A Report on r-Nash Reduction

Jadrian Miles
jadrian@cs.brown.edu
For Claire Mathieu's CS295-8 at Brown University

2006-12-11

1 Introduction

The problem of computing Nash equilibria of two-player zero-sum games may be reduced to a polynomial-time solvable linear programming problem. For $r > 2$, the equilibrium strategies of r-player zero-sum games may be irrational, and it is unclear whether non-zero-sum games, even for two players, may be solved quickly. In general, one gets the sense that the complexity of computing Nash equilibria in a general r-player game, henceforth r-Nash, may increase without bound with r.

A series of recent papers has placed increasingly tighter upper bounds on this growth in complexity. Storage space for general games increases exponentially with the number of players, and so these and other works investigated reductions between r-Nash and d-Graphical-Nash, graphical games of degree $\leq d$, in which each player is a vertex and only affects the payoffs of its neighbors in the graph. In [3], Goldberg and Papadimitriou first proved that d-Graphical-Nash and r-Nash for $r \geq 4$ are reducible to 4-Nash, and in [2] followed up by proving that 4-Nash is complete in the complexity class **PPAD** by a reduction of the **PPAD**-complete problem 3-Dimensional Brouwer to 3-Graphical-Nash. Chen and Deng completed the series in [1] by proving, with a refinement of the previous reduction that bypasses graphical games, that 2-Nash is **PPAD**-complete. While **PPAD** is not a richly-described class at this time and therefore completeness within it gives no good general sense of the complexity of 2-Nash, this final result limits the sufficient scope of future work by establishing that r-Nash is equally hard for all r (up to a polynomial-time reduction), and therefore work may concentrate on 2-Nash without loss of generality.

This paper proceeds by first presenting a shallow overview of the concepts and work leading up to Chen and Deng's paper, then demonstrating a sketch of their reduction, and finally making explicit portions of the reductions in [2] and [1].

2 Summary of [1]

2.1 3-Dimensional Brouwer

Brouwer's fixed-point theorem states that any continuous map of a closed k-ball (or any other space homeomorphic to a k-simplex) to itself must have a fixed point. To turn the fixed-point theorem into a computational search problem, consider a function on the unit cube $\phi(x) = x + \delta$, where δ is a small displacement value. The cube is broken into 2^{3n} cubelets, each 2^{-n} on a side, and ϕ is further restricted such that δ at the center of each cubelet may take on one of only four values, described in figure 1.

A circuit C may be defined that takes $3n$ input bits and computes, in polynomial time, six output bits. Given the binary representation of input integers $i, j, k \in [0, 2^n - 1]$, C's output represents δ at the center of cube c_{ijk}, defined naturally. Only four combinations of the six output bits are legal, corresponding to the four possible displacement vectors; again see figure 1.

C therefore "colors" each cubelet with one of the four displacements. Each 2-by-2-by-2 cube of cubelets has a single common vertex at its center, and the search problem on C is to find a *panchromatic* vertex, for which at least one of its eight adjacent cubelets is of each color. The function value at points between the sampled points may be interpolated by any natural rule to define the continuous function ϕ. We are

Δx^-	Δx^+	Δy^-	Δy^+	Δz^-	Δz^+	δ
1	0	1	0	1	0	$\delta_0 = (-\alpha, -\alpha, -\alpha)$
0	1	0	0	0	0	$\delta_1 = (\alpha, 0, 0)$
0	0	0	1	0	0	$\delta_2 = (0, \alpha, 0)$
0	0	0	0	0	1	$\delta_3 = (0, 0, \alpha)$

Figure 1: The four possible displacements of ϕ encoded by the six output bits of C, for some $\alpha \ll 2^{-n}$.

guaranteed, regardless of the specific interpolation rule, that a fixed point of ϕ, where $\delta = 0$, may only exist within the eight cubelets surrounding a panchromatic vertex. Finding such a vertex by calls to C is the **PPAD**-complete computational problem 3-Dimensional Brouwer [4].

2.2 Gadgets in [2]

The reduction in [2] from 3-Dimensional Brouwer to 3-Graphical Nash builds a bipartite graphical game $\mathcal{GG}$ in which each vertex has a binary pure strategy set $\{0, 1\}$. The strategy of vertex v may therefore be characterized by its probability $\mathbf{p}[v]$ of playing strategy 1. The left set of vertices are called *arithmetic* vertices (always designated v), while those on the right are *interior* vertices (designated w).

