

# Learning Equilibrium in Resource Selection Games

Itai Ashlagi and Dov Monderer and Moshe Tennenholtz

Technion–Israel Institute of Technology  
Faculty of Industrial Engineering and Management  
Haifa 32000, Israel  
ashlagii@tx.technion.ac.il  
{dov,moshet}@ie.technion.ac.il

## Abstract

We consider a resource selection game with incomplete information about the resource-cost functions. All the players know is the set of players, an upper bound on the possible costs, and that the cost functions are positive and nondecreasing. The game is played repeatedly and after every stage each player observes her cost, and the actions of all players. For every  $\epsilon > 0$  we prove the existence of a learning  $\epsilon$ -equilibrium, which is a profile of algorithms, one for each player such that a unilateral deviation of a player is, up to  $\epsilon$  not beneficial for her regardless of the actual cost functions. Furthermore, the learning equilibrium yields an optimal social cost.

## 1. Introduction

In a *resource selection game* there is a set of  $K$  resources, and a set of  $n$  players. Each player has to select a resource from among the set of resources. Each resource  $k$  is associated with a cost function, where  $c_k(l)$  is the cost suffered by any player who has selected resource  $k$ , if there are  $l$  players who have selected this resource. Resource selection games are a particular and useful type of congestion games (Rosenthal 1973; Monderer & Shapley 1996),<sup>1</sup> and are central to work in CS/AI, Game Theory, Operations Research, and Economics. In particular, resource selection games have been extensively discussed in the price of anarchy literature, e.g., (Koutsoupias & Papadimitriou 1999). Most of the work on resource selection games assumes that all parameters of the game are commonly known, or at least that there is a commonly known Bayesian information regarding the unknown parameters (see (Gairing, Monien, & Tiemann 2005; Garg & Narahari 2005)). However, in many situations, the game, and in particular the resource cost functions are unknown. When the game under discussion is played only once, one has to analyze it using solution concepts for games with incomplete information without probabilistic information (known also as pre-Bayesian games).<sup>2</sup> However,

if the game is played repeatedly the players may learn about the resource cost functions, by observing the feedback for actions they performed in the past. This brings us to the study of reinforcement learning in (initially unknown) repeated resource selection games.<sup>3</sup>

Learning in the context of multi-agent interactions has attracted the attention of researchers in psychology, economics, artificial intelligence, and related fields for quite some time ((Littman 1994; Hu & Wellman 1998; Brafman & Tennenholtz 2002; Bowling & Veloso 2001; Conitzer & Sandholm 2003; Greenwald, Hall, & Serrano 2002; Erev & Roth 1998; Fudenberg & Levine 1998)). Much of this work uses repeated games as models of such interactions. Initially all that is known is that the game played is taken from a set of possible games. Then, after each iteration the players receive some information about the game history. In this paper we will assume *partial monitoring*: each player is able to observe the actions selected by all players in the previous iteration, but only its own payoff in that iteration. There are various definitions of what would define a satisfactory learning process. In this paper we adopt a most desirable and highly demanding requirement: we wish the players' learning algorithm to conform a *learning equilibrium* (Brafman & Tennenholtz 2004; 2005; Ashlagi, Monderer, & Tennenholtz 2006; Monderer & Tennenholtz 2007) leading to optimal social cost.

A learning equilibrium determines learning algorithms for a given class of games, which capture the following: it is not beneficial for any player to deviate from its algorithm assuming that the other players stick to their algorithms, *regardless* of the game being played, as long as this game is taken from the prescribed class of games.<sup>4</sup> If we add the requirement that, if adopted by the players, the learning algorithms will yield optimal social cost, then we get a very desirable outcome, which may be very hard to obtain. Nevertheless, previous studies have shown some positive results. In this work we contribute to both the theory of learning

Copyright © 2008, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup>In a general congestion game every player can choose a set of resources.

<sup>2</sup>Such an analysis was done in (Ashlagi, Monderer, & Tennenholtz 2006) in a resource selection game, in which the lack of information is about the number of participants.

<sup>3</sup>This study should be distinguished from the study of best-response (or even better-response) dynamics that is known to converge to equilibrium in congestion games with complete information (see (Monderer & Shapley 1996)).

