Finding Gale Strings

Marta M. Casetti, Julian Merschen, Bernhard von Stengel

Dept. of Mathematics, London School of Economics, London WC2A 2AE, United Kingdom

Abstract

The problem 2-NASH of finding a Nash equilibrium of a bimatrix game belongs to the complexity class PPAD. This class comprises computational problems that are known to have a solution by means of a path-following argument. For bimatrix games, this argument is provided by the Lemke–Howson algorithm. It has been shown that this algorithm is worst-case exponential with the help of dual cyclic polytopes, where the algorithm can be expressed combinatorially via labeled bitstrings defined by the "Gale evenness condition" that characterize the vertices of these polytopes. We define the combinatorial problem ANOTHER COMPLETELY LABELED GALE STRING whose solutions define the Nash equilibria of games defined by cyclic polytopes, including games where the Lemke–Howson algorithm takes exponential time. If this problem was PPAD-complete, this would imply that 2-NASH is PPAD-complete, in a much simpler way than the currently known proofs, including the original proof by Chen and Deng [3]. However, we show that ANOTHER COMPLETELY LABELED GALE STRING is solvable in polynomial time by a simple reduction to PERFECT MATCHING in graphs, making it unlikely to be PPAD-complete. Although this result is negative, we hope that it stimulates research into combinatorially defined problems that are PPAD-complete and imply this property for 2-NASH.

Keywords: Nash Equilibria, Lemke-Howson, Gale Evenness Strings,
Complexity, Perfect Matching, Polynomial Time Algorithm

1. Labeled Gale strings

Let $[k] = \{1, \ldots, k\}$ for any positive integer k. If S is a set, we often consider a function $s : [k] \to S$ as a string $s(1)s(2)\cdots s(k)$ of k elements of S. For $s : [k] \to S$ and a subset A of $[k]$, let $s(A)$ be the set $\{s(i) \mid i \in [k]\}$. If $S = \{0, 1\}$, we call

Email address: `m.m.casetti@lse.ac.uk`, `j.merschen@lse.ac.uk`, `stengel@maths.lse.ac.uk` (Marta M. Casetti, Julian Merschen, Bernhard von Stengel)

s a string of *bits*. A bit string $s : [k] \to \{0,1\}$ can be considered as an indicator function of a subset of $[k]$ that we denote by $1(s)$, that is,

$$1(s) = s^{-1}(1) = \{ j \in [k] \mid s(j) = 1\}.$$

Definition 1. *$G(d,n)$ is the set of all strings s of n bits so that exactly d bits in s are* 1 *and so that s fulfills the* Gale evenness condition*, that is, whenever* 01^k0 *is a substring of s, then k is even. An element of of $G(d,n)$ is also called a* Gale string *of* dimension *d (and length n).*

For example, $G(4,6)$ consists of the nine strings 111100, 111001, 110110, 110011, 101101, 100111, 011110, 011011, 001111.

For a bit string s, a maximal substring of s of consecutive 1's is called a *run*. A Gale string may only have interior runs (bounded on both sides by a 0) of even length but may start or end with an odd(-length) run. If d is even, then any s in $G(d,n)$ that starts with an odd run also ends with an odd run, and these two odd runs may be "glued together" to form an even run. This shows that the set of Gale strings of even dimension is invariant under a cyclic shift of the strings. We normally assume that d is even.

Given a set G of bit strings of length n and a parameter d, a *labeling* is a function $l : [n] \to [d]$. Given a labeling, a string s in G is called *completely labeled* if $l(1(s)) = [d]$, that is, if every label in $[d]$ appears as $l(i)$ for at least one bit $s(i)$ so that $s(i) = 1$. Clearly, if s is completely labeled, then s has at least d bits that are 1, and if exactly d bits in s are 1, then every label in $[d]$ occurs exactly once.

We consider the following decision problem.

COMPLETELY LABELED GALE STRING
Input: A labeling $l : [n] \to [d]$, where d is even and $d < n$.
Question: Is there a Gale string s in $G(d,n)$ that is *completely labeled*?

For example, for the string of labels $l = 1123143$ (with $d = 4$) the completely labeled Gale strings are 0110011 and 0011110. For $l = 123432$ they are 111100, 110110, 100111, and 101101. For $l = 121314$, there are no completely labeled Gale strings.

The set $G(d,n)$ of Gale strings has a combinatorial structure that allows the use of a "parity argument", which we consider in detail later, to show the following known property; it holds for odd d as well but we assume throughout that d is even.

