

A Reinforcement Learning Approach to Finding Nash Equilibrium of Multi-player Matrix Games

Vishnu Nanduri, *Student member IIE* and Tapas K. Das, *Fellow IIE*
Department of Industrial and Management Systems Engineering
University of South Florida, Tampa, FL 33620
E-mail: vnanduri, das@eng.usf.edu

April 11, 2007

Abstract

A Reinforcement Learning Approach to Finding Nash Equilibrium of Multi-player Matrix Games

With the advent of e-commerce, the contemporary marketplace has evolved significantly toward competition-based trading of goods and services. Many such market scenarios can be modeled as matrix games. This paper presents a computational algorithm to obtain Nash equilibria of n -player matrix games. The algorithm uses a stochastic approximation based reinforcement learning (RL) approach and has the potential to solve n -player matrix games with large action spaces. The proposed RL algorithm uses a value iteration approach, which is well established in the MDP/SMDP literature. Results from the recent studies establishing the existence of equivalent matrix games for stochastic games with both discounted and average rewards are outlined. These results emphasize the fact that the solution of matrix games is also critical to solving a large class of stochastic games. The RL algorithm is tested on a set of sixteen matrix games with up to four players and sixty four action choices. Also studied in detail is an example of a matrix game representing strategic bidding in an electric power network with two players and 625 action choices each. The algorithm presented here can benefit from the availability of tera/peta scale cluster computing in overcoming the computational barriers arising from large state-action spaces, and thus impact a broad range of decision making problems.

1 Introduction

Globalization has played a significant role over the last decade in transforming the marketplace into one where most goods and services are transacted through multi-party competition. Consequently, the study of game theoretic concepts and the development of effective methods for solving multiplayer games have gained increasing attention in the research literature. Games occur in two primary forms: matrix games and stochastic games. An n -player matrix game is characterized by n different reward matrices (one for each player) and a set of action combinations characterizing the equilibria (Nash-equilibria, in particular). Nash [1951] defined equilibrium to be an action combination from which no single player could unilaterally deviate to increase profit. Stochastic games are comprised of finite or infinite horizon stochastic processes with finite states and state transition probability structure, in which the players seek equilibrium actions for every state so as to maximize their rewards from the overall game. Therefore, stochastic games are construed as sequence of matrix games (one for each state) connected with transition probabilities. Further classification of games arises from the nature of reward structure: zero sum games and nonzero (general) sum games. Rewards of stochastic games are classified as discounted reward, average reward, and total reward.

Though the fundamentals of game theory are fairly well established (Nash [1951]), the computational difficulties associated with finding Nash equilibria have constrained the scope of the research literature largely to the study of bimatrix games with limited action choices. Even in the absence of sufficient tools to appropriately analyze stochastic or matrix games, a majority of the marketplaces have evolved to incorporate transactions through competition. Therefore, to ensure healthy growth of the current competition based economy, it is imperative to develop computationally feasible tools to solve large scale stochastic and matrix games. In recent years, researchers have been able to characterize equivalent matrix games for both discounted reward and average reward stochastic games (Li et al. [2007], Hu and Wellman [1998], Borkar [2002], Filar and Vrieze [1997]). They also harnessed the advances in reinforcement learning based techniques to construct these equivalent matrix games (Li et al. [2007], Hu and Wellman [1998]). However, obtaining the Nash equilibrium for these equivalent matrix games has remained an open research issue, which is the focus of this paper.

As discussed in McKelvey and McLennan [1996], the appropriate method of computing Nash equilibria of a matrix game depends on 1) whether it is required to find one or all equilibrium points, 2) number of players in the game, and 3) importance of the *value* of the Nash equilibrium. No computationally viable method addressing all of the above is available in existing literature. Nash equilibria of n -player matrix games can be obtained by solving a nonlinear complementarity problem (NCP), which for a 2-player game becomes a linear complementarity problem (LCP) (McKelvey and McLennan [1996]). Lemke and Howson Jr. [1964] developed an efficient algorithm for obtaining Nash equilibria for bimatrix games by solving the associated LCP. Their algorithm was extended for finding Nash equilibria of n -person games by Rosenmuller [1971] and Wilson [1971]. However, these algorithms still have unresolved computational challenges. Mathiesen [1985] proposed a method of solving NCP for n -player games through a sequence of LCP approximations. A survey by Harker and Pang [1990] summarizes these and other developments on this topic. It may be noted that these

methods are not guaranteed to obtain global convergence and often depend on the choice of the starting point. To our knowledge, the only openly available software that attempts to solve multiplayer matrix games is Gambit (McKelvey et al. [2007]). However, as observed by Lee and Baldick [2003], this software takes an unusually long computation time as the number of players and their action choices increase.

Game theoretic models have been studied extensively in examining market competition in the energy and transmission segments of deregulated power markets (as in Pennsylvania-Jersey-Maryland, New York, New England, and Texas). These games are characterized by multidimensional bid vectors with continuous parameters. Upon suitable discretization of these bid vectors, many of these games can be formulated as matrix games. The degree of discretization dictates both the computational burden and the probability of identifying the Nash equilibria. Almost all of the literature studying power market games is devoted to optimization based approaches, such as mathematical programming (Hobbs [2001], Hobbs et al. [2000], Yao et al. [2003]), co-evolutionary programming (Price [1997]), and exhaustive search (Cunningham et al. [2002]). Even in a limited number of studies, where such games are formulated as matrix games, numerical examples are converted to bimatrix games and are solved using linear programming and LCP approaches (Lee and Baldick [2003], Ferrero et al. [1999], Stoff [1999]).

Mathematical programming approach to finding NE has two primary variants: mathematical program with equilibrium constraints (MPEC, Luo et al. [1996]), and equilibrium problem with equilibrium constraints (EPEC, Su [2005]). MPEC is a generalization of bilevel programming, which in turn is a special case of hierarchical mathematical programming (with two or more levels of optimization). MPECs resemble Stackelberg (leader-follower) games, which form a special case of the Nash game. In a Nash game each player possesses the same amount of information about competing players, whereas, in Stackelberg type games, a leader can anticipate the reactions of the other players, and thus possesses more information in the game. The leader in a Stackelberg game chooses a strategy from his/her strategy set, and the followers choose a response based on the leaders actions (Luo et al. [1996]), while in a Nash game all players choose actions simultaneously. When multiple generators face optimization problems in the form of MPECs, EPEC models have been used to simultaneously find the equilibria of the MPECs (Yao et al. [2003], Su [2005], Ralph and Smeers [2006], Cardell et al. [1997]).