<sup>4</sup>The concept of learning equilibrium has been generalized in (Ashlagi, Monderer, & Tennenholtz 2006) to allow each player to receive an initial private signal regarding the true game.

equilibrium and to the theory of resource selection games by proving, for every  $\epsilon > 0$  the existence of a learning  $\epsilon$ -equilibrium for arbitrary classes of resource selection games with non-decreasing and positive cost functions, where in a learning  $\epsilon$ -equilibrium a deviation is not profitable if it decreases cost by no more than  $\epsilon$ .

Our proof of the existence of a learning  $\epsilon$ -equilibrium for resource selection games is constructive. We show an efficient algorithm, LE-RS, which determines a learning  $\epsilon$ -equilibrium. In addition, if all players follow the LE-RS algorithm, then after a finite number of stages, which is polynomial in the number of resources and in the number of players, the players will obtain at every stage the optimal social cost of the underlying (initially unknown) game. Moreover, the punishment for a deviating player will also be materialized after polynomially many iterations. This result complements results obtained in (Brafman & Tennenholtz 2005) for learning  $\epsilon$ -equilibrium in general 2-person symmetric games, which can be extended to symmetric  $n$ -person games only when we allow any group of players to have a private channel for coordinating their actions; the LE-RS does not require such strong abilities, which typically might not exist.

## 2. Background-Learning equilibrium in repeated games.

A *game in strategic form* (or, for short, a *game*) consists of a finite set of players,  $N$ , and for every player  $i \in N$ , an action set  $A_i$  and a cost function  $c^i : \mathbf{A} \rightarrow \mathbb{R}$ , where  $\mathbf{A} = \times_{i \in N} A_i$  is the set of action profiles. For convenience,  $N = \{1, 2, \dots, n\}$ , where  $n$  is the number of players. An action profile  $\mathbf{a} \in \mathbf{A}$  is a *Nash equilibrium* if for every player  $i$ ,  $c^i(a_i, \mathbf{a}_{-i}) \leq c^i(b_i, \mathbf{a}_{-i})$  for every  $b_i \in A_i$ , where, as usual  $\mathbf{a}_{-i}$  is the  $(n - 1)$  action profile obtained from  $\mathbf{a}$  by removing player  $i$ 's action. The *optimal social cost* in  $G$  is denoted by  $S(G)$ , and it is defined to be  $\min \sum_{i=1}^n c^i(\mathbf{a})$ , where the min ranges over all action profiles  $\mathbf{a} \in \mathbf{A}$ .

A game is *finite* if every action set is a finite game. In a finite game, a mixed action for player  $i$  is a probability distribution over  $A_i$ . The set of mixed actions for player  $i$  is denoted by  $A_i^m$ . The corresponding set of mixed action profiles is denoted by  $\mathbf{A}^m = \times_{i \in N} A_i^m$ . A mixed action that assigns a probability 1 to a single action is called a *pure action*. In this paper we identify pure actions with actions. Hence, we consider  $A_i$  as a subset of  $A_i^m$ , and  $\mathbf{A}$  as a subset of  $\mathbf{A}^m$ . When dealing with mixed actions, the cost function for player  $i$ ,  $c^i$  is naturally extended to the expected cost function, which, with an abuse of notations, we also denote by  $c^i$ . That is,  $c^i : \mathbf{A}^m \rightarrow \mathbb{R}$ .

In a repeated game with complete information, the players play repeatedly a commonly known stage game  $G$ . There are infinite number of stages, and in each stage, each player chooses a mixed action from the set of her possible mixed actions in  $G$ , pays her cost, and move on to the next stage. After each stage every player observes the actual actions chosen by all players at this stage. In a repeated game with incomplete information there exists a set  $\mathcal{M}$  of stage games. One of the games in  $\mathcal{M}$  is played repeatedly,