Gadgets are game subgraphs connecting zero, one, or two input arithmetic vertices to a single interior vertex, which is in turn connected to an arithmetic output vertex. The equilibrium strategies of the arithmetic vertices affect that of the interior vertex, while only the equilibrium strategy of the interior vertex affects that of the output vertex. Note that the chain of influence is unidirectional from input to output vertices, though $\mathcal{GG}$ itself is on an undirected graph and the payoff to the interior vertex likely relies on the strategies of both the input and output vertices. Gadgets may therefore be connected together, with each arithmetic vertex acting as the output of at most one gadget but the input of up to two. The paper defines nine gadgets, G_ζ, $G_=$, G_+, G_-, $G_\times$, $G_<$, $G_\wedge$, $G_\vee$, and $G_\neg$, each of which performs an arithmetic or logical operation on $\mathbf{p}$ of the input vertices to dictate the equilibrium $\mathbf{p}$ of the output vertex. $\mathbf{p}[v]$ may be duplicated to other arithmetic vertices to serve as input for more gadgets with $G_=$, thus keeping the degree of the graph ≤ 3.

Construction of $\mathcal{GG}$ begins by defining the coordinate vertices v_x, v_y, and v_z, for which $\mathbf{p}$ indicates the coordinates of the vertex closest to the origin on cubelet c_{ijk} for, *e.g.*, $i = 2^n x = 2^n \mathbf{p}[v_x]$. The game uses gadgets to extract from these input vertices the binary representation of each of $2^n x$, $2^n y$, and $2^n z$, writing them into the $3n$ vertices $v_{b_\ell(x)}$, $v_{b_\ell(y)}$, $v_{b_\ell(z)}$, $\ell \in [0, n]$. Note that, at equilibrium,

$$\mathbf{p}[v_{b_\ell(q)}] \in \{0, 1\} \; \forall \; \ell \in [0, n], q \in \{x, y, z\}.$$

Now, with the input bits to C established, all the internal logic of C is duplicated by logic gadgets proceeding from these input vertices, arriving finally at three output vertices v_{δ_x}, v_{δ_y}, v_{δ_z}. Each of these values is an input to an addition gadget, the other input of which is a copy of the corresponding coordinate vertex, and its output loops back to the corresponding coordinate vertex, forcing, *e.g.*, $x = x + \mathbf{p}[v_{\delta_x}]$. Starting from an arbitrary strategy profile, then, this graphical game has an incentive for the coordinate vertices to move toward a panchromatic vertex.

In practice, games are solved for so-called ϵ-Nash equilibria, where one strategy is considered better than another only if it increases a player's expected payoff by more than ϵ. The approximate nature of such a solution means that the result of the comparison gadget $G_<$ is uncertain when comparing vertices with near-equal p values. $G_<$ is therefore called a "brittle comparitor", and this brittleness combines with other effects of uncertain evaluation of equilibria to make the game find only approximately panchromatic vertices. To overcome this error, a constant number of copies of $\mathcal{GG}$ evaluate δ in a cube around the cubelet of interest, and their outputs are averaged and looped back to the initial coordinate vertices. For sufficiently many copies, the errors cancel out at panchromatic vertices and we are left with a game whose Nash equilibria are in an unambiguous one-to-one correspondence with the fixed points of the Brouwer function defined by C.

let $M = K^4 2^{4n+1}$
pick an arbitrary bijection $B : N_1 \to N_2$
for all pure strategy profiles $s = ((v,*),(w,*))$ **do**
    if $B(v) = w$ **then**
        let $u_s^{*1} = M$ and $u_s^{*2} = -M$
    else
        let $u_s^{*1} = u_s^{*2} = 0$

Figure 2: Payoffs in the initial game $\mathcal{G}^*$.

2.3 2-Nash is PPAD-Complete

In [1], Chen and Deng use the same scheme as above to simulate the circuit C in reducing from 3-Dimensional Brouwer, but instead realize all the gadgets in a specialized 2-player general game $\mathcal{G}$. The players in this game, P_1 and P_2, correspond to the halves of the bipartite graph in $\mathcal{GG}$, and each vertex v or w corresponds to a pair of pure strategies for the corresponding player: $(v,0),(v,1) \in S_1$ and $(w,0),(w,1) \in S_2$. The two parts of these pure strategies are called *nodes* and *sub-strategies*, but $\mathcal{G}$ is an ordinary two-player game; nodes are distinct from vertices in that they are simply convenient logical constructs, while vertices are actually part of the structure of a graphical game. The sets of nodes are N_1 and N_2, with $|N_1| = |N_2| = K$, a large number polynomial in $|C|$ such that the construction below does not exhaust the supply of strategies.