Theorem 2. *For any labeling $l : [n] \to [d]$, where d is even and $d < n$, the number of completely labeled Gale strings in $G(d,n)$ is even.*

Theorem 2 implies that if there is one completely labeled Gale string, there is also a second one. The following function problem asks to compute a completely labeled Gale string if one such string is already given.

ANOTHER COMPLETELY LABELED GALE STRING

Input: A labeling $l : [n] \to [d]$, where d is even and $d < n$, and a completely labeled Gale string s in $G(d,n)$.

Output: A completely labeled Gale string s' in $G(d,n)$ where $s' \neq s$.

The main result of this paper is that both problems, COMPLETELY LABELED GALE STRING and ANOTHER COMPLETELY LABELED GALE STRING, can be solved in polynomial time. The proof uses a reduction to the following problem, which was first shown to be solvable in polynomial time by Edmonds [5].

PERFECT MATCHING

Input: Graph $G = (V,E)$.

Question: Is there 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 3. *The problems* COMPLETELY LABELED GALE STRING *and* ALMOST COMPLETELY LABELED GALE STRING *can be solved in polynomial time.*

Proof. We give a rather simple reduction to PERFECT MATCHING. Given the labeling $l : [n] \to [d]$, construct the (multi-)graph G with vertex set $V = [d]$ and up to n (possibly parallel) edges with endpoints $l(i), l(i+1)$ for $i \in [n]$ whenever these endpoints are distinct (so G has no loops); here we let $n+1 = 1$ ("modulo n") so that $n, n+1$ is to be understood as $n, 1$. Then a completely labeled Gale string s in $G(d,n)$ splits into a number of runs which are uniquely split into $d/2$ pairs $i, i+1$ so that the labels $l(i)$ and $l(i+1)$ are distinct, and all labels $1, \ldots, n$ occur among them. So this defines a perfect matching for G.

Conversely, a perfect matching M of G defines a Gale string s where $s(i) = s(i+1) = 1$ if the edge that joins $l(i)$ and $l(i+1)$ is in M and $s(i) = 0$ otherwise, so s is completely labeled. This shows how COMPLETELY LABELED GALE STRING reduces to PERFECT MATCHING. Finding a perfect matching, or deciding that G has none, can be done in polynomial time [5].

The reduction for ANOTHER COMPLETELY LABELED GALE STRING is an extension of this. Consider the given completely labeled Gale string s and the matching M for it. If G has multiple edges between two nodes and one of them is in M, simply replace that edge by a parallel edge to obtain another completely labeled Gale string s'. Hence, we can assume that M has no edges that have a parallel edge. Another completely labeled Gale string s' exists by Theorem 2.

The corresponding matching M' does not use at least one edge in M. Hence, at least one of the $d/2$ graphs G which have one of the edges of M removed has a perfect matching M', which is a perfect matching of G, and which defines a completely labeled Gale string s' different from s. The search for M' takes again polynomial time. □

The significance of Theorem 3 is to be understood in the context of equilibrium computation for games, which we discuss next. The remainder of this paper contains only known results.

2. Labeled polytopes and equilibria in games

For a matrix A its transpose is $A^\top$. We treat vectors u, v in $\mathbb{R}^d$ as column vectors, so $u^\top v$ is their scalar product. By $\mathbf{0}$ we denote a vector of all 0's, of suitable dimension, by $\mathbf{1}$ a vector of all 1's. A unit vector, which has a 1 in its ith component and 0 otherwise, is denoted by e_i. Inequalities like $u \geq \mathbf{0}$ hold for all components. For a set of points S we denote its convex hull by $\operatorname{conv} S$.

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 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 $\operatorname{conv} F$ of a set F of d vertices of P that lie on a hyperplane $\{x \in \mathbb{R}^d \mid a^\top x = a_0\}$ so that $a^\top u < a_0$ 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.

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

$$F \text{ is a facet of } P \quad \Longleftrightarrow \quad F = \operatorname{conv}\{\mu(t_i) \mid i \in 1(s)\} \text{ for some } s \in G(d,n),$$

as shown by Gale [7]. 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.

The following theorem, due to Balthasar and von Stengel [1, 2], establishes a connection between general labeled polytopes and equilibria of certain $d \times n$ bimatrix games (U, B).