and the players observe the actual actions after each stage. However the players do not know initially, which game is played. All they know is the set of players and the sets of actions in the chosen game.<sup>5</sup> Every player may partially learn the costs in the true game through additional feedback, which in this paper is defined to be her cost. Hence, after each stage every player knows her actual cost. A repeated game with this particular feedback is called a *repeated game with incomplete information with partial monitoring*. Hence, a strategy for player  $i$  in the repeated game with incomplete information determined by  $\mathcal{M}$  is a function  $f^i$  (depending on the set of players and their action sets) that assigns a mixed action of  $i$  to every pair of finite sequences,  $[(\mathbf{a}(1), \mathbf{a}(2), \dots, \mathbf{a}(t)), (c^i(1), c^i(2), \dots, c^i(t))]$ . One sequence is the sequence of action profiles, which is called an history, and the other sequence consists of the costs. That is, at every stage  $j \leq t$ , the players chose the profile of actions  $a(j)$ , and player  $i$  paid  $c^i(j)$ . Note that  $a(j) = (a_1(j), \dots, a_n(j))$ , where  $a_i(j)$  is the realization of the randomized strategy used by  $i$  at stage  $j$ . The set of all strategies of  $i$  at  $\mathcal{M}$  is denoted by  $\Sigma_i$ , and the set of strategy profiles is denoted by  $\Sigma$ . Let  $\mathbf{f} = (f^1, f^2, \dots, f^n)$  be a strategy profile in  $\Sigma$ . For every  $G \in \mathcal{M}$   $\mathbf{f}$  determines a probability distribution over the set of histories of action profiles  $\mathbf{a}(1), \mathbf{a}(2), \dots$ . We denote the expected average cost of player  $i$  up (and including) to stage  $t$  determined by this probability distribution by  $C^i(G, \mathbf{f}, t)$ .

**Definition 1** *Let  $\mathcal{M}$  be a repeated game with incomplete information, and let  $\epsilon > 0$ . A strategy profile  $\mathbf{f} = (f_1, \dots, f_n)$  is an learning- $\epsilon$ -equilibrium in  $\mathcal{M}$  if there exists  $T \geq 1$  such that for every  $t \geq T$ , for every  $G \in \mathcal{M}$ , and for every player  $i$ ,  $C^i(G, \mathbf{f}, t) \leq C^i(G, (g_i, \mathbf{f}_{-i}), t) + \epsilon$  for every  $g_i \in \Sigma_i$ .  $\mathbf{f}$  is socially optimal learning- $\epsilon$ -equilibrium if, in addition,  $\sum_{i=1}^n C^i(G, \mathbf{f}, t) \leq S(G) + \epsilon$  for every  $G \in \mathcal{M}$ , and for every  $t \geq T$ .*

## 3. Resource Selection Games

A resource selection game is a tuple  $G = (N, R, (c_k)_{k \in R})$ , where  $N = \{1, \dots, n\}$  is a set of players,  $R = \{1, \dots, K\}$  is a set of resources and for every  $k \in R$ ,  $c_k : \{1, \dots, n\} \rightarrow \mathbb{R}_+$  is resource  $k$ 's cost function. The cost incurred by a player who chooses resource  $k$  is  $c_k(l)$ , where  $l$  is the number of players that choose  $k$ . We assume the resource cost functions are non-decreasing and positive. When every player  $i$  chooses a resource  $z_i \in R$ , an action profile  $\mathbf{z} \in \mathbf{Z} = R^N$  is generated. We denote by  $\sigma_k(\mathbf{z})$  the number of players that chose  $k$  in  $\mathbf{z}$ . That is,  $\sigma_k(\mathbf{z}) = |\{j \in N : z_j = k\}|$ . The cost paid by  $i$  when the action profile  $\mathbf{z}$  is selected is denoted by  $c^i(\mathbf{z})$ , that is  $c^i(\mathbf{z}) = c_{z_i}(\sigma_{z_i}(\mathbf{z}))$ .

Resource selection games belong to a larger class of games called congestion games. It is well-known (Rosenthal 1973) that every congestion game possesses a Nash equilibrium. Before we prove the next Lemma we need the following notation: let  $c^{max}(\mathbf{z})$  denote the maximal cost in the action profile  $\mathbf{z}$ . That is  $c^{max}(\mathbf{z}) = \max_{i \in N} c^i(\mathbf{z})$ .

<sup>5</sup>In a most general model, which is not discussed at this paper, a player may initially know only her own set of actions.

**Lemma 1** Let  $G$  be a resource selection game. Let  $\mathbf{z}$  and  $\mathbf{y}$  be Nash equilibria. Then  $c^{max}(\mathbf{z}) = c^{max}(\mathbf{y})$ .

