Challenge Instances for NASH

Rahul Savani

Department of Mathematics, London School of Economics
Houghton St, London WC2A 2AE, United Kingdom
email: rahul@maths.lse.ac.uk

October 15, 2004

CDAM Research Report LSE-CDAM-2004-03
(PRELIMINARY VERSION)

Abstract. The problem NASH is that of finding one Nash equilibrium of a bimatrix game. The computational complexity of this problem is a long-standing open question. The Lemke–Howson algorithm is the classical algorithm for NASH. In an earlier paper, this algorithm was shown to be exponential, even in the best case, for square bimatrix games using dual cyclic polytopes. However the games constructed there are easily solved by another standard method, support enumeration. In this paper we present "challenge instances" for NASH. We extend the use of dual cyclic polytopes and construct a class of games which are shown to be hard to solve for both the Lemke–Howson algorithm and support enumeration. Other general methods for NASH are discussed. It is not obvious that they could solve these games efficiently.

1 Introduction

NOTE: This paper extends [13]. The reader is referred to that paper for the required background material as we have tried not to duplicate such material here.

The problem NASH is that of finding one Nash equilibrium (NE) of a bimatrix game. The computational complexity of this problem is a classic open question [10]. Instances from well-known classes of bimatrix games can be solved efficiently by one of the standard methods available. In this paper we present *challenge instances* for NASH, that is, games for which no efficient solution method is known.

The main algorithm for NASH is the Lemke–Howson (LH) algorithm [4], which is a pivoting algorithm similar to the simplex algorithm for linear programming. The algorithm is started by choosing a pure strategy of one of the players. For each such choice the algorithm progresses deterministically, giving rise to a different LH path, which terminates at some equilibrium of the game. In [13] a class of $\mathtt{d} \times \mathtt{d}$ games, for even $\mathtt{d}$, is presented with the property that the lengths of *all* LH paths grow exponentially with the dimension of the game. Thus the LH algorithm was shown to be inefficient for NASH. However, the games

presented in [13] are easy to solve by another method, *support enumeration*, and so do not serve as challenge instances.

The *support* of a mixed strategy is the set of pure strategies that have positive probability under that strategy. A *support profile* for a bimatrix game is a pair of supports, one for each player. It is easy to efficiently check if a NE with a particular support profile exists. One method for NASH is support enumeration: Support profiles are enumerated one by one and the existence of a corresponding NE is checked. The games constructed in [13] possess a unique equilibrium, where both players use full support, that is, all pure strategies are used with positive probability. Therefore these games cannot serve as challenge instances as this support profile is guessed easily; there is no way to "hide" the equilibrium, since both players use full support.

Random games, generated with payoff matrix entries drawn from uniform distributions, tend to possess pure strategy equilibria. For a recent study on the expected number of pure strategy equilibria in random games, and related work, see [12] and the references therein. A pure NE will be found by some LH path in a single step, and could easily be found by support enumeration. Thus, these random games do not provide challenge instances for NASH.

A recent study [6] includes a taxonomy of many "interesting" classes of bimatrix games from the literature. In [11] tests show that support enumeration tends to work well on (random) instances from all the classes of [6], and so none of these can serve as challenge instances.

In this paper we extend the construction in [13] to create $d \times 2d$ games where all LH paths are exponentially long. These games possess exponentially many equilibria, where each player uses d strategies, but their supports form an exponentially small fraction of all $\binom{2d}{d}$ supports of player 2 of size d. We show that support enumeration takes an exponential expected number of attempts to find an NE.

Another method of finding equilibria is to enumerate the vertex pairs of the best response polyhedra, and test them for the equilibrium property. In general, the number of vertices is exponential. However, there are many vertex enumeration methods, and we do not analyse whether they solve our games in expected exponential time. In fact, knowing the exact way that our games are constructed, it is possible to "unscramble" the hidden support by special pivoting steps, and thus arrive quickly at an equilibrium. However, this very specialized method only applies to our construction, and does not compute an equilibrium for any other games. Hence, we consider the games constructed here as suitable challenge instances for general algorithms for NASH. This is discussed further in Section 7. For an introduction to bimatrix games, their equilibria, the relevant complexity classes, a detailed description of the LH algorithm, and further discussion of related approaches to NASH see [13] and [16].

The next section outlines the construction, and describes the equilibria of the games. In Section 3 we show that all the LH paths are exponentially long. In Section 4 we show that support enumeration is inefficient for these games. We argue in Section 5 that general vertex enumeration may also be inefficient for these games, but that further analysis is required. A highly specialized pivoting method for efficiently finding a sample equilibrium, when the construction is known, is described in Section 6. We conclude in Section 7.