The pure strategy $(v,0)$ is interpreted as P_1 first "choosing" node v and then playing sub-strategy 0. A pure strategy profile $s \in S_1 \times S_2$ for $\mathcal{G}$ is thus an ordered pair of ordered pairs: one node and one sub-strategy per player. The payoff to player P_i under pure strategy profile s is denoted u_s^i. In the discussions below, an asterisk is used to replace arbitrary parts of strategy profiles; *e.g.*, $((v,0),(*,1))$ is any strategy profile in which P_1 plays node v, substrategy 0 and P_2 plays substrategy 1 of any node. For any v, $S_v = ((v,*),*)$ and similarly $S_w = (*,(w,*))$ for any w.

Since each node in $\mathcal{G}$ may play one of two sub-strategies of a chosen node, two values take the place of $\mathbf{p}$ above:

$$\begin{aligned} \mathbf{p}[v] &= Pr[P_1 \text{ plays } (v,1)] \\ \mathbf{p}_C[v] &= Pr[P_1 \text{ plays } (v,0) \textbf{ or } (v,1)] \end{aligned}$$

Construction of $\mathcal{G}$ begins with $\mathcal{G}^*$, defined in figure 2, which is a generalization of the two-player matching pennies game. The Nash equilibrium of $\mathcal{G}^*$ is a mixed strategy profile in which all strategies are played uniformly. [1] defines the class $\mathcal{L}$ of games whose payoffs are the same as those in $\mathcal{G}^*$, with each payoff increased by at most 1. The paper asserts without proof that, for any game $\mathcal{G}' \in \mathcal{L}$ at equilibrium, $\mathbf{p}_C$ is equal for all nodes in N_1 and N_2. This assertion is pivotal to the success of the reduction and is proven in the next section.

The construction of $\mathcal{G}$ from $\mathcal{G}^*$ proceeds parallel to that of $\mathcal{GG}$; where a gadget is added to $\mathcal{GG}$ that affects the payoffs for internal vertex w and output vertex v, a subset of the following payoffs in $\mathcal{G}^*$ are each increased by constants in $[0,1]$:

$$\begin{aligned} &\text{Payoffs to } P_1\text{:} \quad u_s^{*1}\ \forall\ s \in S_v \\ &\text{Payoffs to } P_2\text{:} \quad u_s^{*2}\ \forall\ s \in S_w \end{aligned}$$

Each strategy's payoff for either player is affected by at most one gadget, and so adding a gadget cannot interfere with the equilibria of previously added gadgets. Beginning with coordinate nodes v_x, v_y, and v_z as above, $\mathcal{G}$ derives the binary representations of i, j, and k and operates on them with logic gadgets, closing the loop back to the payoffs for the coordinate nodes. Each gadget increases payoffs for its corresponding nodes by at most 1, and therefore $\mathcal{G} \in \mathcal{L}$. At equilibrium, then, $0 \le K\mathbf{p}[v_x] \le K\mathbf{p}_C[v_x] \le 1$, and $x = K\mathbf{p}[v_x]$, truncated to a multiple of 2^{-n}, is the x coordinate of a panchromatic vertex; similarly v_y and v_z locate the vertex in the other dimensions.

With the same caveats as applied to the previous reduction regarding the brittle comparitor and approximate equilibria, requiring the same $O(1)$ duplication of effort and averaging, this construction reduces 3-Dimensional Brouwer to 2-Nash, proving its completeness in **PPAD**.

3 Explicit Expansions of Some Arguments in [2] and [1]

3.1 Selected Gadgets in $\mathcal{GG}$

[2] explicitly defines only $G_=$, G_+, and $G_<$, and proves the proper functioning of only the first two. We describe G_- and $G_\wedge$ in detail below; G_- is of particular use in this method of reducing to 3-Graphical Nash, and $G_\wedge$ serves as an example construction for all the logic gadgets. Note that in the proofs below we indicate by $v:0$ the case that vertex v plays strategy 0 and by $E[v]$ the expected payoff for v.

