On the Complexity of Two-Player Win-Lose Games

Tim Abbott, Daniel Kane, Paul Valiant*

April 7, 2005

Abstract

The efficient computation of Nash equilibria is one of the most formidable computational-complexity challenges of today. The problem remains open for two-player games.

We show that the complexity of two-player Nash equilibria is unchanged when all outcomes are restricted to be 0 or 1. That is, win-or-lose games are as complex as the general case for two-player games

1 Game Theory

Game theory asks the question: given a set of players playing a certain game, what happens? Computational game theory asks the question: given a representation of a game and some fixed criteria for reasonable play, how may we efficiently compute properties of the possible outcomes?

Needless to say, there are many possible ways to define a game, and many more ways to efficiently represent these games. Since the computational complexity of an algorithm is defined as a function of the length of its input representation, different game representations may have significantly different algorithmic consequences. Much work is being done to investigate how to take advantage of some of the more exotic representations of games (see [PR, KM] and the references therein). Nevertheless, for two player games, computational game theorists almost exclusively work with the representation known as a *bimatrix game* which we define as follows.

Definition 1 *A bimatrix game is a game representation that consists of a matrix of pairs of numbers (or equivalently a pair of identically sized real matrices.) The game has two players, known as the* row *and* column *players respectively. The matrix is interpreted to represent the following interaction: the row and column players simultaneously pick a row and column respectively of the matrix; these choices specify an entry—a pair—at the intersection of this row and column, and the row and column players receive* payoffs *proportional respectively, to the first and second entries of the pair.*

In this model, a *strategy* for the row or column player consists of a distribution on the rows or columns respectively, and is represented as a vector r or c. When a strategy puts all its weight on a single row or column, it is called a *pure strategy.* To motivate the definition of a *Nash equilibrium*, we define the notion of a *best response.* Given a strategy r for the row player, we may ask which strategies c the column player might pick to give her the maximal payoff. Such a strategy c is said to be a *best response* to the strategy r. Game theorists model "reasonable play" in a bimatrix game with the following criterion:

Definition 2 *A pair of strategies r, c is said to be a* Nash equilibrium *if r is a best response to c and c is simultaneously a best response to r.*

2 Nash equilibria

A fundamental property of Nash equilibria is that *they always exist.* Indeed, it is far from obvious that this should be the case—Nash equilibria for constant-sum two-player games were first shown to exist by von Neumann as a consequence of Brouwer's fixed point theorem. This result was later generalized by Nash to multi-player games using the more general Kakutani fixed point theorem.

A purely combinatorial existence proof for Nash equilibria in two-player games was found by Lemke

*MIT Computer Science and Artificial Intelligence Laboratory, The Stata Center, Building 32, 32 Vassar Street, Cambridge, MA 02139. Email: pvaliant@mit.edu.

and Howson[LH] that has the additional advantage of being *constructive*. Unfortunately, the Lemke-Howson algorithm has exponential worst-case running time[SS].

An alternate algorithm for finding Nash equilibria for two-player games is suggested by the following observation. It turns out that if we know the *support* of the strategies in a Nash equilibria, namely the set of rows and columns that are played with positive probability, we can reconstruct the set of Nash equilibria with that support by solving a linear program. This suggests the *support enumeration* algorithm, wherein we nondeterministically guess supports and check their feasibility. This algorithm has the important consequence of placing the Nash equilibria search problem in the complexity class FNP, the search problem version of NP. This algorithm also has the consequence that if the payoffs of the game are rational, then every support set that has a Nash equilibrium has a Nash equilibrium with *rational* probabilities.

THE DIFFICULTY OF THE NASH PROBLEM. It is natural to ask whether the problem of finding a Nash equilibrium is in fact in P, the class of problems with polynomial-time algorithms. Quite recently there have been significant results on the complexity of several related problems, which have been shown to be NP- or ♯-P hard [GZ, CS1]. Specifically, counting the number of Nash equilibria is ♯-P hard, while determining if there exist Nash equilibria with certain properties—such as having specific payoffs or having specific strategies in their support—is NP-complete. However, the original problem of finding a single Nash equilibrium remains open and, as Christos Papidimitriou is often quoted as saying, "Together with factoring, the complexity of finding a Nash equilibrium is in my opinion the most important concrete open question on the boundary of P today" [Pap].

SOURCE OF COMPLEXITY FOR THE NASH PROBLEM. There are many aspects of games that might make the Nash problem hard to solve. Specifically, considering multi-player games as multi-dimensional arrays of numbers, it is natural to ask which properties of these arrays make finding Nash equilibria hard: is it the number of dimensions of the array, the number of options available to each player, or the complexity of the individual numbers involved?

The first question remains unresolved, as the problem is wide open even for two-player games. We consider two-player games exclusively for the remainder of this paper. The second question has a negative answer, as there exist fixed-parameter tractable algorithms with parameter the size of the strategy space available to one player of a two-player game. The third question, asking whether having complicated payoffs makes the Nash problem hard, is the subject of this paper. We answer this question in the negative. The first results of this kind were shown in [CS2]: determining whether there is more than one Nash equilibrium is NP-complete even in a $\{0, 1\}$-game, and determining if there exists a Nash equilibrium with 0-payoff for one player is NP-complete for $\{0, 1\}$-games. They raised the question as to whether $\{0, 1\}$-games are as *hard* as general games.