2 The construction

In this section we describe the construction of the challenge instances and their equilibria. We extend [13] and the reader is referred to that paper for more details. Let d be even. Here, unlike [13], we construct non-square games, where players 1 and 2 have d and $2d$ pure strategies respectively[1]. Now the best response polytopes of the two players are not identical, but are still both *dual cyclic polytopes* [18][3]. Player 1's best response polytope, P, is in dimension d with $3d$ facets, and player 2's, Q, is in dimension $2d$ with $3d$ facets. A standard method of obtaining these polytopes is described in [15] and [13].

We recall the Gale evenness condition [2]. It says that for a suitable ordering of the inequalities in the description of a dual cyclic polytope [13]: A bitstring with 1's in positions corresponding to binding inequalities represents a vertex if and only if any substring of the form $01\cdots10$ has even length, so 0110, 011110, etc., is allowed, but not 010, 01110, and so on. A maximal substring of 1's is called a *run*. We only consider *even* dimensions d and $2d$, where the allowed odd runs of 1's at both ends of the string can be glued together to form an even run, which shows the cyclic symmetry of the Gale evenness condition. Let $G(p,q)$ be the set of these Gale evenness bitstrings of length q with p ones. Vertices of P and Q are represented as bitstrings and their vertex sets are $G(d,3d)$ and $G(2d,3d)$, respectively.

The equilibrium condition and the LH algorithm depend on which facets a vertex belongs to, as encoded in the Gale evenness bitstrings in $G(d,3d)$ and $G(2d,3d)$, and on the facet labels. These are defined by permutations l and l' of $1,\ldots,3d$ for P and Q, respectively. For a vertex $u \in G(d,3d)$ of P its labels are given by $\{l(k) : u_k = 1\}$, and the labels of a vertex $v \in G(d,3d)$ of Q are $\{l'(k) : v_k = 1\}$, for $1 \le k \le 3d$. The kth facet of P (corresponding to the kth position in a bitstring) has label $l(k) = k$, so l is simply the identity permutation. The kth facet of Q has label $l'(k)$. The permutation l' has the fixed points $l'(1) = 1$ and $l'(d) = d$, and otherwise exchanges adjacent numbers, as follows:

[1] In fact we have observed that $d \times d + k$ games, with $k \ge 0$ even, constructed in an analogous way, with the labeling changed appropriately, yield exponentially long LH paths. The choice of $k = d$ means that in any equilibrium player 2 plays d out of $2d$ strategies, which is best for making the equilibrium support of player 2 hard to guess.

$$l'(k) = \begin{cases} k, & k = 1, d, \\ k + (-1)^k, & 2 \le k \le d-1, \\ k - (-1)^k, & d+1 \le k \le 3d. \end{cases} \quad (1)$$

Notice that the number of facets in [13] was $2d$ and is now $3d$ but otherwise the setup is completely analogous. Before looking at the LH paths in the next section, we study the equilibria of these games. The artificial equilibrium $\hat{e}_0^d$ is a vertex pair $(u, v) \in P \times Q$, so that u has labels $1, \ldots, d$ and v labels $d+1, \ldots, 3d$. In terms of bitstrings, $u = 1^d 0^{2d}$ (which are d ones followed by $2d$ zeros) and $v = 0^d 1^{2d}$, so that

$$\hat{e}_0^d = (1^d 0^{2d}, 0^d 1^{2d}) \in G(d, 3d) \times G(2d, 3d). \quad (2)$$

The equilibria of the games, which correspond to the complementary vertex pairs of $P \times Q$, excluding $\hat{e}_0^d$, are characterized by the following two lemmas. Lemma 1 says that in any NE player 1 uses full support.

Lemma 1 *In any complementary vertex pair* (u, v) *of* $P \times Q$*, except* $\hat{e}_0^d$*,* $u_1 = \cdots u_d = 0$.

Proof. Suppose that (u, v) is a complementary vertex pair of $P \times Q$ other than $\hat{e}_0^d$. Assume that $u_d = 0$. Now $u_d = 0$ implies $v_d = 1$ by complementarity. We show that this means $u_1 = u_2 = \cdots = u_d = 0$. Suppose not. Consider the largest k, $0 < k < d$, such that $v_k = 0$. If k is odd then $v_{d+1} = 1$ by Gale evenness. Then $u_{d+2} = 0$ by complementarity and so $u_{d+1} = 0$ by Gale evenness. Then $v_{d+2} = 1$ by complementarity. By repeatedly using the Gale evenness condition and complementarity in this way, we see the $v_{k+1} = \cdots = v_{3d} = 1$. This is a contradiction. If k is even we see that $u_{k+1} = 1$ but $u_k = u_{k+2} = 0$, which violates Gale evenness. Similarly, it can be shown that there is no complementary pair with $u_d = 1$, other than $\hat{e}_0^d$.