$\mathbf{G}_-$ involves two input vertices, a and b, one interior vertex c, and an output vertex d, with the action that $\mathbf{p}[d] = \mathbf{p}[a] - \mathbf{p}[b]$ at equilibrium. The payoffs to c and d are as follows:

Payoffs to c, $c:0$

	$a:0$	$a:1$
$b:0$	0	-1
$b:1$	1	0

Payoffs to c, $c:1$

$d:0$	0
$d:1$	1

Payoffs to d

	$c:0$	$c:1$
$d:0$	0	1
$d:1$	1	0

G_- carries the precondition that $\mathbf{p}[a] \geq \mathbf{p}[b]$. We analyze the possible strategies in cases.

$$\begin{array}{rcl} c:0 & \Rightarrow & E[c] = \mathbf{p}[a] - \mathbf{p}[b] \\ c:1 & \Rightarrow & E[c] = \mathbf{p}[d] \end{array}$$

so

$$\begin{array}{rcl|rcl} \mathbf{p}[d] > \mathbf{p}[a] - \mathbf{p}[b] & \Rightarrow & \mathbf{p}[c] = 1 & \mathbf{p}[d] < \mathbf{p}[a] - \mathbf{p}[b] & \Rightarrow & \mathbf{p}[c] = 0 \\ & \Rightarrow & \mathbf{p}[d] = 0 & & \Rightarrow & \mathbf{p}[d] = 1 \\ & \Rightarrow & \textit{contradiction} & & \Rightarrow & \textit{contradiction} \end{array}$$

Thus we see that $\mathbf{p}[d]$ may be neither greater nor less than $\mathbf{p}[a] - \mathbf{p}[b]$ at equilibrium, and so the gadget works.

$\mathbf{G}_\wedge$ similarly forces $\mathbf{p}[d] = \mathbf{p}[a] \wedge \mathbf{p}[b]$ at equilibrium, for binary vertices a and b. The payoffs to c and d are as follows:

Payoffs to c, $c:0$

	$a:0$	$a:1$
$b:0$	0	0
$b:1$	0	1

Payoffs to c, $c:1$

$d:0$	0
$d:1$	1

Payoffs to d

	$c:0$	$c:1$
$d:0$	0	1
$d:1$	1	0

Again we analyze the possible strategies in cases:

$$\begin{array}{rcl} c:0 & \Rightarrow & E[c] = \mathbf{p}[a] \wedge \mathbf{p}[b] \\ c:1 & \Rightarrow & E[c] = \mathbf{p}[d] \end{array}$$

so

$$\begin{array}{rcl|rcl} \mathbf{p}[d] > (\mathbf{p}[a] \wedge \mathbf{p}[b]) & \Rightarrow & \mathbf{p}[c] = 1 & \mathbf{p}[d] < (\mathbf{p}[a] \wedge \mathbf{p}[b]) & \Rightarrow & \mathbf{p}[c] = 0 \\ & \Rightarrow & \mathbf{p}[d] = 0 & & \Rightarrow & \mathbf{p}[d] = 1 \\ & \Rightarrow & \textit{contradiction} & & \Rightarrow & \textit{contradiction} \end{array}$$

Thus we see that $\mathbf{p}[d]$ may be neither greater nor less than $\mathbf{p}[a] \wedge \mathbf{p}[b]$ at equilibrium, and so the gadget works.

3.2 Bit Extraction Structure in $\mathcal{GG}$

While the duplication of 3-Dimensional Brouwer circuit C with logic gadgets is straightforward, it relies on a novel assembly of gadgets to extract a binary representation of the cubelet indices i, j, and k from the real values $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z] \in [0,1]$. Here we describe a sub-game G_b composed of a system of gadgets that may be chained together n times to extract all the necessary bits from one coordinate vertex. This system is illustrated schematically in figure 3.