Theorem 4. *Consider a labeled d-dimensional simplicial polytope Q with $\mathbf{0}$ in its interior, with vertices $-e_1, \ldots, -e_d, c_1, \ldots, c_n$, so that $F_0 = \operatorname{conv}\{-e_1, \ldots, -e_d\}$ is* *a facet of Q. Let $-e_i$ have label i for $i \in [d]$, and let c_j have label $l(j) \in [d]$ for $j \in [n]$. Let (U,B) be the $d \times n$ bimatrix game with $U = [e_{l(1)} \cdots e_{l(n)}]$ and $B = [b_1 \cdots b_n]$, where $b_j = c_j/(1+\mathbf{1}^\top c_j)$ for $j \in [n]$. Then the completely labeled facets F of Q, with the exception of F_0, are in one-to-one correspondence to the Nash equilibria (x,y) of the game (U,B) as follows: if v is the normal vector of F, then $x = (v+\mathbf{1})/\mathbf{1}^\top(v+\mathbf{1})$, and $x_i = 0$ if and only if $-e_i \in F$ for $i \in [d]$; any other label j of F, 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 pure best replies to x.*

In the preceding theorem, any simplicial polytope can take the role of Q as long as it has one completely labeled facet F_0. Then an affine transformation, which does not change the incidences of the facets of Q, can be used to map F_0 to the negative unit vectors $-e_1,\ldots,-e_d$ as described, with Q if necessary expanded in the direction $\mathbf{1}$ so that $\mathbf{0}$ is in its interior.

A $d \times n$ bimatrix game (U,B) is a *unit vector game* if all columns of U are unit vectors. For such a game B with $B = [b_1 \cdots b_n]$, the columns b_j for $j \in [n]$ can be obtained from c_j as in Theorem 4 if $b_j > \mathbf{0}$ and $\mathbf{1}^\top b_j < 1$. This is always possible via a positive-affine transformation of the payoffs in B, which does not change the game. The unit vectors $e_{l(j)}$ that constitute the columns of U define the labels of the vertices c_j. The corresponding polytope with these vertices is simplicial if the game (U,B) is nondegenerate [15], which here means that no mixed strategy x of the row player has more than $|\{i \in [d] \mid x_i > 0\}|$ pure best replies. Any game can be made nondegenerate by a suitable “lexicographic” perturbation of B, which can be implemented symbolically.

Unit vector games encode arbitrary bimatrix games: An $m \times n$ bimatrix game (A,B) with (w.l.o.g.) positive payoff matrices A,B can be symmetrized so that its Nash equilibria are in one-to-correspondence to the symmetric equilibria of the $(m+n) \times (m+n)$ symmetric game $(C^\top,C)$ where

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

In turn, as shown by McLennan and Tourky [10], the symmetric equilibria (x,x) of any symmetric game $(C^\top,C)$ are in one-to-one correspondence to the Nash equilibria (x,y) of the “imitation game” (I,C) where I is the identity matrix; the mixed strategy y of the second player is simply the uniform distribution on the set $\{i \mid x_i > 0\}$. Clearly, I is a matrix of unit vectors, so (I,C) is a special unit vector game.

Special games are obtained by using cyclic polytopes in Theorem 4, suitably affinely transformed with a completely labeled facet F_0. 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 4 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. Then the problem ANOTHER COMPLETELY LABELED GALE STRING defines exactly the problem of finding a Nash equilibrium of the unit vector game (I,B). Note again that B is here not a general matrix (which would define a general game) but obtained from the last n of $d+n$ vertices of a cyclic polytope in dimension d.

3. Lemke–Howson and PPAD

The algorithm due to Lemke and Howson [9], here called the LH algorithm, finds one Nash equilibrium of a bimatrix game. It can be translated to labeled simplicial polytopes as follows. Start with a completely labeled facet (such as F_0 above). Select one label i that is allowed to be missing (or "dropped") and move from F_0 to the unique adjacent facet that shares all vertices with F_0 except the one with label i. This is computationally implemented as a *pivoting* step as in the simplex algorithm, which is a local transformation of the current normal vector by using the other vertices not on the current facet. The newly obtained facet F_1, say, has a new vertex with a label j; if $j = i$, then F_1 is completely labeled and the algorithm stops. Otherwise, take the vertex v of F_1 that had label j so far and move to the unique adjacent facet F_2 that has all vertices of F_1 except v, and continue as before. This defines a unique "path" of facets that must eventually terminate at a completely labeled facet different from F_0. Applied to cyclic polytopes, this proves Theorem 2.

The result of Morris [11] implies that for suitably labeled cyclic polytopes in dimension d with $2d$ vertices, the described path can be exponentially long, for *any* initially dropped label. His labeling l for $d = 6$ is given by the string of labels 123456 645321, for $d = 8$ it is 12345678 86745231, which shows the general pattern. With the help of imitation games [10], this defines exponentially long LH paths for bimatrix games. Savani and von Stengel [13] obtained this result differently by considering payoff matrices that for both players are defined via cyclic polytopes, rather than a matrix of unit vectors for the row player as in Theorem 4.