*Proof.* Suppose in negation that there exist a couple of equilibria  $\mathbf{z}$  and  $\mathbf{y}$  such that  $c^{max}(\mathbf{z}) > c^{max}(\mathbf{y})$ . Let  $k \in R$  be a resource in which  $c_k(\sigma_k(\mathbf{z})) = c^{max}(\mathbf{z})$ . Since  $c_k(\sigma_k(\mathbf{z})) > c_k(\sigma_k(\mathbf{y}))$ ,  $\sigma_k(\mathbf{z}) > \sigma_k(\mathbf{y})$ . Therefore there exists a resource  $k'$  such that  $\sigma_{k'}(\mathbf{y}) > \sigma_{k'}(\mathbf{z})$ . But then

$$c_{k'}(\sigma_{k'}(\mathbf{z}) + 1) \leq c_{k'}(\sigma_{k'}(\mathbf{y})) < c_k(\sigma_k(\mathbf{z})),$$

which contradicts  $\mathbf{z}$  being in equilibrium.  $\square$

Let  $G$  be a resource selection game. We define  $NE^{max}(G)$  to be the maximal cost in a Nash equilibrium. That is  $NE^{max}(G) = c^{max}(\mathbf{z})$  for an arbitrary equilibrium  $\mathbf{z}$ . By Lemma 1,  $NE^{max}(G)$  is well defined.

We need to explore additional properties of resource selection games. For every action profile  $\mathbf{z}_{-i}$  of all players but  $i$  let  $B_i(\mathbf{z}_{-i})$  be the best-response correspondence of  $i$ . That is, it is the set of resources which yield the minimal cost for  $i$  given that all other players are playing  $\mathbf{z}_{-i}$ ;  $B_i(\mathbf{z}_{-i}) = \{k \in R | c^i(k, \mathbf{z}_{-i}) \leq c^i(k', \mathbf{z}_{-i}) \text{ for every } k' \in R\}$ .

**Observation 1** Let  $G = (N, R, (c_k)_{k \in R})$  be a resource selection game, let  $i \in N$ , and let  $N' = N \setminus \{i\}$ . If  $\mathbf{z}_{-i}$  is a Nash equilibrium in the resource selection game  $G' = (N', R, (c_k)_{k \in R})$ , then for every  $r \in B_i(\mathbf{z}_{-i})$ —the best response correspondence of  $i$  at the game  $G$ ,  $\mathbf{y} = (r, \mathbf{z}_{-i})$  is a Nash equilibrium in  $G$ . Moreover,  $c^i(\mathbf{y}) = c^{max}(\mathbf{y}) = NE^{max}(G)$ .

The following simulation method, which, by Observation (1) yields a Nash equilibrium, will be used in the learning algorithm we present in the next section.

#### NE-SimProcess

*Input:* A resource selection game  $G = (N, R, (c_k)_{k \in R})$ .

*An  $n$ -stage Process:* Every player  $i = 1, \dots, n$ , in this order chooses  $z_i$ , where  $z_i$  is the best response to  $(z_1, \dots, z_{i-1})$  at the resource selection game,  $(\{1, \dots, i\}, R, (c_k)_{k \in R})$ . If there are multiple best responses we assume that every resource has an index, and that  $i$  chooses the one with the lowest index. Observation (1) implies the following observation:

**Observation 2** Let  $\mathbf{z}$  be the Nash equilibrium obtained at the end of the NE-SimProcess. Then,  $c^n(\mathbf{z}) = c^{max}(\mathbf{z})$ .

## 4. A description of the LE-RS algorithm

In this section we present an outline of the LE-RS algorithm, and provide some hints behind the proof that if all players adopt the algorithm, it obtains the desired learning  $\epsilon$ -equilibrium. A more detailed specification of the algorithm and its analysis appear in sections 5 and 6, respectively. The algorithm receives as input the set of players,  $N = \{1, \dots, n\}$ , the set of resources,  $R = \{1, \dots, K\}$ , an upper bound on all costs,  $C_{max}$ , and an  $\epsilon > 0$ . Hence, we define  $\mathcal{M}$  as the set of all vectors  $c = (c_k)_{k \in R}$ , where for each  $k$ ,  $c_k$  is positive and non-decreasing and  $c_k(l) \leq C_{max}$  for every  $1 \leq l \leq K$ .