□

Let $d = 2l$. Let $S(l)$ be the set of bitstrings of length $4l$, containing $2l$ ones, and composed only of contiguous substrings 00, 11, and 0110. If $s \in S$ then let $\bar{s}$ be the bitstring of length $4l$ with a one in the ith position if and only if s contains a zero in the ith position. The set of complementary vertex pairs of $P \times Q$ is characterized by lemma 2, the proof of which is omitted, since given Lemma 1, it is very similar to Proposition 3.2 of [15].

Lemma 2 *Every complementary vertex pair of* $P \times Q$*, except* $\hat{e}_0^d$*, is of the form* $(0^d s, 1^d \bar{s})$ *where* $s \in S(l)$.

If a string in $S(l)$ contains k substrings 00, $0 \le k \le l$, then it contains the same number of substrings 11, and $l - k$ substrings 0110. These substrings may be arranged in

any manner, with $(l+k)!/(k!\,k!\,(l-k)!)$ many possibilities. Hence,

$$\sigma(l) = \sum_{k=0}^{l} \frac{(l+k)!}{k!\,k!\,(l-k)!} = \sum_{k=0}^{l} \binom{l+k}{k}\binom{l}{k}.$$

An asymptotic approximation, which is proved in [15], states that

$$\sigma(l) \sim \tilde{\sigma}(l) := \frac{1+\sqrt{2}}{2^{5/4}\sqrt{\pi}} \frac{(1+\sqrt{2})^{2l}}{\sqrt{l}} \approx 0.81\frac{2.414^d}{\sqrt{d}}.$$

Suppose that, using the labelings l and l', we generate the $d \times 2d$ bimatrix game (A, B) according to Proposition 2.1 of [15], which gives details on how to generate the payoff matrices of a bimatrix game corresponding to a pair of labeled polytopes with a least one complementary vertex pair. Then the order of facets given by taking the rows and then the columns of A or B in order is consistent with the Gale evenness condition. Thus Lemma 2 could be used to immediately find an NE of (A, B): When presented with a $d \times 2d$ game, a trivial addition to any general algorithm for NASH is to check if there is an NE corresponding to the support profile where player 1 uses full support and player 2 uses her *first* d strategies, which indeed corresponds to an equilibrium of (A, B). Since our goal is to construct "hard to solve" bimatrix games, we therefore disguise the structure of the equilibria, in order to ensure that Lemma 2 cannot be used to find a sample equilibrium quickly.

There is no way to "hide" a strategy that uses full support, which was precisely the limitation, in terms of support guessing, of the construction in [13]. In the present construction we can however "hide" the equilibrium supports of player 2, which use only d out of $2d$ possible pure strategies. This is done by randomly permuting the payoff matrices. Although it is only necessary to permute the columns of the payoff matrices to hide the structure of player 2's equilibrium supports, we permute the rows as well. The reason for this is that vertex enumeration depends upon the indexing of variables, which is affected by permuting the rows (see Section 5).

Let $[d] = \{1, \ldots, d\}$, and let S_d be the set of permutations of $[d]$, and define $[2d]$ and S_{2d} similarly. We choose $\pi_d \in S_d$ according to a uniform distribution on S_d, and similarly choose $\pi_{2d} \in S_{2d}$. We randomly permute the rows of A and B according to π_d and the columns according to π_{2d} to produce $(\overline{A}, \overline{B})$, which is an instance of our construction.

3 The LH paths

The LH paths[2] in the present construction are obtained as mappings of the paths in [13]. For each LH path, d contiguous bits of every vertex in Q on the path are constant at 1. (The Gale

[2]Permuting the payoff matrices amounts to a relabeling (see [13]) and does not affect the LH paths beyond permuting all the bitstrings that represent vertices. For simplicity we study the LH paths using the unpermuted labeling, which is consistent with the Gale evenness condition.

evenness condition shows that such intersections of facets are also dual cyclic polytopes in lower dimension.) The following lemmas describe the paths in this new construction through their relationship to the paths in [13].

Denote by $\rho(d,k)$ the LH path, for the present construction, when label k is dropped from $\hat{e}_0^d$ in $P \times Q$. The path is regarded as a sequence $(u^0,v^0)\ (u^1,v^1)\ \cdots\ (u^M,v^M)$ of vertex pairs in $G(d,3d) \times G(2d,3d)$. Let $M(d,k) = M$ be the length of that path. We now call the best response polytopes from the construction in [13] $\hat{P}$ and $\hat{Q}$, using P and Q for the best response polytopes in the present construction. Following the definitions of [13], denote by $\pi(d,k)$ the path when label k is dropped from e_0^d in $\hat{P}$ and $\hat{Q}$, which is a sequence of vertex pairs in $G(d,2d) \times G(d,2d)$. Let $L(d,k)$ be the length of that path. These paths are studied and fully described in [13]. Lemma 3 is analogous to Theorem 8 (a) of [13] in both content and proof; it describes a relationship among the paths arising from the cyclic symmetry of the Gale evenness condition, and simplifies the work needed to describe the lengths of the paths for labels $2d+1 \le k \le 3d$.

