts 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 "loop" obtained by "glueing together" the extremes of the string to form an even run. 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 2.1. For instance, as a case where d is even, we have

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

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).

Example 2.2. As a case where d is odd, we consider

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

Note how 0**1011** is a cyclic shift of **1011**0, but it is not a Gale string.

The relation between cyclic polytopes and Gale strings is given by the following theorem by Gale [6].

Theorem 2 ([6])**.** *For any positive integers d, n let P be the cyclic polytope in dimension d with n vertices. Then the facets of P are encoded by $G(d, n)$; that is,*

$$F \text{ is a facet of } P$$

$$\iff$$

$$F = \text{conv}\{\mu(t_j) \mid s(j) = 1 \textit{ for some } j \in [n] \textit{ and } s \in G(d, n)\}$$

pf - see Ziegler
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 - pg 35 JM has nice one

We now give define a *labeling* for Gale strings, corresponding to the labeling of best-response polytopes.

A string s is *completely labeled* for some labeling function $l : [n] \to [d]$ if $\{\bar{l} \in [d] | s(i) = 1 \text{ and } l(i) = \bar{l} \text{ for some } i \in [n]\} = [d]$. If $s \in G(d, n)$, this implies that for every $\bar{l} \in [d]$ there is exactly one $s(i)$ such that $s(i) = 1$ and $l(i) = \bar{l}$, since there are exactly d positions such that $s(i) = 1$.

Example 2.3. Given the string of labels $l = 123432$, there are four associated completely labeled Gale strings: 111100, 110110, 100111 and 101101.

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

Sometimes it is not possible to find a completely labeled Gale string.

Example 2.4. 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.

By theorem 2, the Gale strings $s \in G(d, n)$ encode the facets of a cyclic polytope in dimension d with n vertices. Let $S = P \times Q$ be the best-response polytope for a unit vector game (U, B) as in theorem 1,

We now give a correspondence between the labeling of the facets of a cyclic polytope chosen as best-response polytope Q for a unit vector game (U, B) as in theorem 1 and a Gale string $s \in G(d, d + n)$.

We look for: labeling $l : [d + n] \to [d]$ for each vertex $\mu(t_j)$, where $j \in [d + n]$.

how labeled GS are corresponding to br cyclic polytopes

NB: we have given DUAL version (SvS 15) of Balthasar!!! vertices ¡-¿ facets

graphics

Another Gale

For this cyclic polytope P, a labeling $l : [n] \to [d]$ can be understood as a label $l(j)$ for each vertex $\mu(t_j)$ for $j \in [n]$. A completely labeled Gale string s therefore represents a facet F of P that is completely labeled.

When Q is a cyclic polytope in dimension d with $d + n$ vertices, then the string of labels $l(1) \cdots l(n)$ in Theorem 1 defines a labeling $l' : [d + n] \to [d]$

where $l'(i) = i$ for $i \in [d]$ and $l'(d+j) = l(j)$ for $j \in [n]$. In other words, the string of labels $l(1) \cdots l(n)$ is just prefixed with the string $1\,2 \cdots d$ to give l'. Then l' has a trivial completely labeled Gale string $1^d 0^n$ which defines the facet F_0.

From this point forward, we will assume that d is even.

Then the problem Another Gale defines exactly the problem of finding a Nash equilibrium of the unit vector game (I, B).

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

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

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

textbooks - papad, ziegler, sth gth