Before we present the algorithm we need some notations. We say that a resource  $k \in R$  is *known* to a player, or the player knows resource  $k$ , if the player observed the

costs  $c_k(j)$  for  $j = 1, \dots, n - 1$ . We say that a resource  $k \in R$  is *nearly known* to a player, or the player nearly knows resource  $k$ , if the player observed the costs  $c_k(j)$  for  $j = 2, \dots, n - 1$ . In the sequel  $s(i)$  denotes the set of all resources  $k \in R$ , in which player  $i$  hasn't observed  $c_k(1)$ . For every resource  $k \in R$  let  $b(k) = k + 1 \pmod{K}$ .

#### The LE-RS algorithm:

The algorithm is described for  $K \geq 3$ . The case  $K = 2$  will be discussed in the full version.

##### On-path:

At the first stage, "Cooperative Exploring", the players learn all the resource cost functions. This is obtained by defining a pre-determined sequence (of polynomial length) in which each player  $i$  visits each resource  $k$  when there are additional  $l$  players in that resource, for any  $0 \leq l \leq n - 1$ .

After all resource cost functions are learned, the players enter the "Playing Optimal" stage. First an action profile  $\mathbf{z}$  which optimizes the social surplus is chosen (in a way that all players know which one it is). In the first iteration of this stage  $\mathbf{z}$  is played. Then, in every iteration  $j$ , player  $i + 1$  plays the action played by player  $i$  in iteration  $j - 1$ , where player  $n$  plays the action played by player 1.

If the players follow the behavior prescribed in the "Cooperative Exploring" stage, then during the "Playing Optimal" stage a socially optimal action profile is played. The "Playing Optimal" stage is structured such that the actual reward for all players in that stage will be identical. Therefore, after a sufficient (but polynomial) number of iterations each player's average cost is not more than  $\frac{S(G)}{n} + \epsilon$  of the true game  $G \in \mathcal{M}$ .

The above assumes that no player deviates during these stages. The way we deal with such deviation is described in the off-path stages below. Such deviator will be referred to as the *adversary*. Notice that since the on-path stages are fully deterministic, all players will be able to simultaneously identify the adversary, if exist.

##### Off-path:

In all off path stages we assume w.l.o.g. the adversary is player  $n$ . Below, when we use the term player, we will refer only to non-adversarial players.

During the off-path stages, players still need to learn "enough" in order to (eventually) punish the adversary. Since we require that the adversary will be punished after a finite (which we will also show to be polynomial) period of time, the players also need to punish the adversary "quickly". However, since the adversary can prevent a deterministic algorithm from learning some values of the resource cost functions, we use randomization techniques. This will leave us with some small probability of failure. In this case, we move to a "Failure" stage, in which the players perform arbitrary actions.

*The "Exploring with an adversary" stage:* in this stage every player  $i \neq n$  learns  $c_k(l)$  for every resource  $k \in R$  and for every  $l = 2, \dots, n - 1$ , in some fixed order. Hence, there are  $(n - 1)(n - 2)K$  phases during this stage. We denote by  $t(i, k, l)$  the phase when it is player  $i$ 's turn to learn  $c_k(l)$ . At every phase  $t(i, k, l)$  the learning is done as follows:

Player  $i$ , and additional pre-determined  $l - 2$  players select resource  $k$ . An additional player  $i'$  chooses randomly (with uniform distribution) between  $k$  and  $b(k)$ . Note that player  $i$  will learn  $c_k(l)$  if and only if either  $i'$  or  $n$ , and only one of them, selects  $k$ . This gives a probability of at least 0.5 that player  $i$  will learn  $c_k(l)$  for any adversary behavior. Hence,  $i$  will learn  $c_k(l)$  quickly with overwhelming probability. Technically, we bound the number of iterations until such learning occurs; if no such learning occurs we move to the "Failure" stage.