OUR CONTRIBUTION. We give a strong positive answer to the above question, exhibiting a specific mapping from rational-payoff bimatrix games into $\{0, 1\}$-payoff bimatrix games that preserves the Nash equilibria in an easily recoverable form. We make this statement more precise in the next section by introducing the notion of a *Nash homomorphism*.

3 Nash homomorphisms

Our general strategy here is to start with a general two-player game, and modify it through a sequence of essentially similar games until we reach an equivalent game with entries entirely in $\{0, 1\}$. Specifically, we relate the original game to the $\{0, 1\}$-game via a sequence of *Nash homomorphisms*, defined as follows:

Definition 3 *A* Nash homomorphism *is a map h, from a set of two-player games into a set of two-player games, such that there exists a polynomial-time function f that, when given a game $B = h(A)$, and a Nash equilibrium of B, returns a Nash equilibrium of A, and further, the map f_B is* surjective *onto the set of Nash equilibria of A. (Here f_B is the map f with the first argument fixed to B.)*

Thus if we relate the original game to the $\{0,1\}$-game via a polynomial sequence of Nash homomorphisms $\{h_i\}$, then the reverse functions $\{f_i\}$ will compose into a polynomial-time computable function. In general, these sequences will provide us with a one-query Cook reduction from NASH to what we dub $NASH_{\{0,1\}}$, the problem of finding a Nash equilibrium in a $\{0,1\}$-game.

The final homomorphism we use will have the effect of translating payoff entries into binary. We begin, however, with a few more fundamental examples of Nash homomorphisms that will be useful in manipulating the game to the point where we may apply this translation homomorphism.

1 The identity homomorphism $h(A) = A$ is clearly a Nash homomorphism, since f_B may be taken to be the identity.

2 The shift homomorphism that takes a game, and shifts the row player's payoffs by an arbitrary additive constant: note that shifting these payoffs does not change the row player's relative preferences at all; so Nash equilibria of the original game will be Nash equilibria in the modified game, and we may thus take f_B to be the identity again.

3 The scale homomorphism that takes a game, and scales the row player's payoffs by a *positive* multiplicative constant: as above, this does not modify the Nash equilibria.

4 The player-swapping homomorphism: if h swaps the roles of the two players, by taking the matrix of the game, transposing it, and swapping the order of each pair of payoffs, then the Nash equilibria of the modified game will just be the Nash equilibria of the original game with the players swapped.

We note that these homomorphisms already give us significant power to transform games into $\{0,1\}$-games. For example, suppose we have the game

1,3	2,-2
1,-2	1,3

.

While this does not appear similar to a $\{0,1\}$-game, we can map it into one as follows: first subtract 1 from each of the row player's payoffs, making her payoffs $\{0,1\}$; next apply the player-swapping homomorphism; then add 2 to each of the payoffs of what is now the row player, and divide these new payoffs by 5. This produces the following game,

1,0	0,0
0,1	1,0

,

whose Nash equilibria are identical to those of the original game with the players swapped.

The above homomorphisms, however, have the weakness that they modify all the entries of the matrix at once, and thus may not be fine-grained enough to transform more intricate games. We note immediately, however, that the scale and shift homomorphisms may be modified to work on the row player's payoffs a column at a time—for some reason it is much easier to work with columns of the row player's payoffs instead of rows. We thus have the following:

2' The column-shift homomorphism that takes a game, and shifts the row player's payoffs in a single row by an arbitrary constant: we note that both the row and column players' notions of *best response* remain unaffected by this shift, so Nash equilibria are preserved under this transformation.

3' The column-scale homomorphism that takes a game, and scales the row player's payoffs in a single column by an arbitrary positive constant α: it turns out that if we take a Nash equilibrium of the original game, and scale the column player's strategy *in this column* by $1/\alpha$, then re-scale the column player's strategy to have unit sum, we will have a Nash equilibrium of the new game. This is proved in the appendix.

We note at this point, that all the homomorphisms introduced so far are linear, and in addition have the property that the map they induce on Nash equilibria, f_B, is a bijection. These properties make the above homomorphisms almost trivial to verify, but also limit their usefulness.

Indeed, if all Nash homomorphisms satisfied these properties, then our program would have serious impediments. Using only linear transformations, we could never use combinatorial tricks like converting integers to their binary representation.

Moreover, the fact that f_B is a bijection means that whatever game appears at the end of our sequence of homomorphisms must have *identical* Nash equilibrium structure. We might call such maps *Nash isomorphisms*. To see why this is a problem, we note that games may be classified as being either degenerate, or non-degenerate(see [Ste]), in analogy with the corresponding definition for matrices. The Nash equilibria set of a non-degenerate game has certain characteristic properties, for example, having a cardinality that is finite and odd, and we could not expect a general *Nash isomorphism* that maps non-degenerate games to degenerate games. The catch here, is that *all* (non-trivial) $\{0,1\}$-games are degenerate!

We therefore conclude that we must find some non-linear non-isometric Nash homomorphisms. It turns out that two are enough for our task.

4 The "split-glue" homomorphism

This next homomorphism is motivated by the following observation:

> Since the row player's payoffs seem much easier to modify in units of whole columns, if we wish to do anything really drastic to the row player's payoffs, we should
>
> **a.** work in columns, and
>
> **b.** make sure to get the column player's payoffs well away from those columns!