The primary contribution of this paper is a novel stochastic approximation based reinforcement learning algorithm for obtaining NE of n -player matrix games. Extensive numerical experimentation is presented in Section 5, which demonstrates the ability of the learning algorithm to obtain NE. This section includes sixteen matrix games with up to four players and sixty four actions for each player, followed by an example of a deregulated power network with competing generators. The numerical results indicate that the learning based approach presented in this paper holds significant promise in its ability to obtain NE for large n -player matrix games. To our knowledge, the algorithm is the first of its kind that harnesses the power of stochastic value approximation method that has been successfully used in solving large scale Markov and semi-Markov decision process problems with single decision makers (Das et al. [1999], Gosavi et al. [2002], Gosavi [2004]).

In what follows, we first present a formal definition of a matrix game and its Nash equilibrium (Section 2). Thereafter, we discuss the key results from literature that show that for both discounted and average reward stochastic games there exist equivalent matrix games and such matrix games can be constructed using a stochastic approximation based learning approach. These results serve to demonstrate two key elements: 1) importance of matrix games in solving stochastic games and 2) the concept of stochastic approximation based learning of values (Section 3). We subsequently present the learning algorithm, a detailed numerical study, and concluding remarks in Sections 4, 5, and 6 respectively.

2 Matrix Games

A matrix game can be defined by a tuple $\langle n, A^1, \dots, A^n, R^1, \dots, R^n \rangle$. The elements of the tuple are as follows.

- n denotes the number of players.
- A^k denotes the set of actions available to player k .
- $R^k : A^1 \times \dots \times A^n \rightarrow \mathbb{R}$ is the payoff function for player k , where an element $r^k(a^1, \dots, a^n)$ of R^k is the payoff to player k when the players choose actions a^1 through a^n .

It is called a matrix game, since R^k for all k , can be written as n -dimensional matrices as follows

$$R^k = [r^k(a^1, a^2, \dots, a^n)]_{a^1=1, \dots, a^n=1}^{|A^1|, \dots, |A^n|}$$

The players select actions from the set of available action choices with the goal of maximizing their payoffs which depend on all the players' actions. A pure strategy Nash equilibrium for a matrix game is an action profile (a_*^1, \dots, a_*^n) , for which $r^k(a_*^k, a_*^{-k}) \geq r^k(a^k, a_*^{-k})$, $\forall a^k \in A^k$, and $k = 1, 2, \dots, n$. The equilibrium values denoted by $Val[\cdot]$ for player k with payoff matrices R^k are obtained as $Val[R^k] = r^k(a_*^1, \dots, a_*^n)$. A mixed strategy Nash equilibrium for matrix games is a vector $(\pi_*^1, \dots, \pi_*^n)$, for which we can write

$$\sum_{a^1=1}^{|A^1|} \dots \sum_{a^n=1}^{|A^n|} \pi_*^k(a^k) \pi_*^{-k}(a^{-k}) r^k(a^k, a^{-k}) \geq \sum_{a^1=1}^{|A^1|} \dots \sum_{a^n=1}^{|A^n|} \pi_*^k(a^k) \pi_*^{-k}(a^{-k}) r^k(a^k, a^{-k}),$$

where $\pi_*^{-k}(a^{-k}) = \pi_*^1(a^1) \dots \pi_*^{k-1}(a^{k-1}) \pi_*^{k+1}(a^{k+1}) \dots \pi_*^n(a^n)$.

A matrix game may not have a pure strategy Nash equilibrium, but it always has a mixed strategy Nash equilibrium (Nash [1951]). Since matrix games are static, they can be viewed as recursive stochastic games with a single state.

3 Equivalent Matrix Games for Stochastic Games

As alluded to in Section 1, a general sum stochastic game has equivalent matrix games. Therefore, once its equivalent matrix games are established, solution of a stochastic game reduces to solving the set of matrix games (one for each state). The intent of this section is to provide a brief overview of the main results from the recent literature concerning the existence of equivalent matrix games and the algorithms to construct these matrix games. We adopted the concept of stochastic approximation based learning of values, which is used in the above algorithms (Hu and Wellman [1998], Li et al. [2007]), in developing our methodology for obtaining Nash equilibrium of matrix games (presented in Section 4.1).

3.1 Stochastic Games

A stochastic game can be defined by a tuple $\langle n, S, A^1, \dots, A^n, P, \tilde{R}^1, \dots, \tilde{R}^n \rangle$. The elements of the tuple are as follows.

- n denotes the number of agents/players/decision makers.
- S denotes the finite set of states of the environment.
- A^1, \dots, A^n denote the collection of finite set of admissible actions to the agents $1, \dots, n$, where $m^i(s) = |A^i(s)|, i = 1, \dots, n$, are the cardinalities of the action spaces in state s . Let $\mathbf{A} = A^1 \times \dots \times A^n$. Each agent chooses an action $a^i \in A^i$ resulting in a joint action vector $\mathbf{a} = (a^1, \dots, a^n)$. Let $\mathbf{a}^{-i} = (a^1, \dots, a^{i-1}, a^{i+1}, \dots, a^n)$ be vector of actions of all players except agent i .
- $P : S \times \mathbf{A} \rightarrow \Pi(S)$ is the transition function, where $\Pi(\cdot)$ indicates the probability distribution. An element $p(s' | s, \mathbf{a})$ of P is the transition probability of reaching state s' as a result of a joint action \mathbf{a} by all of the n players. $\tilde{R}^k : S \times \mathbf{A} \rightarrow \mathbb{R}$ is the reward function that gives the expected immediate reward gained by the player k for each set of actions of each agents in each state. For example, an element $r^k(s, \mathbf{a})$ of \tilde{R}^k is the expected reward of player k in state s when the agents choose the action vector \mathbf{a} .