*The "Learn Alone" stage:* in this stage there are  $n$  phases. At each of the first  $n - 1$  phases, a different player  $i = 1, \dots, n - 1$  learns  $c_k(1)$  ( $k \in K$ ) for at least  $K - 1$  resources, i.e. phase  $i$  ends as  $|s(i)| \leq 1$ . Consider the phase in which it is player  $i$ 's turn to reach this goal: Each player  $i' \neq i$  selects resource  $k \notin s(i)$ , and player  $i$  randomizes (with uniform distribution) between a couple of resources  $k_1, k_2 \in s(i)$ . With probability  $\frac{1}{2}$  player  $i$  will learn either  $c_{k_1}(1)$  or  $c_{k_2}(1)$ . This is played repeatedly until player  $i$  is able to learn either  $c_{k_1}(1)$  or  $c_{k_2}(1)$ , which will happen after a short time with overwhelming probability. Similar process is applied until  $s(i) \leq 1$ . As before, we bound the number of iterations until such learning occurs; if no such learning occurs we move to the "Failure" stage.

The purpose of the  $n$ -th phase of this stage, is to ensure that for all players  $i, j$  in which  $|s(i)| = |s(j)| = 1$  we have that  $s(i) = s(j)$ . At this phase, at every iteration, a couple of players  $i$  and  $j$  for which  $|s(i)| = 1$  and  $|s(j)| = 1$ , select the resources  $k_i \in s(i)$  and  $k_j \in s(j)$ , respectively, and all other players select different resources. Notice that the adversary can not simultaneously prevent  $i$  from learning  $c_{k_i}(1)$  and  $j$  from learning  $c_{k_j}(1)$ . At the end of this stage the players move to the "Punishing" stage.

*The "Punishing" stage:*

Notice that in the beginning of this stage there will be at most one resource  $k$  for which some of the players do not know  $c_k(1)$ , while  $c_{k'}(l)$  is known by all players for every  $k' \in K$  and every  $2 \leq l \leq n - 1$ . We distinguish between two cases:

(i) All players know all resources: in this case the players will indefinitely play a Nash equilibrium  $\mathbf{z}$ , determined by the NE-SimProcess, assuming only  $n - 1$  players. The best the adversary can do is to choose a best response  $k' \in B_n(\mathbf{z})$  in every iteration. However, by Observation 1, this will yield him a cost of at least  $NE^{max}(G)$  (where  $G$  is the true game). Hence, deviation will not be beneficial.

(ii) There exists a player  $i$  with  $s(i) = 1$ . Let  $k \in s(i)$  be the only resource for which  $c_k(1)$  is not known by some of the players. In this case, let  $\mathbf{z}$  be the Nash equilibrium of the game with  $n - 1$  players, determined by the NE-SimProcess, assuming  $c_k(1) = 0$ . Notice that  $\sigma_k(\mathbf{z}) \geq 1$ .

If  $\sigma_k(\mathbf{z}) > 1$  then the players will play  $\mathbf{z}$  indefinitely. Notice that when  $\sigma_k(\mathbf{z}) > 1$  then  $\mathbf{z}$  is also an equilibrium in the true game with  $n - 1$  players. Therefore, as before, the best the adversary can do is to choose a best response  $k' \in B_n(\mathbf{z})$  in every iteration, which will yield him a cost of at least  $NE^{max}(G)$  (and therefore deviation will not be beneficial).

If  $\sigma_k(\mathbf{z}) = 1$ : let  $i'$  be the player who selects resource  $k$

in  $\mathbf{z}$ . Each player  $j \neq i'$  will play its action in  $\mathbf{z}$  indefinitely. Player  $i'$  will play  $k$  until he observes  $c_k(1)$  (unless he already observed it in the past). Once  $c_k(1)$  is observed (or was observed in the past) by  $i'$ , player  $i'$  chooses a resource  $k' \in B_{i'}(\mathbf{z}_{-i'})$ , and plays it in every following iteration. Observe that  $(\mathbf{z}_{-i'}, k')$  is a Nash equilibrium in  $G$  when there are only  $n - 1$  players. Once again, the best the adversary can do is to choose a best response against that profile, which will yield him a cost of at least  $NE^{max}(G)$ .