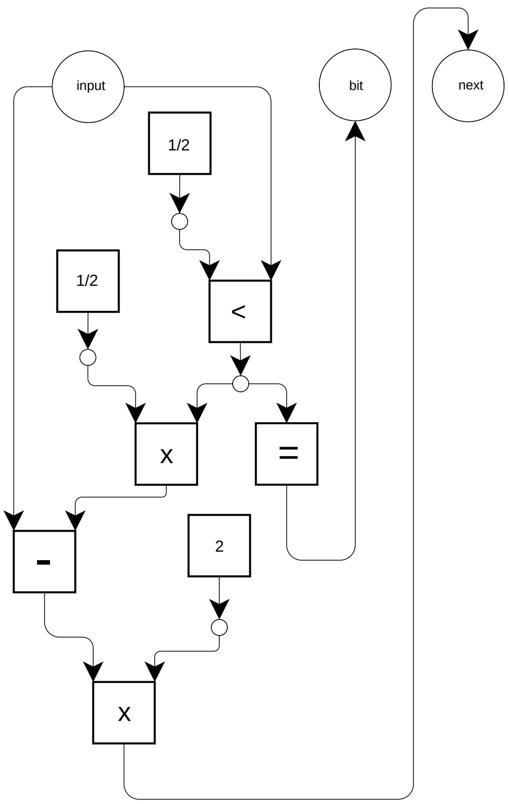

Figure 3: Graphical gadgets (squares) assembled together with additional arithmetic vertices (circles) into a system that extracts one bit of a cubelet index.

G_b builds up i from $\mathbf{p}[v_x]$ one bit at a time, starting with the highest-order bit, by multiplying $\mathbf{p}[v_x]$ by 2 a total of n times and each time extracting the non-fractional part of the result. Therefore we need only design a system that extracts a bit from $2\mathbf{p}[v]$ and passes on the remainder to another vertex. For any vertex v, $\mathbf{p}[v] \in [0,1]$, and so operations must be rearranged to insure $\mathbf{p}[v] \leq \frac{1}{2}$ before the multiplication.

The highest order bit of $2\mathbf{p}[v]$ will be 1 only if $\mathbf{p}[v] \geq \frac{1}{2}$, so first a $G_<$ compares the output of a G_ζ with $\zeta = \frac{1}{2}$ to v. A $G_=$ copies the value of the output vertex for this gadget to the appropriate v_{b_ℓ}. If the bit is 0, then nothing more needs to be done, and we may double $\mathbf{p}[v]$ and pass it on to the next iteration. If it is 1, though, we must first subtract $\frac{1}{2}$, so we use the output of $G_<$ and another G_ζ as the inputs to a $G_\times$, and then pass the output (which has $\mathbf{p} \in \{0, \frac{1}{2}\}$) and v to G_-. We know that the output of this gadget must have $\mathbf{p} \leq \frac{1}{2}$, and so finally we use another pair of G_ζ and $G_\times$ to multiply this value by 2 and pass it on to the next system. For the vertices v_{in}, v_{bit}, and v_{next}, we have $2\mathbf{p}[v_{in}] = \mathbf{p}[v_{bit}] + \mathbf{p}[v_{next}]$, and the iterative structure of G_b is complete.

3.3 Equilibria of Games in $\mathcal{L}$

In [1] it is claimed that increasing payoffs in $\mathcal{G}^*$ by at most 1 keeps $\mathbf{p}_C[v]$ equal for all nodes v at equilibrium. However, as mentioned above, it is expected that our solutions need only be approximate equilibria, and so the claim must be stated and proven more accurately in this context. Note that it is only for clarity and brevity that this report has elided epsilons from most of its statements; the source papers include uncertainty in all their results and proofs.

For players P_1 and P_2, $\mathcal{G}^*$ and $\mathcal{G}$ have node sets N_1 and N_2 with $|N_1| = |N_2| = K$ sufficient to define all the necessary gadgets to simulate the Brouwer circuit C, which operates on three binary inputs each of length n. S_1 and S_2 are the strategy sets for the two players; with strategies $\{0,1\}$ available for each node, this gives $|S_1| = |S_2| = 2K$.

$\mathcal{G}^*$ is defined in figure 2; based on this game we define a class of games $\mathcal{L}$ as follows:

$$\mathcal{G}' \in \mathcal{L} \textit{ iff } {u'}_s^i \in [u_s^{*i}, u_s^{*i}+1] \;\forall\; s \in S_1 \times S_2, i \in \{1,2\}.$$

At the conclusion of section 2.3, we determined that the game $\mathcal{G}$ simulating C is in $\mathcal{L}$, so for the rest of this section we ignore C and concentrate on general properties of games in $\mathcal{L}$. For constants $\epsilon = K^{-1}2^{-4n}$ and $M = K^4 2^{4n+1}$, we wish to prove the following claim about a δ-Nash equilibrium ($\delta \leq 1$) with mixed strategy profile $\mathbf{p}$ in a game $\mathcal{G} \in \mathcal{L}$:

$$\mathbf{p}_C[v], \mathbf{p}_C[w] \in \left[\frac{1}{K} - \epsilon, \frac{1}{K} + \epsilon\right] \;\forall\; v \in N_1, w \in N_2. \tag{1}$$