This suggests the need for a "splitting" homomorphism, that takes a column and splits it into two columns, one containing the row player's payoffs from the original column, the other containing the corresponding payoffs of the column player, with the remaining entries of these two columns set to 0, or some other number strictly less than all the other payoffs in the matrix. An example of such a splitting map would be as follows:

2,3	4,5
4,1	1,2

$\rightarrow$

2,3	4,0	0,5
4,1	1,0	0,2

Unfortunately, this splitting operation is not a homomorphism, for the simple reason that the column with 0 payoff for the column player—in this case the second column—will never be played, and thus whatever structure is in the row player's payoffs for this column will be effectively ignored.

We thus need to find some way to "glue" these columns back together to give this "split" game more of the semantics of the original game. The solution is to add another row to the game, specifically, a row consisting of specially designed entries in these two columns that we aim to "glue". This concept is formalized in the following definition.

Definition 4 *We define the* split-glue *map as follows. Given a game defined by a pair of matrices (R, C), with R, C containing the row and column players' payoffs respectively, and a specific column i, the split-glue map transforms (R, C) into a new game (R', C') with the following steps:*

1. *Make sure that all the payoffs in R and C are non-negative; otherwise, abort.*
2. *Split the column i into columns i' and i'', where column i' in matrix R' receives column i from matrix R and column i'' in matrix C' receives column i from matrix C, filling the empty columns i'' in matrix R' and i' in matrix C' with zeros.*
3. *Check to see whether all the entries in the i'th column of matrix R' are strictly greater than some fixed constant ϵ, otherwise add a constant to all the entries in this column to make this so.*
4. *Add a new row k to matrices R' and C', and fill it with zeros except at the intersection of this row with the columns i' and i''.*
5. *Make $R'_{k,i'} = \epsilon$, let $C'_{k,i'} \stackrel{\text{def}}{=} \delta$ and $R'_{k,i''}$ be arbitrary strictly-positive numbers, and let $C'_{k,i''}$ remain 0.*

Recalling our example from above, if we let $\epsilon = \frac{1}{2}$, the split-glue map would produce the following:

2,3	4,5
4,1	1,2

$\rightarrow$

0,0	$\frac{1}{2}$,1	1,0
2,3	4,0	0,5
4,1	1,0	0,2

We claim that this map in fact preserves the structure of Nash equilibria.

Claim 5 *The* split-glue *map is a Nash homomorphism. The reverse map f may be defined as follows: given an equilibrium of the game (R', C'), discard the probabilities of playing row k or column i'', treat the weight of the remaining strategies in (R', C') as weights of the corresponding strategies in (R, C), interpreting strategy i' as representing strategy i in the original game, and then re-scale these probability distributions to have weight 1.*

The proof of this claim is somewhat technical, and we deferit to the appendix.

At first glance, the split-glue homomorphism might not seem such a powerful tool. However, as hinted above, the fact that we may now isolate a column of row player payoffs means that we can design much more intricate homomorphisms that take advantage of the fact that the column-player's payoffs no longer interfere in this column.

To motivate our general scheme for translating columns into binary, we first present a simple example of how to use the game "rock-paper-scissors" to simulate fractional payoffs in a 0-1 game.

5 The subgame substitution homomorphism

We begin by noting that, in our model of games, getting a payoff of 1 a third of the time is the same as getting a payoff of a third. Thus if we want to give, for example, the row player an effective payoff of a third, we need to find a randomizer that takes any of three actions with probability one third, and reward the row player in one out of the three cases. In the game theory context the natural choice for this randomizer is, of course, the other player.

The simplest example of a game like this is the children's game "rock paper scissors", in which both players simultaneously commit to a move of either "rock", "paper", or "scissors", and the winner is determined according to the rule that paper beats rock, rock beats scissors, and scissors beats paper. In our notation, this game is represented as the following game, which conveniently happens to be a $\{0, 1\}$-game.

0,0	0,1	1,0
1,0	0,0	0,1
0,1	1,0	0,0

As promised, this game has a Nash equilibrium where both players play each strategy with probability one third, and receive expected payoff one third. This game also has the significant property that this is the *unique* Nash equilibrium. Thus, any time we see an instance of the game "rock paper scissors" being played in our model, we know the exact distribution of the three strategies for each player.

This observation leads to the following question:

> how can we insert the "rock paper scissors" game into a larger game so as to preserve this predictability property?

We note the following crucial fact, the proof of which consists of a straightforward application of the definition of Nash equilibria.

Claim 6 *Suppose a game G appears embedded in a larger game H. Specifically, if H has row set R and column set C, let the game G appears at the intersection of rows $r \subset R$ and columns $c \subset C$, with 0 payoff for the column player at the intersection of rows r and columns $C \backslash c$, and 0 payoff for the row player at the intersection of columns c and rows $R - r$. Then in any Nash equilibrium of H where some row of r and some column of c is played with positive probability, the restriction of this Nash equilibrium to rows r and columns c will be a scaled version of a Nash equilibrium of G.*

Further, applying the column shift homomorphism, we find that we can relax the restriction that the row player payoffs at the intersection of rows r and columns $C \backslash c$ be 0 to the condition that they instead be column-wise uniform, with a corresponding relaxation applying to the intersection of columns c and rows $R \backslash r$.

This suggests that if we have a $\{0, 1\}$-game S such as "rock paper scissors" that has a unique equilibrium with payoffs (α, β), we can take a game that contains an entry (α, β), and replace it with S as a subgame. This is formalized as the following homomorphism:

Definition 7 *Suppose for some fixed (α, β), there is an $m \times n$ game S that has an equilibrium with payoffs (α, β), and furthermore, that this equilibrium is unique. Then define the "subgame* *substitution" map as follows: for games G that have payoffs (α, β) at some the intersection of some specified row r and column c, map G to game G' by replacing row r with m copies of itself, replacing column c with n copies of itself, and placing the subgame S at the intersection of these m rows and n columns.*

We claim that this map is in fact a homomorphism.