In a stochastic game, the transition probabilities and the reward functions depend on the choices made by all agents. Thus, from the perspective of an agent, the game environment is nonstationary during its evolution phase. However, for stochastic games, optimal strategies constitute stationary policies and hence it is sufficient to consider only the stationary strategies (Filar and Vrieze [1997]). We define $\pi^k(s)$ as the mixed strategy at state s for agent k , which is the probability distribution over available action set, A^k , of player k . Thus $\pi^k(s) = \{\pi^k(s, a) : a \in A^k(s)\}$, where $\pi^k(s, a)$ denotes the probability of player k choosing action a in state s , and $\sum_{a \in A^k(s)} \pi^k(s, a) = 1$. Then $\pi = (\pi^1, \dots, \pi^n)$ denotes a joint mixed strategy of all n players, also called a policy. A pure action $a \in A^k$ can be treated as a mixed strategy π^k for which $\pi^k(a) = 1$.

Under policy π , the transition probability can be given as

$$p(s' | s, \pi) = \sum_{a^1=1}^{m^1(s)} \dots \sum_{a^n=1}^{m^n(s)} p(s' | s, a^1, \dots, a^n) \pi^n(s, a^n) \dots \pi^1(s, a^1).$$

The immediate expected reward of player k induced by a mixed strategy π in a state s is given by

$$r^k(s, \pi) = \sum_{a^1=1}^{m^1(s)} \dots \sum_{a^n=1}^{m^n(s)} r^k(s, a^1, \dots, a^n) \pi^n(s, a^n) \dots \pi^1(s, a^1).$$

Then the overall discounted value of a policy π to player k starting in state s can be given as

$$V_\beta^k(s, \pi) = \sum_{t=0}^{\infty} \beta^t E_s(r_t^k) = \sum_{t=0}^{\infty} \beta^t \sum_{s' \in S} p^t(s' | s, \pi) r^k(s', \pi),$$

where $p^t(\cdot)$ denotes an element of the t^{th} power of the transition probability matrix P . The overall average value of a policy π to player k starting in state s can be given as

$$V_\alpha^k(s, \pi) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E_s(r_t^k) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} p^t(s' | s, \pi) r^k(s', \pi).$$

3.1.1 Equivalent Matrix Games for NE of Discounted Reward Stochastic Games

Let $Q^k(\cdot)$ denote the payoff (value) matrix of player k for all possible pure action sets \mathbf{a} , which take into account both the immediate reward and the future opportunities. For $s \in S$, the matrix with size $m^1(s) \times m^2(s) \times \dots \times m^n(s)$ for the k^{th} player can be given by

$$\begin{aligned} Q^k(s) &= Q^k(s, \mathbf{a}) \\ &= \left[(1 - \beta)r^k(s, a^1, \dots, a^n) + \beta \sum_{s' \in S} p(s' | s, a^1, \dots, a^n) V_\beta^k(s', \pi_*) \right]_{\substack{a^1=1, \dots, a^n=1 \\ a^1=m^1(s), \dots, a^n=m^n(s)}} \end{aligned} \quad (1)$$

where $V_\beta^k(s', \pi_*)$ is the equilibrium value for the stochastic game starting at state s' for player k . Note that this equivalent matrix, $Q^k(\cdot)$ captures the information from the matrix game resulting from the pure strategies as well as the equilibrium payoff of the stochastic game. This enables us to establish a connection between the matrix games and stochastic games.

If the players use a mixed strategy π , the value of the above matrix game for player k can

be obtained as:

$$\begin{aligned}
Val[Q^k(s), \pi] &= \sum_{a^1=1}^{m^1(s)} \dots \sum_{a^n=1}^{m^n(s)} \pi^1(s, a^1) \dots \pi^n(s, a^n) Q^k(s, a^1, \dots, a^n) \\
&= \sum_{a^1=1}^{m^1(s)} \dots \sum_{a^n=1}^{m^n(s)} \pi^1(s, a^1) \dots \pi^n(s, a^n) \{(1 - \beta)r^k(s, a^1, \dots, a^n) + \\
&\quad \beta \sum_{s' \in S} p(s' | s, a^1, \dots, a^n) V_\beta^k(s', \pi_*)\} \\
&= (1 - \beta)r^k(s, \pi) + \beta \sum_{s' \in S} p(s' | s, \pi) V_\beta^k(s', \pi_*)
\end{aligned}$$

For the equivalent matrix game (1), the equilibrium point π_* is defined such that

$$Val[Q^k(s), \pi_*] \geq Val[Q^k(s), \pi_*^{-k}, \pi^k]. \quad (2)$$

The following result from Li et al. [2007] formally establishes the connection between the matrix games and the stochastic game.

- **Theorem 1** *The following are equivalent: (i) π_* is an equilibrium point in the discounted reward stochastic game with equilibrium payoffs $(V_\beta^1(\pi_*), \dots, V_\beta^n(\pi_*))$. (ii) For each $s \in S$, the strategy $\pi_*(s)$ constitutes an equilibrium point in the static n -dimensional matrix game $(Q^1(s), \dots, Q^n(s))$ with equilibrium payoffs $(Val[Q^1(s), \pi_*], \dots, Val[Q^n(s), \pi_*])$. The entry of $Q^k(s)$ corresponding to actions $\mathbf{a} = (a^1, \dots, a^n)$ is given by $Q^k(s, \mathbf{a}) = (1 - \beta)r^k(s, \mathbf{a}) + \beta \sum_{s' \in S} p(s' | s, \mathbf{a}) V_\beta^k(s', \pi_*)$, for $k = 1, \dots, n$, where $\mathbf{a} \in \prod_{k=1}^n A^k(s)$.*

Since it can be shown that $V_\beta^k(s, \pi_*) = Val[Q^k(s), \pi_*]$, the element of the equivalent matrix can be written as:

$$Q^k(s, a^1, \dots, a^n) = (1 - \beta)r^k(s, a^1, \dots, a^n) + \beta \sum_{s' \in S} p(s' | s, a^1, \dots, a^n) Val[Q^k(s'), \pi_*], \quad (3)$$

where π_* is the Nash equilibrium point for matrix game $Q(\cdot)$, and

$$Val[Q^k(s), \pi_*] = r^k(s, \pi_*) + \beta \sum_{s' \in S} p(s' | s, \pi_*) Val[Q^k(s'), \pi_*]. \quad (4)$$

The above results establish the existence and the structure of the equivalent matrix games for discounted reward stochastic games. The assumption is that once the equivalent matrix games are constructed, they can be solved using existing methods. However, there are two primary hurdles in the process of constructing matrix games using (3).