Lemma 3 $M(d,k) = M(d,d{+}1{-}k)$ *and* $M(d,d{+}k) = M(d,2d{+}1{-}k)$, *for* $1 \le k \le d$.

Proof. Let ψ be defined by $\psi(k) = d-k+1$ for $1 \le k \le d$ and $\psi(d+k) = 3d-k+1$ for $1 \le k \le 2d$. Applied to the bitstrings of vertices, ψ is a cyclic shift by $2d$ positions to the right, followed by a reversal. This leaves the sets $G(d,3d)$ and $G(2d,3d)$ invariant. Furthermore, ψ commutes with the labelings l and l' of P and Q, so the LH algorithm proceeds in the same manner. That is to say, the vertex pairs on the path $\rho(d,k)$, seen as a pairs of 0-1 strings, are step by step the same when the positions in each string are permuted according to ψ. Under ψ, the first vertex pair $\hat{e}_0^d$ is mapped to itself, but the positions are changed as above. This means that under ψ, the path $\rho(d,k)$ is mapped to $\rho(d,d-k+1)$, so these two paths have the same length, and similarly for $\rho(d,d+k)$ and $\rho(d,2d+1-k)$. □

In the following three lemmas, ϵ, ζ, and η are mappings from the vertices of $\hat{P} \times \hat{Q}$ to the vertices of $P \times Q$. We represent the vertices as bitsrings so,

$$\epsilon, \zeta, \eta : G(d,2d) \times G(d,2d) \to G(d,3d) \times G(2d,3d).$$

Each mapping induces, in the obvious way, an injective mapping from the labels $1,\ldots,2d$ of $\hat{P} \times \hat{Q}$ to the labels $1,\ldots,3d$ of $P \times Q$.

Lemma 4 describes the paths for dropped label i, $1 \le i \le d$. Lemma 5 describes the paths for dropped labels $d+1 \le i \le 2d$ and i even. Then the paths for labels $2d+1 \le i \le 3d$ and i odd are described by Lemmas 5 and 3. Lemma 6 describes the paths for dropped labels with $2d+1 \le k \le 3d$ and k even. Then the paths for labels $d+1 \le k \le 2d$ with k odd are described by Lemmas 6 and 3. Thus all paths in the present construction are described by Lemmas 3,4,5, and 6 in terms of the paths in [13].

Lemma 4 $M(d,k) = L(d,k)$, *for* $1 \le k \le d$.

Proof. We show that $\rho(d,k) = \epsilon(\pi(d,k))$ for $k = 1,\ldots,d$, where

$$\epsilon(u,v) = \begin{cases} (u_1 \cdots u_{2d-k} 0^d u_{2d-k+1} \cdots u_{2d}, v_1 \cdots v_{2d-k} 1^d v_{2d-k+1} \cdots v_{2d}), & k \text{ even}, \\ (u_1 \cdots u_{2d-k+1} 0^d u_{2d-k+2} \cdots u_{2d}, v_1 \cdots v_{2d-k+1} 1^d v_{2d-k+2} \cdots v_{2d}), & k \text{ odd}. \end{cases}$$

The starting point of $\pi(d,k)$ is e_0^d and of $\rho(d,k)$ is $\hat{e}_0^d$. Now $\epsilon(e_0^d) = \hat{e}_0^d$ as required. Complementarity of the constant positions of ϵ is immediate. Both paths $\rho(d,k)$ and $\epsilon(\pi(d,k))$ start by dropping the same label, that is, considering ϵ as a mapping of labels, $\epsilon(k) = k$. Now ϵ cyclically preserves the adjacency of all labels except for a single pair in $\hat{P}$ and a single pair in $\hat{Q}$. (If $k = 1$ it preserves the adjacency of all pairs in $\hat{P}$ and $\hat{Q}$.) In $\hat{P}$ and $\hat{Q}$ these pairs correspond to positions $2d-k$ and $(2d-k+1 \mod 2d)$ if k is even and $2d-k+1$ and $(2d-k+2 \mod 2d)$ if k is odd. Now the proof of Theorem 8 c) of [13] shows that no "action" occurs across these positions. □

Lemma 5 $M(d,d+k) = L(d,d+k)$, *for* $1 \le k \le d$ *and* k *even.*

Proof. We show that $\rho(d,d+k) = \zeta(\pi(d,d+k))$ for $k = 1,\ldots,d$ and k even, where

$$\zeta(u,v) = (u_1 \cdots u_{d+k} 0^d u_{d+k+1} \cdots u_{2d}, v_1 \cdots v_{d+k} 1^d v_{d+k+1} \cdots v_{2d})$$