Claim 8 *The subgame substitution map is a homomorphism.*

Proof: Denoting the m copies of r as rows r', and the n copies of c as columns c', we note that since the matrix is row-wise uniform on the intersection of rows $R \backslash r'$ with columns c', and column-wise uniform on the intersection of columns $C \backslash c'$ with rows r', the new game G' satisfies all the requirements of claim 6.

Thus, in every Nash equilibrium of G' where some row of r' and some column of c' is sometimes played, the weights of the rows in r' and the columns in c' must be proportional to those of the Nash equilibrium of S. Further, if the total weight in the rows r' is r'_Σ and the total weight in the columns c' is c'_Σ, then the payoffs to the row and column players from the subgame S will be exactly $r'_\Sigma c'_\Sigma(\alpha, \beta)$ respectively, which is exactly the payoff in game G of playing row r with probability r'_Σ and column c with probability c'_Σ.

We now observe that a similar thing happens to all the other entries in rows r' and columns c'. Specifically, consider the intersection of the rows r' with some column $j \in C \backslash c'$. Since all the row player payoffs in this intersection are identical, and all the column player payoffs are also identical, the total payoffs represented by this intersection are exactly the product of the summed weight r'_Σ with the weight in this column, with the row and column payoffs respectively at the intersection of row r and column j in the original game G.

A straightforward application of the definition of Nash equilibria reveals that we may thus map Nash equilibria of G' onto Nash equilibria of G by merging the weights of rows in r' and columns in c' into their sums r'_Σ and c'_Σ respectively. Thus the subgame substitution map is indeed a Nash homomorphism. ▮

The naive approach from here is to take a game and try replacing all of its rational entries with $\{0, 1\}$-subgames. Aside from the question of how to find $\{0, 1\}$ games with arbitrary rational payoffs, this approach also has the drawback that whenever we "fix" one bad entry by replacing it with an $m \times n$ subgame, we multiply the number of other "bad" entries in this element's row and column by m and n respectively; such a process would never end. We avoid this problem by instead fixing *all* the entries of a column *at once*. To do this, we need the added structure of the split-glue homomorphism.

6 Combining homomorphisms

Recall that the split-glue homomorphism did not strictly separate out the row player component of a column from the column player component—the column we denoted by i' is *not* devoid of column player payoffs, but in fact contains a single positive entry in the added row k. However, in the words of our favorite tech support guy, *it's not a bug ... it's a feature.* The insight here is that since the column i' contains a *single* profitable payoff for the column player, if this column is ever played it must be because the row player chooses to play row k. Thus every time we have to worry about column i', we can offset our lack of knowledge of this column with a guarantee about the behavior of a row.

It turns out that this property makes the entry (ϵ, δ) at the intersection of row k and column i' is an ideal candidate for the subgame substitution map.

We note that one of the weaknesses of the subgame substitution map is that is the uniformity requirement for those payoffs in the same rows or columns as the substituted game. This weakness in fact originates from the corresponding weakness in claim 6—namely that we proved that in any Nash equilibrium of the modified game G', the weights of the rows and columns r' and c' will be proportional to those of a Nash equilibrium of S *only provided that either both r' and c' contain nonzero weights, or neither does.* Thus it might occur in a Nash equilibrium of G' that, while the weights of r' are uniformly 0, the weights on columns in c' are arbitrary, and not in the ratio specified by the unique Nash equilibrium of the game S.

However, as mentioned above, the split-glue homomorphism has the property that whenever the weight in column i' is nonzero, the weight in row k must also be strictly positive. Thus, if we apply the subgame substitution homomorphism to this element, we find that any time there are nonzero weights in the columns c', there must be nonzero weights in the rows of r', from which we conclude that the weights on c' must be in the ratio prescribed by the Nash equilibrium of S. These results are summarized in the following lemma:

Lemma 9 *Suppose we have a game G, whose payoffs are all at least 0, and which additionally has the property that every row of G contains a strictly positive payoff for the column player. First apply the split-glue homomorphism to the column i to produce a game G', with payoffs (ϵ, δ) at the intersection of the new row k and new column i'. Then, assuming we have a game S that has a unique Nash equilibrium, with payoffs to the row and column players of (ϵ, δ) respectively, apply the subgame substitution homomorphism to this element (ϵ, δ), expanding column i' into columns c', and row k into rows r' ot produce the game G''. Then:*

> *In any Nash equilibrium of the new game G'', the weights on columns c' will be proportional to those in the Nash equilibrium of S.*

Further, this property will be preserved even if we change the entries of the row player's payoff in any rows other than those in r'.

We have already outlined this proof in the above text. It appears below more formally.

Proof: We first note that the property that every row of G contains a strictly positive payoff for the column player is preserved through the operation of the split-glue and subgame substitution homomorphisms—provided of course that the subgame S possesses this property. Further, we note that this property implies that any Nash equilibrium of the game G'' has strictly positive payoff for the column player, since no matter what strategy the row player picks, there will be a column with positive payoff for the column player.

We observe that if the weights in columns c' are uniformly 0, then they are trivially proportional to those of the Nash equilibrium of S and there is nothing to prove. Otherwise, there is a column of c' that is played with positive probability.