1. It is difficult to obtain the transition probabilities $(p(s'|s, a^1, \dots, a^n))$ for most real life problems, and
2. existing methods for obtaining NE value $(Val[Q^k(s'), \pi_*])$ of n -player ($n > 2$) matrix games are fraught with computational and convergence related challenges (Rosenmuller [1971], Wilson [1971]).

The unavailability of the transition probabilities was addressed in the literature (Hu and Wellman [1998]) by the use of a stochastic approximation approach, in which the matrix elements are constructed through a sequential learning approach as follows.

$$Q_{t+1}^k(s, \mathbf{a}) = (1 - \gamma_t)Q_t^k(s, \mathbf{a}) + \gamma_t [r^k(s, \mathbf{a}) + \beta Val(Q_t^k(s'), (\pi_t)_*)], \quad (5)$$

where γ_t is the learning parameter.

However, development of a computationally viable method of finding the NE value of a matrix game $(Val(Q_t^k(s'), (\pi_t)_*))$, as needed in (5), is still an open research issue and is addressed in the following section. For the sake of completeness, we provide below the equations for equivalent matrix games and the corresponding learning algorithm for average reward stochastic games.

3.1.2 Equivalent Matrix Games for NE of Average Reward Stochastic Games

Let $\hat{R}^k(\cdot)$ denote the reward matrix for player k and $V_\alpha^k(\pi_*)$ denote the gain equilibrium value for an average reward matrix game. We can write an element of $\hat{R}^k(\cdot)$ as

$$\hat{R}^k(s, a^1, \dots, a^n) = r^k(s, a^1, \dots, a^n) - V_\alpha^k(\pi_*) + \sum_{s' \in S} p(s'|s, a^1, \dots, a^n) Val[\hat{R}^k(s'), \pi_*], \quad (6)$$

where π_* is the Nash equilibrium point for matrix game $\hat{R}(\cdot)$, and

$$Val[\hat{R}^k(s), \pi_*] = r^k(s, \pi_*) - V_\alpha^k(\pi_*) + \sum_{s' \in S} p(s'|s, \pi_*) Val[\hat{R}^k(s'), \pi_*]. \quad (7)$$

A two time scale algorithm presented in (Li et al. [2007]) for *learning* the $\hat{R}^k(\cdot)$ elements is given below:

$$\hat{R}_{t+1}^k(s, \mathbf{a}) = (1 - \gamma_t)\hat{R}_t^k(s, \mathbf{a}) + \gamma_t [r^k(s, \mathbf{a}) - V_\alpha^k((\pi_t)_*) + Val(\hat{R}_t^k(s'), (\pi_t)_*)], \quad (8)$$

$$V_\alpha^k(\pi_{(t+1)*}) = V_\alpha^k((\pi_t)_*) + \theta_t Val [\hat{R}_t^k(s'), (\pi_t)_*], \quad (9)$$

where γ_t and θ_t are the learning parameters.

4 Finding NE of Matrix Games

In this section we present a new approach to obtain Nash equilibrium of n -player matrix games. Let $R^k(\mathbf{a})$ denote the reward matrix of the k^{th} player of which $r^k(a^1, \dots, a^n)$ are the matrix elements. Define the value of an action a^k to player k as

$$Val[R^k(a^k)] = \sum_{\{a^1, \dots, a^n \setminus a^k\}} p(a^{-k}, a^k) r^k(a^1, \dots, a^k, \dots, a^n), \quad (10)$$

where $p(a^{-k}, a^k)$ denotes the probability of choice of an action combination a^{-k} by all the players while player k chose action a^k . In decision making problems with a single player (MDPs and SMDPs), there exist optimal *values* for each state-action pair, which determine the optimal action in each state (Puterman [1994]). Drawing an analogy, for matrix games that have multiple players and a single state, we conjecture that there exist optimal *values* for all actions of the players that can yield pure and mixed NE strategies. However, the probabilities ($p(a^{-k}, a^k)$) needed to compute these *values* are impossible to obtain for real life problems without prior knowledge of players' behavior. This situation is similar to those for discounted and average reward stochastic games discussed earlier, where, in the absence of the transition probabilities, a stochastic approximation based learning approach was used in estimating the elements of the equivalent matrices (see (5), (8), and (9)). Therefore, we employ a learning approach to estimate the *values* of the actions as follows. We rewrite (10) as

$$Val[R_{t+1}^k(a^k)] = (1 - \gamma_t)[R_t^k(a^k)] + \gamma_t [r^k(a^1, \dots, a^k, \dots, a^n)]. \quad (11)$$

The algorithm presented below utilizes the value learning scheme (11) to derive pure and mixed NE strategies for n -player matrix games.

4.1 A Value Iteration Algorithm for n -Player Matrix Games

We assume that the game has n -players and each player k has a set of A_k action choices. Hence, n different reward matrices of size $|A_1| \times |A_2| \times \dots \times |A_n|$ are available.

The Algorithm

1. Eliminate rows and columns of the matrices associated with the dominated strategies. A dominated strategy is one that will never be adopted by a rational player irrespective of the choices of other players. A strategy $a \in A_k$ for player k is said to be dominated if $r(k, a, a^{-k}) \leq r(k, \bar{a}, a^{-k})$, where $\bar{a} \in A_k \setminus a$ and a^{-k} denotes the actions of all other players.
2. Let iteration count $t = 0$. Initialize the R -values for all player and action combinations $R(k, a)$ to an identical small positive value (say, 0.001). Also initialize the learning parameter γ_0 , exploration parameter ϕ_0 , and parameters γ_τ, ϕ_τ needed to obtain suitable decay rates of learning and exploration. Let *Maxsteps* denote the maximum iteration count.

3. If $t \leq \text{Maxsteps}$, continue learning of the R -values through the following steps.

(a) **Action Selection:**

Greedy action selection for pure strategy Nash equilibrium:

Each player k , with probability $(1 - \phi_t)$, chooses a greedy action for which $R^k(a) \geq R(k, \bar{a})$. A tie is broken arbitrarily. With probability ϕ_t , the player chooses an exploratory action from the remaining elements of A_k (excluding the greedy action), where each exploratory action is chosen with equal probability.