In the following proof, all variants of v are assumed to be in N_1 and those of w in N_2, and B is the bijection between N_1 and N_2 that defines G^*. We modify our notation of expected payoff so that $E[v]$ is the total expected payoff to P_1 of strategies $(v,0)$ and $(v,1)$. Let us assume the claim does not hold for some v; then either $\mathbf{p}_C[v] > \frac{1}{K} + \epsilon$ or $\mathbf{p}_C[v] < \frac{1}{K} - \epsilon$. We approach these cases separately:

- Assume $\mathbf{p}_C[v] > \frac{1}{K} + \epsilon$; then $\exists\, v' \textit{ s.t. } \mathbf{p}_C[v'] < \frac{1}{K} - \frac{\epsilon}{K-1}$. Let $w = B(v)$, $w' = B(v')$.

  P_1's high weight on v and low weight on v' mean that P_2 will want to de-emphasize w and emphasize w' to maximize her payoff. Equation 1 holds under equilibrium, then, only if $E[w'] - E[w]$ is sufficiently small.

$$\begin{aligned}
E[w] &\leq \mathbf{p}_C[v](-M+1) + (1 - \mathbf{p}_C[v]) \\
&= -\mathbf{p}_C[v]M + 1 \\
&< \frac{-M}{K} - \epsilon M + 1 \\
E[w'] &\geq \mathbf{p}_C[v'](-M) \\
&> \frac{-M}{K} + \frac{\epsilon M}{K-1} \\
E[w'] - E[w] &> \frac{K\epsilon M - K + 1}{K-1} \\
&= \frac{2K^4 - K + 1}{K-1} > K^3 > 1 \geq \delta
\end{aligned}$$

  By definition of δ-Nash equilibrium, this implies that $\mathbf{p}_C[w] = 0$. Thus $\exists\, w'' \neq w \textit{ s.t. } \mathbf{p}_C[w''] \geq \frac{1}{K-1}$. Now, we avoid contradiction under equilibrium only if $E[B^{-1}(w'')] - E[v]$ is sufficiently small.

$$\begin{aligned}
E[v] &\leq K - 1 \\
E[B^{-1}(w'')] &\geq \frac{1}{K-1} M \\
E[B^{-1}(w'')] - E[v] &\geq \frac{M}{K-1} - K + 1 \\
&> 2^{4n+1} K^3 > 1 \geq \delta
\end{aligned}$$

  Thus $\mathbf{p}_C[v] = 0$ and we derive a contradiction.

- Assume $\mathbf{p}_C[v] < \frac{1}{K} - \epsilon$; then $\exists\, v' \textit{ s.t. } \mathbf{p}_C[v'] > \frac{1}{K} + \frac{\epsilon}{K-1}$. Let $w = B(v)$, $w' = B(v')$.

  P_1's weights on v and v' mean that P_2 will want to emphasize w and de-emphasize w'. Equation 1 holds under equilibrium, then, only if $E[w] - E[w']$ is sufficiently small.

$$\begin{array}{rcl}
E[w] & \geq & \mathbf{p}_C[v](-M) \\
 & > & \dfrac{-M}{K} + \epsilon M \\
E[w'] & \leq & \mathbf{p}_C[v'](-M+1) + (1 - \mathbf{p}_C[v']) \\
 & < & \dfrac{-M}{K} - \dfrac{\epsilon M}{K-1} + 1 \\
E[w] - E[w'] & > & \dfrac{K\epsilon M - K + 1}{K-1} \\
 & = & \dfrac{2K^4 - K + 1}{K-1} > K^3 > 1 \geq \delta
\end{array}$$

This implies $\mathbf{p}_C[w'] = 0$, and so $\exists\ w'' \neq w'\ s.t.\ \mathbf{p}_C[w''] \geq \frac{1}{K-1}$. Now, we avoid contradiction under equilibrium only if $E[B^{-1}(w'')] - E[v']$ is sufficiently small.

$$\begin{array}{rcl}
E[v'] & \leq & K - 1 \\
E[B^{-1}(w'')] & \geq & \dfrac{1}{K-1} M \\
E[B^{-1}(w'')] - E[v'] & \geq & \dfrac{M}{K-1} - K + 1 \\
 & > & 2^{4n+1} K^3 > 1 \geq \delta
\end{array}$$

Thus $\mathbf{p}_C[v'] = 0$ and we derive a contradiction.