From the definition of a Nash equilibrium, the column player will only play *best responses* in a Nash equilibrium. Thus, since the Nash equilibria of G'' give strictly positive payoff to the column player, she will always have an option to play on a column with positive payoff, and will thus always take such an option.

Thus the fact that the column player plays in a column of c' implies that she receives positive payoff for this choice. Since the only positive payoffs in these columns for the column player lie in rows r', the row player must play in one of these rows with positive probability. Thus some row of r' and some column of c' are played with probability.

We now invoke claim 6 to conclude that the weights on columns c' are proportional to those of the Nash equilibrium of S, as desired. ∎

We now introduce our final homomorphism, a surprisingly simple one, but one which, when combined with the above homomorphisms will enable us to transform integer payoffs into binary, and thereby transform rational games into $\{0, 1\}$-games.

7 Translating to $\{0, 1\}$

We observe that if we have a game G and two columns c_1, c_2 such that for *every* Nash equilibrium of G the weights on these columns are in some fixed ratio $a : b$ then we can replace a row player payoff of p_1 in column c_1 with the payoff $p_1 \frac{a}{b}$ in column c_2 and carry all the Nash equilibria into the new game. More generally, we prove the following:

Claim 10 *Suppose we have a game G and a set of columns c with the property that in any Nash equilibrium of G, the columns in c are played with weights proportional to some vector γ. For some row j, denote by $p_{j,c}$ the payoffs for the row player at the intersection of row j and columns c. We claim that if we modify the entries $p_{j,c}$ to any vector $p'_{j,c}$ such that*

$$p_{j,c}\gamma^T = p'_{j,c}\gamma^T,$$

then all the Nash equilibria of G will also be Nash equilibria of the new game. We call such a map a linear reexpression *map.*

Proof: Take any Nash equilibrium of G and consider it in the context of the modified game. By definition, the probabilities of playing the columns c will be in proportion to the vector γ, and thus the transformation does not change the value of any row to the row player. Further, no entries of the column player's payoffs were changed, so the columns have the same value as they did in the Nash equilibrium of G. Thus, from the definition of *best response*, if the row and column player strategies were mutual best responses in G, they must remain so in the modified game. Thus this Nash equilibria of G remains a Nash equilibria under the modification. ▮.

We note that the above map is *not* necessarily a Nash homomorphism, since the modified game could introduce equilibria of its own. However, under the conditions of lemma 9 the linear reexpression map is in fact a homomorphism.

Lemma 11 *Suppose we have a game G and a set of columns c with the property that in any Nash equilibrium of G, the columns in c are played with weights proportional to some vector γ* even when the row player payoffs in some row j are changed. *Then, if we modify the payoffs $p_{j,c}$ at the intersection of row j and columns c to some other payoffs $p'_{j,c}$ such that*

$$p_{j,c}\gamma^T = p'_{j,c}\gamma^T,$$

, then the Nash equilibria of the resulting game will be identical *with those of G.*

Proof: We apply claim 10 to show that every Nash equilibrium of G is a Nash equilibrium of the new game G', and then apply claim 10 to G' to show that every Nash equilibrium of G' is also a Nash equilibrium of G. ▮

This lemma has the following amazing consequence:

Corollary 12 *Given a game G, apply the split-glue homomorphism on some column i followed by the subgame substitution homomorphism on the corresponding column i', for some appropriate subgame S with a unique Nash equilibrium. If the equilibrium strategy for the column player has weights γ, then for any rational payoff in column i' that is expressible as the inner product of a $\{0, 1]\}$ vector with γ, we can represent it in G'* by this $\{0, 1\}$ vector!

We give an example of the application of this corollary. Suppose we start with the—admittedly simple—game

2,3	$\frac{2}{3}$,5
4,1	1,2

.

As illustrated below, we now apply the split-glue homomorphism to the second column, with $(\epsilon, \delta) = (\frac{1}{3}, \frac{1}{3})$, and then apply the subgame substitution homomorphism to replace this entry with the game "rock paper scissors". We now note that the rational entries $\frac{2}{3}$ in the third game may be replaced by the equivalent $\{0, 1\}$-vector $(1, 1, 0)$, since the inner product of $(1, 1, 0)$ with S's Nash equilibrium $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is exactly $\frac{2}{3}$. Explicitly, this sequence of homomorphisms produces:

2,3	$\frac{2}{3}$,5
4,1	1,2

→

0,0	$\frac{1}{3}$, $\frac{1}{3}$	1,0
2,3	$\frac{2}{3}$,0	0,5
4,1	1,0	0,2

→

0,0	0,0	0,1	1,0	1,0
0,0	1,0	0,0	0,1	1,0
0,0	0,1	1,0	0,0	1,0
2,3	$\frac{2}{3}$,0	$\frac{2}{3}$,0	$\frac{2}{3}$,0	0,5
4,1	1,0	1,0	1,0	0,2

→

0,0	0,0	0,1	1,0	1,0
0,0	1,0	0,0	0,1	1,0
0,0	0,1	1,0	0,0	1,0
2,3	1,0	1,0	0,0	0,5
4,1	1,0	1,0	1,0	0,2

Our general strategy now becomes clear.

1. Find a subgame S with a sufficiently expressive equilibrium strategy γ for the column player.
2. Then for each column in a game G, repeat the following:
   (a) Scale and shift the row player's payoffs in this column so that
      i. each payoff is strictly greater than ϵ, the column player's payoff in the equilibrium of S, and
      ii. each shifted payoff is expressible as the inner product of γ with a $\{0, 1\}$-vector.