Probabilistic action selection for mixed strategy Nash equilibrium:

Compute the probabilities for the action choices using the ratio of R -values at iteration t as follows. For each player k , the probability of choosing the action $a \in A_k$ is given by $\frac{R(k,a)}{\sum_{b \in A_k} R(k,b)}$.

(b) **R -Value Updating:** Update the specific R -values for each player k corresponding to the chosen action a using the learning scheme given below.

$$R_{t+1}(k, a) \leftarrow (1 - \gamma_t)R_t(k, a) + \gamma_t (r(k, \mathbf{a})), \quad (12)$$

where \mathbf{a} denotes the action combination chosen by players.

(c) Set $t \leftarrow t + 1$.

(d) Update the learning parameters γ_t and exploration parameter ϕ_t following the DCM scheme given below (Darken et al. [1992]):

$$\Theta_t = \left(\frac{\Theta_0}{1 + u} \right), \quad \text{where } u = \left(\frac{t^2}{\Theta_\tau + t} \right), \quad (13)$$

where Θ_0 denotes the initial value of a learning/exploration rate, and Θ_τ is a large value (e.g., 10^6) chosen to obtain a suitable decay rate for the learning/exploration parameters. Exploration rate generally has a large starting value (e.g., 0.8) and a quicker decay, whereas learning rate has a small starting value (e.g., 0.01) and very slow decay rate. Exact choice of these values depends on the application (Das et al. [1999], Gosavi et al. [2002]).

(e) If $t < \text{MaxSteps}$, go to Step 3(a), else go to Step 4.

4. **Equilibrium Strategy Determination:** From the final set of R -values, obtain the equilibrium strategies as follows.

Pure strategy equilibrium: For each player k , the pure strategy is action a for which $R(k, a) \geq \max_{b \in A_k} R(k, b)$. The pure strategies for all players combined constitute the pure strategy equilibrium.

Mixed strategy equilibrium: For each player k , the mixed strategy is to select each action $a \in A_k$ with probability $\frac{R(k,a)}{\sum_{b \in A_k} R(k,b)}$.

5 Numerical Evaluation of the Learning Algorithm

In this section we first present results from an extensive comparative numerical study conducted with an objective to establish the ability of the RL algorithm to obtain Nash equilibrium for n -player matrix games. For this purpose, sixteen matrix game examples with known Nash equilibria were solved by using both an openly available software (Gambit) and the RL algorithm. To further evaluate the RL algorithm, we studied a matrix game arising from a deregulated power market application that was examined by Berry et al. [1999].

5.1 Matrix Games with Known Equilibria

Matrix games that were studied consisted of up to four players and sixty four different action choices. Ten out of the sixteen examples that have pure strategy Nash equilibria were solved using the variant of the RL algorithm that seeks a pure strategy. The remaining six games were solved using the mixed strategy version of the RL algorithm.

Table 1: Sample matrix games with pure strategy Nash equilibria.

Matrix Games Studied for Pure Strategy Nash Equilibrium																					
Game 1	<table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>1</td><td>3, 3</td><td>0, 5</td></tr> <tr><td>2</td><td>5, 0</td><td>1, 1</td></tr> </table>				1	2	1	3, 3	0, 5	2	5, 0	1, 1	Game 4: GAMBIT © wink3 (2P: 3X3)								
		1	2																		
1	3, 3	0, 5																			
2	5, 0	1, 1																			
			Game 5: GAMBIT © perfect1 (2P: 3X3)																		
Game 2	<table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>1</td><td>3, 3</td><td>0, 5</td></tr> <tr><td>2</td><td>5, 0</td><td>1, 1</td></tr> </table>				1	2	1	3, 3	0, 5	2	5, 0	1, 1	Game 6: GAMBIT © 2x2x2 (3P: 2X2X2)								
		1	2																		
	1	3, 3	0, 5																		
2	5, 0	1, 1																			
			Game 7: GAMBIT © 8x2x2 (3P: 8X2X2)																		
			Game 8: GAMBIT © Palf (2P: 3 X 3)																		
Game 3	<table border="1"> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>1, 4</td><td>2, 2</td><td>2, 3</td></tr> <tr><td>2</td><td>3, 1</td><td>1, 5</td><td>4, 1</td></tr> <tr><td>3</td><td>2, 0</td><td>3, 4</td><td>1, 2</td></tr> </table>				1	2	3	1	1, 4	2, 2	2, 3	2	3, 1	1, 5	4, 1	3	2, 0	3, 4	1, 2	Game 9: GAMBIT © 2x2x2x2 (4P: 2X2X2X2)	
		1	2	3																	
	1	1, 4	2, 2	2, 3																	
2	3, 1	1, 5	4, 1																		
3	2, 0	3, 4	1, 2																		
			Game 10: GAMBIT © 8x8 (2P: 8 X 8)																		

Table 1 summarizes the matrix games specifying the number of players and their available action choices. Some of these problems are adopted from Gambit library of matrix games, for which the file names used in Gambit are used as identifiers. The Nash equilibrium solutions obtained by both Gambit and RL algorithm are summarized in Table 2. The following observations can be made from the results. For all ten games, the RL algorithm found a Nash