The starting point of $\pi(d,d+k)$ is e_0^d and of $\rho(d,d+k)$ is $\hat{e}_0^d$. Now $\zeta(e_0^d) = \hat{e}_0^d$ as required. Complementarity of the constant positions of ζ is immediate. Both paths $\rho(d,d+k)$ and $\zeta(\pi(d,d+k))$ start by dropping the same label, that is, considering ζ as a mapping of labels, $\zeta(d+k) = d+k$. Now ζ cyclically preserves the adjacency of all labels except a single pair in $\hat{P}$ and a single pair in $\hat{Q}$. In $\hat{P}$ and $\hat{Q}$ this pair corresponds to positions $d+k$ and $(d+k+1 \mod 2d)$. The proof of Theorem 8 d) of [13] shows no "action" occurs across these positions. □

Lemma 6 $M(d,2d+k) = L(d,1)+1$, *for* $1 \le k \le d$ *and* k *even.*

Proof. We show that $\rho(d,2d+k) = \hat{e}_0 + \eta(\pi(d,1))$ for $k = 1,\ldots,d$ and k even, where the "$+$" denotes joining $\hat{e}_0$ to that first vertex pair of $\eta(\pi(d,1))$ by an edge, and

$$\eta(u,v) = (u0^d, 1v_2 \cdots v_{2d} 1^{k-2} v_1 1^{d-k+1}).$$

In the first step of $\rho(d,2d+k)$ label $2d+k$ is dropped in Q, label 1 is picked up, and we reach the vertex pair (u^1,v^1). Notice that $\eta(u^1,v^1) = e_0^d$. Label 1 is duplicate in P and is dropped in the next step. Because no edge of the path $\pi(d,1)$ *wraps-around* (see [13]) the

rest of path $\rho(d, 2d+k)$ continues as $\eta(\pi(d,1))$. So label 1 is never dropped in Q, that is, $v_1^i = 1$ for all $i = 1, \ldots, M$. In the second to last step of $\pi(d,1)$ label d is picked up in $\hat{P}$ and then dropped in $\hat{Q}$ in the last step, as label 1 is picked up. In (u^{M-1}, v^{M-1}), the second to last vertex pair of $\rho(d, 2d+k)$, label d is duplicate, and in the last step it is dropped in Q. Now however, label 1 is already present in Q and this last edge wraps-around and the missing label $2d+k$ being picked up, thus terminating the path $\rho(d, 2d+k)$. □

We have now showed that the length of any LH path, $M(d,k)$, in the present construction is equal to the length of some path from [13], $L(d,k)$, possibly plus one. Thus the lengths of the LH paths in the present construction and [13] have the same exponential order of growth.

4 Support enumeration

In this section we show that support enumeration cannot be used to efficiently find a sample equilibrium of these games. Let $U(d)$ be the "universe" of all sets of d-subsets of $[1, ..., 2d]$, which corresponds to all player 2's supports of size d, and let $u(d) = |U(d)| = \binom{2d}{d}$. Let $N(d) \subset U(d)$ be the set of supports of player 2 corresponding to equilibria, and let $n(d) = |N(d)|$. In what follows we refer to U, N, u, and n for convenience, since d is fixed.

Suppose an oracle $\mathcal{O}$, presented with a support profile, answers the question of the existence of a corresponding equilibrium. The following lemma shows the power of randomly permuting the payoff matrices as a method for hiding equilibria from support enumeration: If N is known, a negative query to $\mathcal{O}$ rules out certain permutations, but the lemma says that this does not help to pick subsequent supports. In the following, $x \in U$ is a query to the oracle $\mathcal{O}$. For $x, y \in U$ and $\pi \in S_{2d}$ the condition $\pi(y) = x$ means that $\pi(i) \in x$, $\forall i \in y$.

Lemma 7 *$\forall x \in U$, we have,*

$$\frac{|\{\pi \in S_{2d} : \exists y \in N \textit{ with } \pi(y) = x\}|}{(2d)!} = n/u.$$

Proof. The left hand side of this equation is the number of permutations you can exclude following a negative query x, the right hand side is the trivial success rate. This holds because for all $\forall x, y \in U$, $|\{\pi \in S_{2d} : \pi(y) = x\}| = (d!)^2$, which divided by $(2d!)$ gives exactly $1/u$, and if x and x' are distinct then $\{\pi \in S_{2d} : \pi(y) = x\}$ and $\{\pi \in S_{2d} : \pi(y) = x'\}$ are disjoint, because the permutations are bijections. □