We have proven that $\mathbf{p}_C[v] \leq \frac{1}{K} + \epsilon$ and $\mathbf{p}_C[v] \geq \frac{1}{K} - \epsilon$, and the proof for $\mathbf{p}_C[w] \leq \frac{1}{K} + \epsilon$ and $\mathbf{p}_C[w] \geq \frac{1}{K} - \epsilon$ proceeds similarly. Some equilibrium must exist, and therefore we have proven that, at equilibrium in $\mathcal{G} \in \mathcal{L}$, $\mathbf{p}_C[v], \mathbf{p}_C[w] \in \left[\frac{1}{K} - \epsilon, \frac{1}{K} + \epsilon\right]\ \ \forall\ v \in N_1, w \in N_2$.

3.4 Gadget G_- in $\mathcal{G}$

Because negative values of $\mathbf{p}$ are not possible, the subtraction gadget in [1] deserves special scrutiny. Here, therefore, we state its construction and prove that it gives the desired equilibrium weights.

We add G_- to a game $\mathcal{G}' \in \mathcal{L}$ by modifying payoffs from $\mathcal{G}^*$ for two input nodes $v_1, v_2 \in N_1$, intermediate node $w \in N_2$, and output node $v_3 \in N_1$. The following payoff matrices show increases to u^* that give u', with v' and w' representing all other nodes, respectively, in N_1 and N_2 such that $v' \neq v_1, v_2, v_3$ and $w' \neq w$.

Payoffs to P_1

		v_1		v_2		v_3		v'	
w	0	0	0	0	0	+1	0	0	0
	1	0	0	0	0	0	+1	0	0
w'	0	0	0	0	0	0	0		
	1	0	0	0	0	0	0		

Payoffs to P_2

		v_1		v_2		v_3		v'	
w	0	0	0	0	+1	0	+1	0	0
	1	0	+1	0	0	0	0	0	0
w'	0	0	0	0	0	0	0		
	1	0	0	0	0	0	0		

Once again expanding our notation for expected payoffs, $E[v, 0]$ is the expected payoff to P_1 for playing strategy $(v, 0)$. Then

$$\begin{array}{rcl}
E[w,1] - E[w,0] & = & \mathbf{p}[v_1] - \mathbf{p}[v_2] - \mathbf{p}[v_3] \\
E[v_3,1] - E[v_3,0] & = & \mathbf{p}[w] - (\mathbf{p}_C[w] - \mathbf{p}[w])
\end{array}$$

If, at ϵ-Nash equilibrium, $\mathbf{p}[v_3] > \max(\mathbf{p}[v_1] - \mathbf{p}[v_2], 0) + \epsilon$, the first equation implies $\mathbf{p}[w] = 0$, and so the second equation implies $\mathbf{p}[v_3] = 0$, and so we derive a contradiction.

If instead $\mathbf{p}[v_3] < \min(\mathbf{p}[v_1] - \mathbf{p}[v_2], \mathbf{p}_C[v_3]) - \epsilon$, the first equation implies $\mathbf{p}[w] = \mathbf{p}_C[w]$, and so the second equation implies $\mathbf{p}[v_3] = \mathbf{p}_C[v_3]$, and we again derive a contradiction. Thus, at ϵ-Nash equilibrium, we have

$$\min(\mathbf{p}[v_1] - \mathbf{p}[v_1], \mathbf{p}_C[v_3]) - \epsilon \leq \mathbf{p}[v_3] \leq \max(\mathbf{p}[v_1] - \mathbf{p}[v_1], 0) + \epsilon$$

and G_- works as expected.

References

[1] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. *Electronic Colloquium in Computational Complexity*, TR05-140, February 2006.

[2] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. The complexity of computing a Nash equilibrium. *Electronic Colloquium in Computational Complexity*, TR05-115, September 2005.

[3] Paul Goldberg and Christos Papadimitriou. Reducibility among equilibrium problems. *Electronic Colloquium in Computational Complexity*, TR05-90, August 2005.

[4] Christos Papdimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Science*, 48(3), June 1994.