Table 2: Pure Strategy Nash Equilibrium Results

Matrix Game #	GAMBIT solution		Learning Algorithm Solution		
	Equilibria	Equilibrium Value	Equilibrium	Equilibrium Value	Convergence time in Seconds
Game 1	(2, 2)	(1, 1)	(2, 2)	(1, 1)	10
Game 2	(2, 2)	(1, 1)	(2, 2)	(1, 1)	14
Game 3	(3, 2)	(3, 4)	(3, 2)	(3, 4)	27
Game 4	(1, 2)	(3, 4)	(1, 2)	(3, 4)	27
	(2, 1)	(1, 2)			
Game 5	(1, 1)	(3, 1)	(2, 2)	(2, 2)	40
	(2, 2)	(2, 2)			
	(3, 3)	(3, 1)			
Game 6	(1, 1, 1)	(9, 12, 8)	(1, 1, 1)	(9, 12, 8)	42
	(2, 2, 1)	(9, 2, 8)			
	(1, 2, 2)	(3, 6, 4)			
	(2, 1, 2)	(3, 6, 4)			
Game 7	(1, 2, 1)	(7.0, 4.6, 7.5)	(2, 1, 7)	(5.7, 4.5, 7.5)	11
	(2, 1, 7)	(5.7, 4.5, 7.5)			
Game 8	(3, 3)	(3, 2)	(3, 3)	(3, 2)	10
Game 9	(1, 1, 2, 1)	(2.3, 3.2, 3.0, 2.4)	(2, 2, 1, 1)	(4.5, 4.9, 5.7, 4.5)	39
	(2, 2, 1, 1)	(4.5, 4.9, 5.7, 4.5)			
Game 10	(4, 6)	(7.5, 7.9)	(4, 6)	(7.5, 7.9)	12
	(6, 3)	(4.9, 5.7)			
	(7, 2)	(5.6, 5.6)			

equilibrium which coincided with a Gambit solution. It may be noted that Gambit obtained multiple pure strategy NE for six out of the ten games, and the equilibrium obtained by the RL algorithm for these games (except in Game #7) are the ones with highest player rewards. Though a formal mathematical proof will be required to support this observation, we believe that, since the RL algorithm learns the values for the actions and chooses actions based on these values, the solution tends to converge to the NE with the highest player rewards.

Table 2 also presents the convergence time of the RL algorithm which was run for 10,000 iterations for all the games on a computer with a 1.6 GHz Pentium M processor. However, an accurate assessment of the convergence time will require further optimization of the learning parameters of the algorithm, which could be problem dependent. For example, many of the games that are presented in the table converged much sooner than 10,000 iterations. Hence, the convergence times presented here are intended only to provide a general idea of the computational efforts required by the algorithm.

Table 3 presents the comparison of mixed strategies obtained by Gambit and the RL algorithm for six matrix games. Though Gambit found multiple mixed NE for most of these problems, for fairness of comparison, only those NE with maximum player rewards are presented in the table. As evident from the table, though the mixed strategies obtained by the RL algorithm are different from the NE obtained by Gambit, player rewards from the RL algorithm in almost all of the games are either better or comparable. It can also be seen from the table that even when the mixed strategy RL algorithm is used, it could yield a pure strategy (if one exists) that provides better rewards for the players (see Games 4 and 5). In Game 6, where the two players have 64 actions each, the mixed strategies for both players have large

Table 3: Mixed Strategy Equilibrium Results

		Mixed Strategy Equilibrium						
		Gambit Solution			Learning Approach			
Game 1		0.33	0.33	0.33	0.55	0	0.45	
	0.33	0, 0	5, 4	4, 5	0	0, 0	5, 4	4, 5
	0.33	4, 5	0, 0	5, 4	1	4, 5	0, 0	5, 4
	0.33	5, 4	4, 5	0, 0	0	5, 4	4, 5	0, 0
		Value to player 1= 2.94 Value to player 2= 2.94			Value to player 1= 4.45 Value to player 2= 4.55			
Game 2		0	1	0	0.5	0.48	0.2	
	0.5	1, 3	3, 4	4, 2	0.5	1, 3	3, 4	4, 2
	0.5	1, 2	3, 1	2, 1	0.5	1, 2	3, 1	2, 1
	0	0, 4	2, 3	3, 3	0	0, 4	2, 3	3, 3
		Value to player 1= 3 Value to player 2= 2.5			Value to player 1= 2.54 Value to player 2= 2.75			
Game 3		0.57	0.43	0.5	0.5	0	0.5	
	0.5	3, 1	0, 0	0, 1	0	3, 1	0, 0	0, 1
	0.5	1.5, 1	2, 2	1.5, 1	0.5	1.5, 1	2, 2	1.5, 1
	0	0, 1	0, 0	3, 1	0.5	0, 1	0, 0	3, 1
		Value to player 1= 1.7 Value to player 2= 1			Value to player 1= 1.5 Value to player 2= 1			
In the Games 4 and 5 Mixed Strategy Learning Algorithm obtains a pure strategy solution								
Game 4		0.82	0.18		0	1		
	0.67	0, 0	9, 3		1	0, 0	9, 3	
	0.33	2, 6	0, 0		0	2, 6	0, 0	
		Value to player 1= 1.64 Value to player 2= 2			Value to player 1= 9 Value to player 2= 3			
Game 5		0.833	0.167		1	0		
	0.833	70, 70	10, 60		1	70, 70	10, 60	
	0.167	60, 10	60, 60		0	60, 10	60, 60	
		Value to player 1= 60 Value to player 2= 60			Value to player 1= 70 Value to player 2= 70			
Game 6	GAMBIT © bayes2a			GAMBIT © bayes2a				
	2 Players 64 actions each			2 Players 64 actions each				
	Value to player 1: 8.9 Value to player 2: 6.6			Value to player 1: 8.3 Value to Player 2: 8.1				

support sets and thus could not be presented in the table. Therefore, we chose to present only the player rewards as means for comparison. In the next subsection, we present a matrix game example from a real life marketplace that is settled through multiparty competition on a periodic basis.

5.2 Strategic Bidding in Electric Power Markets: A Matrix Game Example

In deregulated electric power markets, like in PJM (Pennsylvania-Jersey-Maryland), New York, New England, and Texas, power is traded in long term bilateral market, day ahead market, and spot market. The generators and retailers compete in the market by strategically bidding for price and quantity of power traded in order to maximize profits. The market is settled by an independent system operator, who matches the supply and demand and satisfies the network constraints while maximizing social welfare (total benefit minus total cost). This settlement yields price and quantity allocations at all the network nodes. The generators strategize to raise their prices above the marginal (base) costs, while the retailers' strategies are aimed at maintaining prices close to the marginal costs. The ability of the generators to maintain prices above the marginal costs for a sustained period of time is defined as *market power*. A market is said to be competitive when the prices are at or near the marginal costs, which is one of the primary objectives of a deregulated electricity market design. A day ahead power market can be modeled as a repeated n -player matrix game, of which the reward matrices can be constructed using the producer surplus (for generators) and consumer surplus (for retailers). Definitions of producer and consumer surplus can be found in (Berry et al. [1999]). Readers interested in more details on power market operations are referred to (Stoft [2002]).