We show that a support enumeration algorithm that only looks at support profiles where player 1 uses full support and player 2 uses a support of size d will require an exponential expected number of guesses to find an NE. (So a general support enumeration algorithm

will do worse in expectation.) Index the elements of $U \setminus N$ by i, with $1 \leq i \leq u - n$. Assume that the elements of U are enumerated in some order. Let W_i be the event that the ith support profile comes before all members of N in the enumeration order of U. Let $\mathbb{1}_{W_i}$ be an indicator function, that is, a 0-1 random variable, that takes the value 1 if the ith support profile comes before all members of N in the enumeration order of U. Let $W = \sum_{i=1}^{u-n} \mathbb{1}_{W_i}$, which is a random variable equal to the number of supports checked before the first equilibrium is found. Then, using the linearity of expectation, we have

$$\mathbb{E}W = \mathbb{E}(\sum_{i=1}^{u-n} \mathbb{1}_{W_i}) = \sum_{i=1}^{u-n} \mathbb{E}(\mathbb{1}_{W_i}) = \sum_{i=1}^{u-n} \frac{1}{n+1} = \frac{u-n}{n+1}.$$

So the expected number of guesses to find a sample NE is

$$\frac{u-n}{n+1} + 1,$$

which is exponential in d since u/n is exponential. The variance of W is about $(\mathbb{E}W)^2$.

5 Vertex enumeration

In this section we consider using vertex enumeration of the best response polytopes to find a sample equilibrium in our games. Unlike support enumeration, vertex enumeration is not so easy to analyse. Here we consider a general method for NASH; in the next section we consider a method specialized to the present construction.

Suppose we have a general $m \times n$ bimatrix game with best response polytopes $P \in \mathbb{R}^m$ and $Q \in \mathbb{R}^n$ (see [16]). One approach for NASH is as follows[3]: Enumerate the vertices of one of the best response polytopes, say P, one by one. If the game is not square it is better to enumerate the polytope in lower dimension as this will have fewer vertices in general. For each vertex $v \in P$, there will be m of the $n + m$ inequalities that describe P that are binding. For simplicity we only consider nondegenerate games, in which case the remaining n inequalities will have positive slacks. These n inequalities correspond to pure strategies that are either played with positive probability or that are not best responses. In equilibrium these must be best responses or not played, respectively, for the other player, which means that the corresponding n inequalities in Q must be binding. Thus, we consider Q_v, which is obtained from Q by turning those n inequalities to equalities. We enumerate the vertices of Q_v and if this polytope is nonempty, its unique point, say w, corresponds to an equilibrium strategy, which together with v, corresponds to an NE, when v and w are normalized, so that they are probabilities. If Q_v is empty we continue the enumeration of P.

[3]Enumerating all vertices of either best response polytope cannot be efficient for NASH as the polytopes may have an exponential number of vertices in the dimension, as indeed the dual cyclic polytopes used in the present construction do.

The polytopes P and Q in the present construction have

$$6\binom{5d/2-1}{d/2-1} \text{ and } 3\binom{2d-1}{d-1}$$

vertices, respectively. These numbers are, according to Stirling's formula, of the order $\Theta(3.49\ldots^d/\sqrt{d})$ and $\Theta(4^d/\sqrt{d})$ respectively. To find an NE of an instance of the present construction the vertex enumeration must reach a vertex of P or Q that corresponds to full support for player 1. Such a vertex will be represented by a bitstring of the form $0^d x \in P$ or $1^d y \in Q$, where x and y are bitstrings of length $2d$ with d ones. If this fact is known then one could enumerate only vertices of Q of this form, as we can fix the first d inequalities in the description of Q as equalities. (These d binding inequalities say that all player 1's pure strategies are best responses.) This yields a new polytope Q', which is a d dimensional face of Q, and has

$$4\binom{3d/2-1}{d/2-1}$$

vertices. This number is of the order $\Theta(2.598\ldots^d/\sqrt{d})$. Enumerating Q' would clearly be preferable to enumerating Q for the present construction, but we can actually do better, if the construction is known. In the next section we describe a specialized vertex enumeration method to find an NE in no more than $2d+1$ steps. The number of equilibria is of the order $\Theta(2.414\ldots^d/\sqrt{d})$, therefore, for P, Q, or Q', the ratio of equilibria to vertices is exponentially small.

A rigorous analysis of the efficiency of vertex enumeration on these games will depend on the actual enumeration method used. No such analysis is attempted here. It is worth noting that any such method must use a rule to determine a unique variable to use in each step, in order that no vertices are overlooked. One possible enumeration method is Avis and Fukuda's reverse search method, *lrs* [1]. For lrs, a rule is required (e.g. Bland's rule) to uniquely determine which variable will "enter the basis" at each step. An analogous rule will be needed for any enumeration method, and will depend on the indexing of the variables (order of the columns) in the polytope description. For this reason, the random permutation of columns, used to produce an instance $(\overline{A}, \overline{B})$ of the present construction from (A, B), will affect the order in which vertices are enumerated (by any method). However, the permutation will not make the order of enumeration of vertices random. (If this were so, then vertex enumeration would be exponential in expectation by the same reasoning used in Section 4 for support enumeration.) Yet, it is conceivable that the expected number of vertices that must be enumerated before an NE is found is still exponential, although this needs further study.