(b) Apply the split-glue homomorphism to this column, with (ϵ, δ) set to the payoffs in the game S.

(c) Then substitute the game S into this entry.

(d) Finally, replace each entry in these columns with the corresponding $\{0,1\}$-vector guaranteed to exist by step 2(a)ii.

3. Flip the players in this game, and apply the above procedure to what were the rows of the game.

If we can find such an S, then this sequence of homomorphisms will reduce any game to a $\{0,1\}$ game!

8 Sufficiently versatile subgames

We now derive a class of games S that may be used in the above reduction. Recall that we wish to find a game S, with a unique Nash equilibrium, such that the column player's strategy γ is *expressive* in the following way:

> We can represent a wide variety of numbers as the inner product of γ with a $\{0,1\}$-vector.

The natural scheme that satisfies this property is of course the binary representation scheme. If we make elements of γ equal to powers of two, then we can represent numbers as $\{0,1\}$ vectors by just reading off their binary digits.

We note two things: first, that since γ is a probability distribution, its elements must sum to 1; but second, that this restriction does not matter since we are free to scale the columns of our game to match whatever scale S requires.

Thus, we are done if we can find compact games S with unique equilibria such that the column player's strategy γ contains elements proportional to consecutive powers of 2.

In the appendix, we show the following theorem.

Theorem 13 *There exists an efficiently constructible class of games S_j with the following properties:*

1. *S_j is a size $3j \times 3j$ $\{0,1\}$-game,*
2. *Every row of S_j contains a nonzero payoff for the column player (see lemma 9),*
3. *S_j has a unique Nash equilibrium,*
4. *The column player's strategy γ in this equilibrium consists of weights proportional to three copies of $(1, 2, 4, ..., 2^{j-1})$ (the constant of proportionality being $\frac{1}{3(2^j-1)}$ to make these weights sum to 1),*
5. *The row player's payoff in this equilibrium—which we have been denoting as ϵ—is $\frac{2^j}{3(2^j-1)}$.*

We now prove the following theorem:

Theorem 14 *There is a polynomial-time Nash homomorphism from the set of $m \times n$ rational payoff games expressible using k total bits in binary representation into the set of $\{0,1\}$-games of size at most $(3k+1)(m+n)$ by $(3k+1)(m+n)$.*

Proof: Given a game G in this set, we follow the strategy outlined above. The first step is to choose the appropriate S_j. Since k is the total number of binary digits needed to express G, we conservatively take $j = k$, so that we might use k bits to express *each* entry of G.

Then, for each column i containing non-$\{0,1\}$ payoffs, we run the procedure of step 2 above. For the sake of convenience, we denote the constant of proportionality $\frac{1}{3(2^j-1)}$ by α in the following.

We note that, modulo the factor α, the coefficients expressible with S_k are those integers expressible as the sum of integers in the multiset $\{1, 1, 1, 2, 2, 2, ..., 2^{k-1}, 2^{k-1}, 2^{k-1}\}$, namely any integer between 0 and $3 \cdot 2^k - 3$. If we put the row player's payoffs in column i under a common denominator and scale them to be integers, then each of them will be expressible with at most k binary bits. In order to apply the split-glue homomorphism, we additionally must shift these payoffs to be greater than ϵ, the column player's payoff in the Nash equilibrium of S_k. From theorem 13, S_k is just $\alpha 2^k$, from which we conclude that we may scale and shift the elements of this column to be integers between $2^k + 1 and 2 \cdot 2^k$, and thus apply the split-glue homomorphism to this column.

Next, we apply the subgame substitution homomorphism to the entry (ϵ, δ) created by the split-glue homomorphism, and then apply the *linear* *reexpression* homomorphism to express each entry as a $\{0,1\}$-vector.

We have thus transformed G by replacing column i with $3k+1$ columns and adding $3k$ rows in such a way that all the row player's payoffs in these $3k+1$ columns are now in $\{0,1\}$, the payoffs in the added rows are all in $\{0,1\}$, none of the row player's payoffs outside of these columns have been modified, and none of the column player's payoffs have been modified. Furthermore, since this map is a composition of homomorphisms, all the equilibria of G are preserved in the new game.

Repeating this procedure for every column transforms G by way of a Nash homomorphism to a game where *every* payoff of the row player is in $\{0,1\}$.

We now apply the player-swapping homomorphism, and repeat the above procedure for the (at most) n columns containing non-$\{0,1\}$ payoffs. The new game will be entirely $\{0,1\}$.

We note that we have added rows and columns in groups of $3k$ whenever one of the original $n+m$ rows or columns contained non-$\{0,1\}$ payoffs. Thus the size of the final game will be at most $(3k+1)(m+n)$ by $(3k+1)(m+n)$, as desired. ▌

9 Conclusion

We have exhibited a polynomial-time Nash homomorphism from two-player rational-payoff games of k bits to $\{0,1\}$ games of size polynomial in k. This shows that the complexity of finding Nash equilibria of these two classes of games is polynomially related. It may be hoped that $\{0,1\}$-games could offer algorithmic insights into the general Nash problem.

We also pose as an open problem whether or not our results may be extended to apply to the multi-player case.

References

[CS1] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. In *Proc. of IJCAI*, pages 765–771, 2003.

[CS2] Bruno Codenotti, Daniel Stefankovic. On the computational complexity of Nash equilibria for (0,1)-bimatrix games. *Information Processing Letters*, to appear.