Finally, if the adversary prevents  $i'$  from learning  $c_k(1)$ , then his cost can only increase: if  $k \in B_{i'}(\mathbf{z}_{-i'})$ , then  $(\mathbf{z}_{-i'}, k)$  is a Nash equilibrium of the game with  $n - 1$  players, and therefore by playing  $k$  the adversary will suffer a cost of at least  $NE^{max}(G)$ . Suppose  $k \notin B_{i'}(\mathbf{z}_{-i'})$ . Therefore there exists  $k' \in B_{i'}(\mathbf{z}_{-i'})$  such that  $(\mathbf{z}_{-i'}, k')$  is a Nash equilibrium of the game with  $n - 1$  players. Since  $k' \neq k$   $c_k(1) \geq NE^{max}(G)$  (equality holds if  $k \in B_n(\mathbf{z}_{-i'}, k')$ ). Therefore  $c_k(2) \geq NE^{max}(G)$ , which is the adversary's cost as long as she prevents  $i'$  from learning  $c_k(1)$ .

## 5. The LE-RS algorithm – full specification

Before we present the algorithm recall that  $s(i)$  denotes the set of all resources  $k \in R$ , in which player  $i$  hasn't observed  $c_k(1)$ . In addition, for every resource  $k \in R$ ,  $b(k) = k + 1 \pmod{K}$ .

### The Algorithm - LE-RS

Each player  $i$  has a vector of variables  $\mathbf{c}^i = c_k^i(l)$  for every  $k \in R$ ,  $l \in N$ . Throughout the algorithm, even if not specified, player  $i$  updates  $c_k^i(l)$  when she observes  $c_k(l)$ .

Input: A class  $\mathcal{M}$  of repeated resource selection games in which  $R = \{1, \dots, K\}$  ( $K \geq 1$ ).<sup>6</sup>

Initialize:

Player  $i$  initializes  $c_k^i(l) = 0$  for every  $k \in R$  and every  $l = 1, \dots, n - 1$ , and  $c_k^i(n) = C_{max}$  for every  $k \in K$ .

Cooperative Exploring:

Player  $i$ 's strategy:

For  $k = 1, \dots, K$

(i) repeat  $i - 1$  times: play  $b(k)$  for  $i - 1$  iterations, and play  $k$  for another  $n - i + 1$  iterations.

(ii) play  $k$  for  $n$  iterations.

(iii) repeat  $n - i$  times: play  $b(k)$  for  $i$  iterations, and play  $k$  for another  $n - i$  iterations.

If some player, whom we refer to as the *adversary*, deviates from the above procedure, move to the "Exploring with an adversary" stage. Otherwise, move to the "Playing Optimal" stage.

Playing Optimal: At this stage the true game is commonly known. Let  $\mathbf{z} \in \arg \max_{\mathbf{z}' \in Z} \sum_{i \in N} c^i(\mathbf{z}')$ .  $\mathbf{z}$  is chosen with respect to some pre-defined lexicographical order. Let  $j$  be the current iteration.

Player  $i$ 's strategy:

Play  $z_i$  at iteration  $j$ .

At every iteration  $j' > j$  play as player  $i + 1$  played at iteration  $j' - 1$  (player  $n$  follows player 1).

If some player does not behave according to the above procedure, then move to the "Exploring with an adversary" stage.

<sup>6</sup>We omit the case  $K \geq 2$  due to lack of space.

Exploring with an adversary: From this stage on, we assume w.l.o.g that the adversary is player  $n$ .

Player  $i$ 's strategy:

For  $i' = 1, \dots, n-1$

for  $k = 1, \dots, K$

for  $l = 2, \dots, n-1$

If  $i = i'$  ( $i$ 's turn to learn): play  $T_1$  ( $T_1$  will be defined later) iterations resource  $k$ , or until  $c_k(l)$  is observed (what occurs first).

If  $i' \neq i$ : if  $i > i'$  let  $q = i - i' + 1$ . Otherwise let  $q = n - i' + i$ . Play for  $T_1$  iterations or until player  $i$  observed  $c_k(l)$  the following action -  $b(k)$  if  $q > l$ ,  $k$  if  $q < l$ , and randomize uniformly between  $k$  and  $b(k)$  if  $q = l$ .

If after  $T_1$  iterations player  $i'$  fails to observe  $c_k(l)$  then move to the "Failure" stage.

If all phases are completed successfully then move to the "Learn Alone" stage.

Learn Alone: For every  $i$  let  $s(i)$  be the set of all the resources  $k \in K$ , in which player  $i$  hasn't observed  $c_k(1)$ . In case that  $|s(i)| = 1$  we let  $s(i)$  be the resource itself (and not the set containing this resource).