6 Quickly discovering a sample NE by pivoting

In this section we describe a method, using pivoting, to find a sample equilibrium of $(\overline{A}, \overline{B})$ in at most $2d + 1$ steps. It only works for the present construction and does not provide a general method for NASH. Suppose that a game $(\overline{A}, \overline{B})$ is constructed as in Section 2. We assume that the game is known, but that (A, B), π_d, and π_{2d} are not. The method we describe will find one of two equilibria, either corresponding to supports of player 2 equal to $\pi_{2d}(\{1, \ldots, d\})$ or $\pi_{2d}(\{2d + 1, \ldots, 3d\})$. The method does not distinguish between these two equilibria.

The best response polytopes are

$$\overline{P} = \{x \in \mathbb{R}^d \mid x \geq \mathbf{0},\ \overline{B}^\top x \leq \mathbf{1}\}, \tag{3}$$

$$\overline{Q} = \{y \in \mathbb{R}^{2d} \mid \overline{A}y \leq \mathbf{1},\ y \geq \mathbf{0}\}. \tag{4}$$

These are identical to the polytopes

$$P = \{x \in \mathbb{R}^d \mid x \geq \mathbf{0},\ B^\top x \leq \mathbf{1}\}, \tag{5}$$

$$Q = \{y \in \mathbb{R}^{2d} \mid Ay \leq \mathbf{1},\ y \geq \mathbf{0}\} \tag{6}$$

except that the order of the first d inequalities have been permuted by π_d and the order of the second $2d$ inequalities by π_{2d}. The origin is a vertex of $\overline{Q}$, and can be represented as the bitstring $0^d 1^{2d}$, where, as usual, the ith position in this bitstring corresponds to the ith inequality in (4). Because of the permutation of the rows and columns A and B, the ordering of the inequalities in (3) and (4) will not in general be consistent with the Gale evenness condition. A *facet* of P or Q is defined by a single inequality turned into an equality. Let the facets of P and Q be labeled $1, \ldots, 3d$ according to the order of inequalities in (3) and (4) respectively. The best response polytopes P and Q are *simple*, that is, each vertex is defined by d and $2d$ facets respectively. There are $2d$ possible pivots away from the origin. Half of these pivots lead onto the facet corresponding to the first inequality in (5). The other d pivots will take us onto the facet corresponding to inequality d in (5). These will appear as the $\pi(1)$th and $\pi(d)$th inequalities in 4, respectively. The figure that follows illustrates these $2d$ pivots for the case $d = 4$.

inequality of Q	1	2	3	4	5	6	7	8	9	10	11	12
inequality of $\overline{Q}$	$\pi(1)$			$\pi(4)$								
$0 \in \overline{Q}$	0	0	0	0	1	1	1	1	1	1	1	1
	1	0	0	0	0	1	1	1	1	1	1	1
	0	0	0	1	1	0	1	1	1	1	1	1
	1	0	0	0	1	1	0	1	1	1	1	1
	0	0	0	1	1	1	1	0	1	1	1	1
	1	0	0	0	1	1	1	1	0	1	1	1
	0	0	0	1	1	1	1	1	1	0	1	1
	1	0	0	0	1	1	1	1	1	1	0	1
	0	0	0	1	1	1	1	1	1	1	1	0

First we try out the $2d$ pivots from the origin, one by one, using matrix operations. We stop as soon as two distinct facets are reached (i.e. two different inequalities have become an equality for different pivots). This can take no more than $d+1$ pivots. We can not distinguish between inequalities $\pi(1)$ and $\pi(d)$, which is why this method might find either one of two equilibria. Next we pivot in the other best response polytope, $\overline{P}$. The origin is a vertex of $\overline{P}$ and corresponds to the bitstring 1^d0^{2d}. We choose either one of the two inequalities we know to correspond to $\pi(1)$ or $\pi(d)$, calling this choice of inequality a, and the other inequality, which we didn't choose, b ($a, b \in [3d]$). We pivot away from the origin in $\overline{P}$ by leaving a. We reach a new vertex v_1, picking up a new facet, say f_1. We pivot away from v_1 by leaving b, thereby hitting facet f_2 and reaching a new vertex v_2. Then we pivot way from vertex v_2 by leaving facet f_1. We proceed in this manner, pivoting away from vertex v_i by leaving facet f_{i-1}, which was reached in the previous pivot, thereby hitting facet f_{i+1} and reaching vertex v_{i+1}. Depending on our choice of a each facet f_i either corresponds to inequality $\pi(d+i)$ or to inequality $\pi(3d-i+1)$ of (3). When we reach v_d we stop. Then an equilibrium support of player 2 is given by the strategies corresponding to inequalities $\{f_i : i = 1, \ldots d\}$ in (3). Thus the equilibrium support profile is given by all player 1's pure strategies and columns $\{f_i - d : i = 1, \ldots d\}$ for player 2. We illustrate the case $d = 4$ and $a = \pi(1)$ in the following table.