[GZ] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1989.

[KM] D. Koller and N. Megiddo. Finding mixed strategies with small supports in extensive games. *International Journal of Game Theory*, 1995.

[LH] C. E. Lemke and J. T. Howson Jr. Equilibrium points of bimatrix games. *Journal of SIAM* **12**, pages 413–423, 1964.

[Pap] C. H. Papadimitriou, Algorithms, Games, and the Internet. *STOC* 2001.

[PR] C. H. Papadimitriou and T. Roughgarden. Computing Equilibria in Multi-Player Games. 2004, unpublished.

[SS] R. Savani and B. von Stengel (2004), Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game. In: Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004), 258-267.

[Ste] B. von Stengel. *Computing equilibria for two-person games.* In *Handbook of Game Theory, Vol. 3*, eds. R. J. Aumann and S. Hart, pages 1723–1759,2002.

APPENDIX

We provide here those proofs omitted in the text.

Lemma 15 *The map that rescales by some positive constant α the row player's payoffs in a specific column is a Nash homomorphism.*

Proof: Suppose we have a game $G=(R,C)$, where the ith column of R is scaled as above to create a game $G'=(R',C)$. Consider a Nash equilibrium (r'^*, c'^*) of G'. Construct a new strategy for the column player c^* as follows: start with c'^*, scale the ith weight by factor α, and then rescale c^* so that is has sum 1. We claim that (r'^*, c^*) is a Nash equilibrium of G.

The proof is a straightforward application of the definition of Nash equilibria. Since the ith column of R' is α times the ith column of R, the vectors Rc'^{*T} and $R'c'^{*T}$ will be equal; thus the fact that r'^* is a best response to c'^* in G' means that it is also a best response to c^* in G.

To show that c^* is a best response to r'^* in G, we note that the fact that c'^* is a best response to

r'^* in G' means that *every* column in the support of c'^* is a best response. Thus we can change the relative ratios of elements of c'^* and maintain this best response property. Thus c^* is a best response to r'^* in G', and, since G and G' have the same payoff matrix R for the row player, c'^* is a best response to r'^* in G.

Thus this scaling map has an associated reverse map that maps Nash equilibria of G' to equilibria of G. Because this scaling map is the inverse of the map that scales column i be $\frac{1}{\alpha}$, this reverse map is also surjective. Hence the column-scaling map is a homomorphism, as desired. ∎

Proof of Claim 5: Let G be a game that is mapped to the G' by applying the split-glue map on column i. As in the definition of the split-glue map, denote by i' and i'' the two columns produced in G', and let k index the new row added to G'. Let (ϵ, δ) be the element in G' at the intersection of row k and column i'.

We first show that any Nash equilibrium (r', c') of G' is mapped to a Nash equilibrium of G under the map f of Claim 5. Recall that f maps (r', c') to (r, c) by throwing out the weights of the kth element of r and the i''th element of c and re-scaling these weights to have sum 1, interpreting the i'th element of c' as corresponding to the ith element of c.

We note that f is a proper map provided this re-scaling does not attempt to re-scale a 0 vector. Suppose that the only element in the support of r' were row k. In this case, the column player would only ever play in column i' as this contains the only nonzero payoff for her in row k. But if the column player only ever plays in column i', then the row player would certainly never play row k since, by the hypothesis of the split-glue homomorphism, row k contains the smallest row player payoff in this column. This contradicts the fact that the row player always plays in row k.

We further note that if column i'' were the only column in the support of c', then the row player would only ever play in row k, since row k contains the only nonzero row player payoff in this column, leading to another contradiction. Thus r' contains a nonzero weight outside row k, and c' contains a nonzero weight outside column i'', and the map f is well defined.

We now prove that r is a *best response* to c in G, half the definition of a Nash equilibrium. Since (r', c') is a Nash equilibrium of G', r' must be a best response to c' in G'. We note that since the column i'' that we threw out has nonzero payoff for the row player only in row k, and we are throwing out row k too, the removal of column i'' from the strategy c' does not affect the best response of a row player that does not have row k as an option. Thus r will be a best response to c in G.

We now show that if c' is a best response to r' in G' then c is a best response to r in G. We note that if we remove column i' from G', the column player's payoffs will be identical to those in G, since the added row k (with column i' removed) contains only 0 payoffs for the column player. Thus if the i'th component of c' is 0, then c will be a best response to r in G, as desired.

We now consider the case where the i'th component of c' is positive. We note that if the column players payoff in the Nash equilibrium (r', c') of G' is 0, then every column of G' has 0 incentive for the column player. Since the column player payoffs of G are a subset of those of G', and removing the strategy k from G' cannot make zero payoffs positive, the column player's incentives in G are 0 when the row player plays r. Thus *any* strategy c is a best response, as desired.

This leaves the case where the i'th component of c' is positive and the column player receives positive payoff in the Nash equilibrium (r', c'). We note that since the i'th column of G' is played with positive probability, the column player must thus be receiving positive payoff for playing in this column. Since the kth entry of this column has the only positive payoff for the column payoff, we conclude that row k must be played in r' with strictly positive probability.

We now prove that in this case column i'' is played with positive probability. Suppose for the sake of contradiction that i'' were never played. Thus the only incentive for the row player to play row k comes from the product of ϵ with the probability of the column player playing column i'. However, by the hypothesis of the split-glue homomorphism, all the other row player payoffs of column i' are strictly greater than ϵ. In this case, row k is never part of any best response to c', a contradiction. Thus i'' is played with positive probability.

