

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-subsection

2.3 Bimatrix Games and Best Response Polytopes

file: polytopes-subsection

2.4 Cyclic Polytopes and Gale Strings

Example 2.1. For $l = 121314$, there are no completely labeled Gale strings.

2.5 Labeling and the Problem ANOTHER GALE

Theorem 1. *For any labeling $l : [n] \rightarrow [d]$, where d is even and $d < n$, the number of completely labeled Gale strings associated with l is even.*

file: gale-def-subsection

3 Algorithmic and Complexity Results

3.1 Pivoting

3.2 The Lemke-Howson Algorithm and Parity

file: pivoting-LH-subsection

3.3 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.

Definition 1. 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 .

We define the problem PERFECT MATCHING as follows:

PERFECT MATCHING

input : A graph $G = (V, E)$.

question : Is there a perfect matching for G ?

The complexity of PERFECT MATCHING has been proven to be in P by Edmonds [4].

Theorem 2 ([4]). *The problem PERFECT MATCHING 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 2.

GALE

input : A labeling $l : [n] \rightarrow [d]$, 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 3. *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 2.4, so $n + 1 = 1$.

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

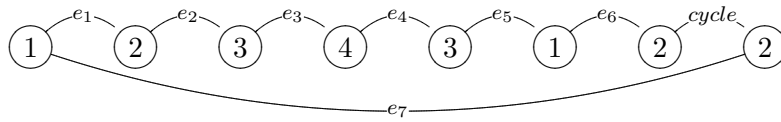
Let $s \in G(d, n)$ be a completely labeled Gale string. Then every run of s splits uniquely into $d/2$ pairs $(i, i+1)$ such that the labels $l(i)$ satisfy the condition $l(i) \neq l(i+1)$, and all the labels $l(i) \in [d]$ occur. Then the labels will correspond to all the vertices of G , and the pairs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

Conversely, let $l : [n] \rightarrow [d]$ 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.

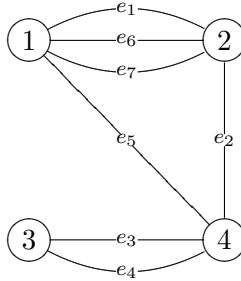
We have a reduction from GALE to the problem PERFECT MATCHING, which is polynomial-time solvable by theorem 2. Finding a Gale string for a given labeling, or deciding that there isn’t one, can therefore be done in polynomial time. \square

We give two examples of the construction used in theorem 3.

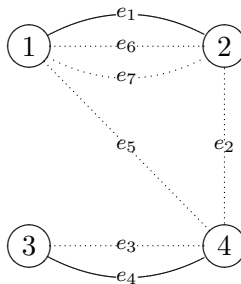
Example 3.1. Let $l = 12343122$ be a string of labels. Then the edges e_i of the graph G obtained from the construction in the proof of theorem 3 will be as follow:



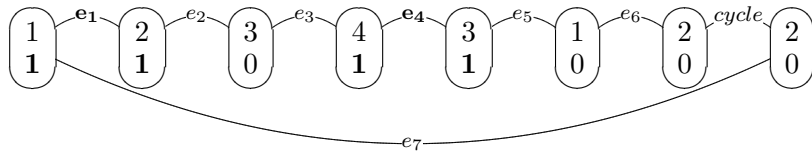
Given the vertices $v \in [4]$, the graph G will be:



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

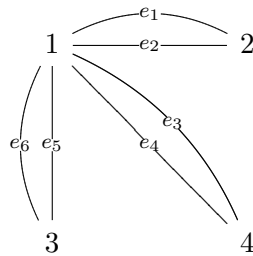


In turn, this corresponds to the completely labeled Gale string 11011000.



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 3.2. 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 2.1 that there isn't any possible completely labeled Gale string for the labeling l .

We finally extend the proof of theorem 3 to show that ANOTHER GALE is polynomial-time solvable.

Theorem 4. *The problem ANOTHER GALE is solvable in polynomial time.*

Proof. Let $l : [n] \rightarrow [d]$ 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 3, 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.

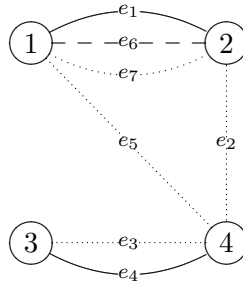
We now assume that all the edges in every perfect matching M for G don't have a parallel edge. Note that this condition is only on the edges in the matching; G can still be a multigraph.

Theorem 1 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 4.

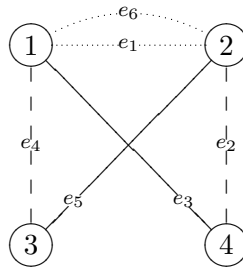
Example 3.3. We consider the labeling the string of labels $l = 1234312$. We have found in example 3.1 the completely labeled Gale string 1101100, corresponding to the perfect matching $M = \{e_1, e_4\}$ in the graph G .



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 3.4. 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 $s' = 001111$.



References

- [1] M. M. Casetti, J. Merschen, B. von Stengel (2010). Finding Gale Strings. *Electronic Notes in Discrete Mathematics* 36, pp. 1065 – 1082.
- [2] X. Chen, X. Deng (2006). Settling the complexity of two-player Nash equilibrium. *Proc. 47th FOCS*, pp. 261–272.
- [3] C. Daskalakis, P. W. Goldberg, C. H. Papadimitriou (2006). The complexity of computing a Nash equilibrium. *Proc. Ann. 38th STOC*, pp. 71–78.
change ref to econometrica(?)
- [4] J. Edmonds (1965). Paths, trees, and flowers. *Canad. J. Math.* 17, pp. 449–467.
- [5] 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.
- [6] J. Merschen (2012). Nash Equilibria, Gale Strings, and Perfect Matchings. PhD Thesis, London School of Economics and Political Science.
- [7] C. E. Lemke, J. T. Howson, Jr. (1964). Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Mathematics* 12, pp. 413–423.
- [8] N. Megiddo, C. H. Papadimitriou (1991). On total functions, existence theorems and computational complexity. *Theoretical Computer Science* 81, pp. 317 – 324.
- [9] 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.
- [10] R. Savani, B. von Stengel (2006). Hard-to-solve bimatrix games. *Econometrica* 74, pp. 397–429.
- [11] J. Vègh, B. von Stengel
ref