The problem n-NASH of computing Nash equilibrium of an n-player game belongs to the complexity class PPAD [12]. It comprises function problems that are known to have a solution via a “polynomial parity argument with direction”. For 2-NASH, this argument is provided by the LH algorithm. Formally, PPAD consists of problems that reduce to the problem END OF THE LINE, given by two polynomial-sized Boolean circuits σ and π with k input and k output bits. This pair σ, π defines an implicit digraph with k-bit strings as vertices and arcs (u, v) whenever $\sigma(u) = v$ and $\pi(v) = u$. If $\sigma(\pi(u)) \neq u$ then u is a source and if $\pi(\sigma(v)) \neq v$ then v is a sink of this digraph. It is assumed that 0^k is a source. The sought output is any sink, or source other than 0^k. It exists because the digraph is a collection of directed paths and cycles, with at least one path which starts at 0^k.

Daskalakis, Goldberg and Papadimitriou [4] and Chen and Deng [3], respectively, have shown that 3-NASH and 2-NASH are PPAD-complete. As indicated in Section 2, the PPAD-completenes of 2-NASH due to [3] shows that the following problem is PPAD-complete: Given a labeled polytope Q as in Theorem 4 with a completely labeled facet F_0, find another completely labeled facet. (If the games in [3] are degenerate then Q is not simplicial; this can be treated by a suitable extension of the LH algorithm.)

The orientation of the path (the “D” in PPAD) can be proved by a suitable orientation of the facets of the polytope, via the determinant of their d vertices in the order of the labels [8, 14].

For the special cyclic polytopes, the LH algorithm can be described very simply in terms of the Gale evenness strings, see [13]. The orientation can also be defined simpler via signs of permutations rather than of determinants, which we omit for reasons of space.

Another abstraction of the LH algorithm is provided by Euler complexes or “oiks” introduced by Edmonds [6]. A special case are abstract manifolds, defined by a family of d-element sets called *rooms* so that any *wall*, obtained by removing one vertex from a room, is the wall of exactly one other room. Given a labeling (called coloring in [6]) of the vertices, any manifold has an even number of completely labeled rooms, in analogy to Theorem 2. If the manifold is orientable, the orientation argument of Lemke and Grotzinger [8] applies; in particular, the endpoints of the LH paths are rooms of opposite orientation.

What is important in our context is that the manifold is not defined as an explicit list of rooms but implicitly with rooms as facets of a simplicial polytope, given by its vertices. For the cyclic polytope with n vertices in dimension d, the rooms are even more simply specified as the sets $1(s)$ for $s \in G(d, n)$.

Our Theorem 3 shows: even though cyclic polytopes may give rise to exponentially long LH paths, the respective computational problem of finding another completely labeled facet is solvable in polynomial time. Hence, Gale evenness strings are most likely too simple to define a PPAD-complete problem.

References

[1] A. V. Balthasar (2009), Geometry and Equilibria in Bimatrix Games. PhD Thesis, London School of Economics.

[2] A. Balthasar, B. von Stengel (2010), Index and uniqueness of symmetric equilibria. In preparation.

[3] X. Chen, X. Deng (2006). Settling the complexity of two-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. *Proc. Ann. 38th STOC*, pp. 71–78.

[5] J. Edmonds (1965). Paths, trees, and flowers. *Canad. J. Math.* 17, pp. 449–467.

[6] J. Edmonds (2007). Euler complexes. Manuscript, 5 pp.

[7] D. Gale (1963), Neighborly and cyclic polytopes. In: Convexity, Proc. Symposia in Pure Math., Vol. 7, ed. V. Klee, American Math. Soc., Providence, Rhode Island, pp. 225–232.

[8] C. E. Lemke, S. J. Grotzinger (1976). On generalizing Shapley's index theory to labelled pseudomanifolds. Math. Programming 10, 245–262.

[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 (2009). Simple complexity from imitation games. *Games and Economic Behavior*, in press, doi:10.1016/j.geb.2009.10.003

[11] W. D. Morris Jr. (1994). Lemke paths on simple polytopes. *Math. Oper. Res.* 19, pp. 780–789.

[12] 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.

[13] R. Savani, B. von Stengel (2006). Hard-to-solve bimatrix games. *Econometrica* 74, pp. 397–429.

[14] M. J. Todd (1976), Orientation in complementary pivot algorithms. *Math. Oper. Res.* 1, pp. 54–66.

[15] B. von Stengel (2002). Computing equilibria for two-person games. In: *Handbook of Game Theory, Vol. 3,* eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, pp. 1723–1759.