We now note that since column i'' is sometimes played, it must be a best response to r'. Since the column player's payoffs of column i'' in G' are identical to those of column i in G, the column i of G is a best response to strategy r. Furthermore, since every other column of G has identical column player payoff to the corresponding column of G' all the columns in the support of c will also be best responses to r. Thus c is a best response to r. This proves that f maps Nash equilibria of G' to Nash equilibria of G.

We now show that this map is surjective, i.e. that given a Nash equilibrium (r, c) of G we can find a preimage (r', c') of it under f. If column i is not in the support of c, then the strategies (r', c') with $r'_k = c'_{i''} = 0$, $r'_{\neq k} = r$ and $c'_{\neq i''} = c$ clearly produce a Nash equilibrium of G' that maps to (r, c) under f.

Otherwise, if $c_i \neq 0$ then construct (r', c') as follows. Let $r'_{\neq k} = r$ and $c'_{\neq i''} = c$. Then set r'_k so that the incentive of the column player to play column i' equals her payoff in the equilibrium (r, c) of G. Note that this is always possible since this payoff will be nonnegative, and the column player payoffs in column i' are positive only in row k. Next set $c''_{i'}$ so that the row player's incentive to play row k equals her payoff in the Nash equilibrium (r, c) of G. Again, note that this is possible since each row play payoff in row k is at least as small as any other payoff in that *column*, with the exception of i''th payoff.

In the above paragraph, we have ignored the restriction that r' and c' must sum to 1. We rescale them now. Note that we have produced a pair of strategies (r', c') such that every row or column that was a best response in G is now a best response in G', and in addition, both row k and column i'' are best responses. Since the support of r' is the just the union of the support of r with possibly the set $\{k\}$, and the support of c' is at most the union of the support of c with $\{i''\}$, the strategies (r', c') will be mutual best responses in G'. Thus every Nash equilibrium of G is the preimage under f of some Nash equilibrium of G'.

This concludes the proof that the split-glue map is a homomorphism.

∎

Proof of Theorem 13: We construct the generator games S_j as follows. Define matrices A, B as

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

For $k = 3j$ define the $k \times k$ matrix S'_j to have the following $j \times j$ block form:

$$S'_j = \begin{pmatrix} A & A & \cdots & A & B \\ A & A & \cdots & B & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ A & B & \cdots & 0 & 0 \\ B & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Explicitly, S'_j has block B on the minor diagonal, block A above, and 0 below. Further, define the game S_j as the pair of matrices $(S'_j, 1 - S'_j)$.

Consider the potential *full-support* Nash equilibria (r, c) of S_j. Note that since r and c are mutual best responses, each of their component *pure strategies* must be a best response to the other player's strategy. For all the column player's choices to be best responses, the *incentives* $r(1 - S'_j)$ must all be optimal, and hence equal. Correspondingly, all the entries of $S'_j c^T$ must be equal. Together with the constraints that r and c must each have total weight 1, we have $2k$ equations and $2k$ unknowns. We show that they have a unique solution.

We show by induction that these equations are satisfied iff both r and c equal

$$v = \frac{1}{3(2^j - 1})(2^{j-1}, 2^{j-1}, 2^{j-1}, ..., 4, 4, 4, 2, 2, 2, 1, 1, 1).$$

Suppose as our induction hypothesis that the first $3i$ entries of c are in the proportions of this vector. Consider the $i + 1$st block row from the bottom of S'_j. These three rows consist of i blocks of A followed by one block B, followed by zeros. As noted above, these three rows, must all have equal incentives for the row player.

Consider the component to this incentive provided by the i blocks of A. By the induction hypothesis, however, the first $3i$ components of

c are arranged in uniform triples, which implies that each of these copies of A produces identical incentive on these three rows. Thus since the total incentive on these rows is equal, the incentives from the B-block must also the same. Writing out these equations, we have

$$c_{3i+1} + c_{3i+2} = c_{3i+1} + c_{3i+3} = c_{3i+2} + c_{3i+3},$$

which implies $c_{3i+1} = c_{3i+2} = c_{3i+3}$. Thus the $i + 1$st block of c is uniform.

To show the ratio of $2 : 1$ between adjacent blocks, we compare the incentive of these rows to the incentive of the subsequent block of rows, which must be the same. These two blocks differ by $A-B$ in the ith column block, and by B in the $i + 1$st column block yielding the condition $(2 - 1)c_{3i} = 2c_{3i+3})$, from which we conclude that the weights in these blocks are indeed in the desired $2 : 1$ ratio.

Clearly, the same argument applies to r. Thus since both x and y must sum to 1, $x = y = v$ is the only Nash equilibrium with full support.

We now note that the game $(S'_j, 1 - S'_j)$ is in fact a constant-sum game, so its Nash equilibria are the solutions to a linear program. This implies that the set of Nash equilibria is convex.

If we suppose for the sake of contradiction that there is another Nash equilibrium in addition to the one $x = y = v$, then all linear combinations of these two equilibria must also be equilibria, and hence by standard topology arguments there are a continuum of full support equilibria, which would contradict the uniqueness argument of the previous paragraph.

Thus $x = y = v$ is the unique equilibrium for the game $(S'_j, 1 - S'_j)$.

Examining the last row of S'_j, we verify that the payoff for the row player is indeed $\frac{2^j}{3(2^j-1}$, as desired. ∎