We consider a four bus (two generators and two retailers) power network as shown in Figure 1, which was studied in (Berry et al. [1999]). The supply function bids of the generators at nodes A and B and the demand functions of the retailers at nodes C and D are as follows: $p_{S_1} = a_1 + m_1 q_1$, $p_{S_2} = a_2 + m_2 q_2$, $p_{D_1} = 100 - 0.52d_1$, $p_{D_2} = 100 - 0.65d_2$, where q_1 and q_2 are the quantities (in megawatt-hour, MWh) produced by generators S_1 and S_2 respectively, and d_1 and d_2 are the quantities demanded by the retailers D_1 and D_2 respectively. The supply function has two strategic bid parameters (intercept a in \$/MWh and slope m) that the generators manipulate to maximize their profits. Demand side bidding by the retailers is not considered and hence the demand function parameters are maintained constant at their base values. As in (Berry et al. [1999]), the reactances are considered to be the same on all lines.

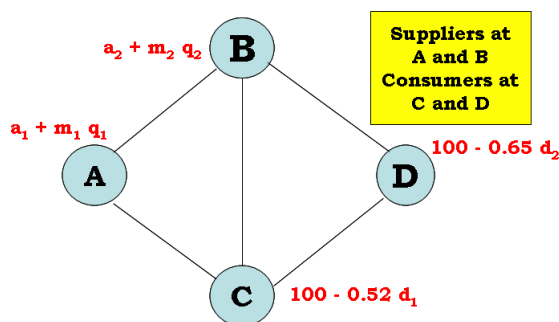


Figure 1: 4-Bus Power Network

In (Berry et al. [1999]), the effects of strategic bidding are studied by imposing transmission constraints on lines AC and BD (one at a time) resulting in network congestion. Nash equilibria for both slope-only and intercept-only bidding scenarios for each of the transmission constrained cases (AC and BD) are separately examined.

Berry et al. [1999] used an iterative algorithm to obtain NE of the above game. The algorithm involves solving the ISO's problem for a series of bid options of a generator, while holding the bids of the other generator constant. The bid option that produces maximum profit is then fixed, and the same procedure is repeated for the other generator. This process is repeated until neither generator has an alternative bid to further improve profit. The matrix game approach developed in this paper differs from the above approach in that all generators select actions simultaneously without any knowledge of the others actions.

In order to apply the learning algorithm, as a first step, the reward matrices for the generators are constructed. To accomplish this, the feasible range of the bid parameters are suitably discretized (which dictate the size of the reward matrices), and the rewards for each combination of the generators bids are calculated. It may be noted that generator reward is a function of the nodal prices and quantities, which are obtained by solving a social welfare maximization problem. Details of the mathematical formulation can be found in (Berry et al. [1999]). The feasible ranges of slope and intercept parameters are discretized to 250 values giving matrix sizes of 250×250 . In particular, the slope parameter ranged from 0.35 to 2.85 for S_1 and 0.45 to 2.95 for S_2 , both in steps of 0.01. The intercept bid parameter for both generators S_1 and S_2 ranged from 10 \$/MWh to 260 \$/MWh with a step length of 1 unit. The solution of the social welfare problem and calculation of the generator rewards for all the above bid combinations are accomplished using GAMS software. The results from (Berry et al. [1999]) and those from the learning algorithm are presented in Table 4. It can be seen from the table that the learning algorithm obtains better or comparable profits for both generators in all cases.

We also extend the numerical experimentation by allowing generators to bid for both slope and intercept together, instead of bidding for one parameter at a time as in (Berry et al. [1999]). The bid parameters in this experiment are discretized as follows. The slope is varied in twenty five steps of 0.1 for both generators ranging from 0.35 to 2.85 for S_1 and 0.45 to 2.95 for S_2 . The intercept is varied in twenty five steps of 3 ranging from 10 \$/MWh to 85 \$/MWh. Hence, each generator has $25 \times 25 = 625$ action choices and the resulting reward matrices are of size 625×625 . The RL algorithm is run for 500,000 iterations, which took 770 seconds on a computer with a 2 GHz Pentium IV processor. As shown in Table 4, in the AC-congestion case, bidding in both slopes and intercepts lead to similar profits as in the cases of one parameter at a time bidding. Whereas, in the case of BD-congestion, the profits obtained by the players through joint bidding is much higher than bidding one parameter at a time.

Table 4: Results from the Study of 4-Bus Power Network

		Bidding with Slopes while Intercepts held constant for both players at 10, 10		Biding with Intercepts while Slopes held constant for both players at 0.35, 0.45		
		Berry et al.	Learning Algorithm	Berry et al.	Learning Algorithm	
Profit Gen A (\$)		1343	1457	1087	1136	} Line AC constrained at 30 MW
Profit Gen B (\$)		3345	3284	3373	3338	
NE Bids		(2.42, 0.72)	(2.26, 0.78)	(60, 25)	(59, 26)	
Profit Gen A (\$)		2478	2640	2115	2156	} Line BD constrained at 30 MW
Profit Gen B (\$)		1374	1422	972	1044	
NE Bids		(0.77, 1.2)	(0.8, 1.36)	(22, 28)	(28, 29)	
Bidding with both Slopes and Intercepts NE obtained using Learning Algorithm						
		Line AC constrained at 30 MW		Line BD constrained at 30 MW		
Profit Gen A (\$)		1518		2851		
Profit Gen B (\$)		3266		1680		
NE Bids		Gen A: (61, 0.35), Gen B: (37, 0.45)		Gen A: (27, 0.69), Gen B: (34, 0.93)		

6 Concluding Remarks

Though the internet era has provided the technological infrastructure necessary to invigorate market competition, lack of commensurate advancements in computational algorithms to solve multiplayer games has been a limiting factor. Meteoric rise in computing power via tera and peta scale computing (made possible by efficient harnessing of cluster computing) has created an opportunity to break through perceived computational barriers of state space explosion. This paper presents a new computational approach to find Nash equilibrium of multiplayer matrix games. The approach is founded on the value function learning strategy that is being successfully used in solving large scale decision making problems modeled as Markov and semi-Markov decision processes. In the wake of recent studies that link a large class of stochastic games to matrix games, our solution approach stands to impact a broad range of decision making problems.