inequality of P	1	2	3	4	5	6	7	8	9	10	11	12
inequality of $\overline{P}$	$\pi(1)$			$\pi(4)$								
$0 \in \overline{P}$	1	1	1	1	0	0	0	0	0	0	0	0
	0	1	1	1	1	0	0	0	0	0	0	0
	0	1	1	0	1	1	0	0	0	0	0	0
	0	1	1	0	0	1	1	0	0	0	0	0
	0	1	1	0	0	0	1	1	0	0	0	0
equilibrium support	0	0	0	0	1	1	1	1	0	0	0	0

7 Conclusions and open questions

In this paper we have presented challenge instances for the problem NASH, of finding one NE of a bimatrix game. The games constructed are hard to solve for both the Lemke–Howson algorithm and support enumeration. The first part of the construction yields games whose equilibrium support profiles have a simple structure. This structure is hidden using random permutations of the payoff matrices. This is an effective method of hiding the support structure of equilibria from support enumeration, even in a general setting. Even if the construction is known, support enumeration is still not efficient. On the other hand, vertex enumeration is more powerful than support enumeration and if the construction is known a sample NE can be found quickly this way. The method used is not general though. Analysis of the efficiency of general vertex enumeration methods for these games is not straightforward.

It would be preferable if challenge instances were "hard to solve", even if the construction is known, but this is a lot to ask. In this case the construction was possible because the combinatorial structure of dual cyclic polytopes is completely known. It is this fact that allows a sample NE to be found quickly when the construction is known. It seems unlikely that games that defeat the LH algorithm and support enumeration can be constructed without using polytopes whose combinatorial structure then makes the games open to attack by specialized methods, if the construction is known.

The games we construct are proposed as challenge instances for NASH, and we feel that a general method that solves these games may shed valuable light on the complexity of NASH. Several directions of further work deserve attention: analyzing general vertex enumeration methods for these games; testing Lemke's algorithm, a variant of the LH algorithm, on these games [17]; analyzing randomized algorithms for NASH.

Acknowledgements

The author thanks Graham Brightwell, Nicholas Georgiou, and Bernhard von Stengel for helpful comments and discussions.

References

[1] Avis, D., and K. Fukuda (1992), A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry* **8**, 295–313.

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

[3] Grünbaum, B. (2003), *Convex Polytopes, 2nd ed.* Springer, New York.

[4] Lemke, C. E., and J. T. Howson, Jr. (1964), Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics* **12**, 413–423.

[5] Lemke, C. E. (1965), Bimatrix equilibrium points and mathematical programming. *Management Science* **11**, 681–689.

[6] Leyton-Brown, K., E. Nudelman, J. Wortman and Y. Shoham (2004), Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. In: *Proc. 3rd AAMAS*.

[7] Megiddo, N. (1986), On the expected number of linear complementarity cones intersected by random and semi-random rays. *Mathematical Programming* **35**, 225–235.

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

[9] Nash, J. F. (1951), Non-cooperative games. *Annals of Mathematics* **54**, 286–295.

[10] Papadimitriou, C. H. (2001), Algorithms, games, and the internet. In: *Proc. 33rd STOC*, 749–753.

[11] Porter, R. W., E. Nudelman, and Y. Shoham (2004), Simple Search Methods for Finding a Nash Equilibrium. In: *Proc. 19th AAAI*.

[12] Rinott, Y., and M. Scarsini (2000), On the Number of Pure Strategy Nash equilibria in Random Games. *Games and Economic Behaviour* **33**, 274–293.

[13] Savani, R., and B. von Stengel (2004) Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game. *CDAM Research Report LSE-CDAM-2004-03*, London School of Economics. [Extended Abstract to appear in *Proc. 45th FOCS* (2004).]

[14] Shapley, L. S. (1974), A note on the Lemke–Howson algorithm. *Mathematical Programming Study* **1***: Pivoting and Extensions*, 175–189.

[15] von Stengel, B. (1999), New maximal numbers of equilibria in bimatrix games. *Discrete and Computational Geometry* **21**, 557–568.

[16] von Stengel, B. (2002), Computing equilibria for two-person games. Chapter 45, *Handbook of Game Theory, Vol. 3*, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, 1723–1759.

[17] von Stengel, B., A. H. van den Elzen, and A. J. J. Talman (2002), Computing normal form perfect equilibria for extensive two-person games. *Econometrica* **70**, 693–715.

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