The comparative numerical results presented for a large number of games help to demonstrate the validity of our conjecture (in Section 4) on value function guided NE determination. The electric power market problem serves as an excellent example of real life application of matrix games. Such games, outcomes of which determine the nature of power transactions, occur at various frequencies ranging from once a year as in bilateral markets to every hour in the spot market. Hence, the ability to accurately obtain NE for matrix games allows for better assessment of market performance and efficient market design, which translate to stable power market operations with limited price spikes. Solutions of relatively large matrix games (of size 625×625) resulting from the sample power network problem indicate the algorithms

potential. Though the numerical results are promising and encourage further exploration of our algorithms performance, a theoretical proof of convergence and optimality is required. We believe that such a proof can be constructed following the logic used in (Li et al. [2007]), and we are currently working on developing such a proof.

Real life applications of matrix games tend to have a limited number of players competing to maximize their individual profits. Such oligopolistic structures of most contemporary markets can be attributed to extensive market segmentation. Some examples of real life matrix games include retail sales, home and auto insurance, mortgage lending, service industries like airlines, hotels, and entertainments.

Acknowledgement: This research was supported in part by National Science Foundation grant # ECS 0400268.

References

- John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.
- J. Li, T. K. Das, and K. Ramachandran. "Learning nash equilibrium policies in average reward irreducible stochastic games". *Working paper, Department of Industrial and Management Systems Engineering, Univ. of South Florida*, 2007.
- J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and algorithm. In *Proc. 15th International Conf. on Machine Learning*, Madison, WI, 1998. IMLC-98.
- V. S. Borkar. Reinforcement learning in Markovian evolutionary games. *Advances in Complex Systems*, 5(1):55–72, 2002.
- J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer Verlag, New York, 1997.
- R. McKelvey and A. McLennan. Computation of equilibria in finite games. In J. Rust H. Amman, D. Kendrick, editor, *Handbook of Computational Economics*, pages 87–142. Elsevier, 1996.
- C. E. Lemke and J. T. Howson Jr. Equilibrium points of bimatrix games. *SIAM J. of App. Math.*, 12:413–423, 1964.
- J. Rosenmuller. On the generalization of the lemke-howson algorithm to noncooperative n-person games. *SIAM J. of App. Math.*, 21(1):73–79, 1971.
- R. Wilson. Computing equilibria of n-person games. *SIAM Applied Math*, 21:80–87, 1971.
- L. Mathiesen. Computational experience in solving equilibrium models by a sequence of linear complementarity problems. *Operations Research*, 33(6):1225–1250, 1985.

- P. T. Harker and J. S. Pang. Finite dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming*, 48():161–220, 1990. .
- R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 0.2007.01.30. 2007.
- K. H. Lee and R. Baldick. Solving three-player games by the matrix approach with an application to an electric power market. *IEEE Transactions on Power Systems*, 18(4):166–172, 2003.
- Benjamin F. Hobbs. Linear complementarity models of Nash-Cournot competition in bilateral and POOLCO power markets. *IEEE Transactions On Power Systems*, 16(2):194–202, May 2001.
- B. F. Hobbs, B. M. Carolyn, and J. S. Pang. "Strategic gaming analysis for electric power systems: An MPEC approach". *IEEE Tran. Power Syst.*, 15(2):638–645, 2000.
- J. Yao, S. Oren, and I. Adler. Computing cournot equilibria in two settlement electricity markets with transmission constraints. In *Proceedings of 37th Hawaii international conference on system sciences*, 2003.
- T. C. Price. Using co-evolutionary programming to simulate strategic behavior in markets. *J. Evol. Econ.*, 7:219–254, 1997.
- L. B. Cunningham, R. Baldick, and M. L. Baughman. An empirical study of applied game theory: Transmission constrained cournot behavior. *IEEE Transactions on Power Systems*, 17(1):166–172, 2002.
- R. W. Ferrero, M. Shahidehpour, and V. C. Ramesh. Transaction analysis in deregulated power systems using game theory. *IEEE Transactions on Power Systems*, 12(3):1340–1347, 1999.
- S. Stoff. Using game theory to study market power in simple networks. *IEEE Tutorial on Game Theory in Electric Power Markets*, pages 33 – 40, 1999.
- Z. Luo, J. S. Pang, and D. Ralph. In *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Boston, 1996.
- C. L. Su. Equilibrium problems with equilibrium constraints: Stationarities, algorithms, and applications. *Dissertation from the Department of Management Sciences, Stanford University*, 2005.
- D. Ralph and Y. Smeers. Epecs as models for electricity markets. *Power Systems Conference and Exposition, Atlanta, GA*, 2006.
- J. B. Cardell, C. C. Hitt, and W. W. Hogan. Market power and strategic interaction in electricity networks. *Resource and Energy Economics*, 19:109137, 1997.

- T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchallick. Solving semi-Markov decision problems using average reward reinforcement learning. *Management Science*, 45(4), 1999.
- A. Gosavi, N. Bandla, and T. K. Das. A reinforcement learning approach to airline seat allocation for multiple fare classes with overbooking. *IIE transactions, Special issue on advances on large-scale optimization for logistics, production, and manufacturing systems*, 34(9):729–742, 2002.
- A. Gosavi. Reinforcement learning for long-run average cost. *European Journal of Operational Research*, 155:654–674, 2004.
- M.L. Puterman. *Markov Decision Processes*. John Wiley and Sons, New York Chichester Brisbane Toronto Singapore, 1994.
- C. Darken, J. Chang, and J. Moody. "learning rate schedules for faster stochastic gradient search". In D.A. White and D.A. Sofge, editors, *Neural Networks for Signal Processing 2 - Proceedings of the 1992 IEEE Workshop*. IEEE Press, Piscataway, NJ, 1992.
- C. A. Berry, B. F. Hobbs, W. A. Meroney, R. P. O'Neill, and W. R. Stewart Jr. Understanding how market power can arise in network competition: A game theoretic approach. *Utilities Policy*, 8:139–158, 1999.
- S. Stoft. *Power System Economics*. IEEE Press, New Jersey, 2002.