Player  $i$ 's strategy:

For  $i' = 1, \dots, n$

If  $i = i'$  ( $i$ 's turn to learn): as long as  $|s(i)| \geq 2$  repeat: let  $k_1, k_2 \in s(i)$  be two different resources such that  $k_1 < k_2$  and  $k_2 < k'$  for every  $k' \in s(i) \setminus \{k_1, k_2\}$ . Play  $k_1$  and  $k_2$  each with probability  $\frac{1}{2}$  for  $T_1$  iterations or until  $c_{k_1}(1)$  or  $c_{k_2}(1)$  are learned. If after  $T_1$  iterations  $i$  fails to learn  $c_{k_1}(1)$  or  $c_{k_2}(1)$  then move to the "Failure" stage.

If  $i \neq i'$ : as long as  $|s(i')| \geq 2$  repeat: let  $k_1, k_2 \in s(i')$  be two different resources as above. Play any resource  $k \notin \{k_1, k_2\}$  for  $T_1$  iterations or until player  $i'$  observed  $c_{k_1}(1)$  or  $c_{k_2}(1)$ . If after  $T_1$  iterations  $i'$  fails to learn  $c_{k_1}(1)$  or  $c_{k_2}(1)$  then move to the "Failure" stage.

Repeat the following: Let  $D = \{i \in \{1, \dots, n-1\} : |s(i)| = 1\}$ . If for every  $i_1, i_2 \in D$ ,  $s(i_1) = s(i_2)$  then move to the "Punishing" stage. Let  $i_1, i_2 \in D$  such that  $i_1 < i_2$  and  $i_2 < i'$  for every  $i' \in D \setminus \{i_1, i_2\}$ .

If  $i \in \{i_1, i_2\}$ , play  $s(i)$  in the next iteration. If  $i \notin \{i_1, i_2\}$ , play any resource other than  $k$  such that  $k \neq s(i_1)$  and  $k \neq s(i_2)$  in the next iteration.

Punishing: Let  $N' = \{1, \dots, n-1\}$  and let  $D = \{i \in \{1, \dots, n-1\} : |s(i)| = 1\}$ . Denote by  $G_{c^i}^{N'}$  the game with the set of players  $N'$  and resource cost functions  $c^i$ .

Player  $i$ 's strategy:

If  $D = \emptyset$ : let  $\mathbf{z}$  be the equilibrium constructed through the NE-SimProcess in  $G_{c^i}^{N'}$ . Play  $z_i$  indefinitely.

If  $D \neq \emptyset$ : let  $i' \in D$ , and let  $k \in s(i')$ . Let  $\tilde{c}^i = c^i$ . Set  $\tilde{c}_k^i(1) = 0$ . let  $\mathbf{z}$  be the equilibrium constructed through the NE-SimProcess in  $G_{\tilde{c}^i}^{N'}$ .

If  $z_i \neq k$  or  $\sigma_k(\mathbf{z}) > 1$ : play  $z_i$  indefinitely.

If  $z_i = k$ : repeat the following: if  $c_k(1)$  has been observed, choose a resource  $k' \in B_i(\mathbf{z}_{-i})$  and play it indefinitely. Otherwise play  $k$ .

Failure: Player  $i$ 's strategy: play every iteration any action.

## 6. Analysis

In this subsection we prove our main result:

**Theorem 1** *Let  $\mathcal{M}$  be a class of resource selection games with a given upper bound, and let  $\epsilon > 0$ . The strategy profile in which all players adopt the LE-RS algorithm is a socially optimal learning  $\epsilon$ -equilibrium in the associated repeated game with incomplete information with partial monitoring.*

*Proof (Sketch).* The proof is given for the case, where the number of resources  $K$ , is  $K \geq 3$ . The case  $K = 2$  requires a special treatment which is given only in the full version.

Let  $\mathbf{f}$  be the strategy profile in which all players use the LE-RS algorithm. We assume that  $C_{max} > \epsilon$  otherwise the result is trivial. Let  $\hat{S}(G) = \frac{S(G)}{n}$ .

**Claim 1:** There exists  $\bar{T}$ , polynomial in  $n, K, C_{max}$  and  $\frac{1}{\epsilon}$ , such that if all players follow the LE-RS algorithm, then for every  $t \geq \bar{T}$ , and every  $i \in N$ ,  $|C^i(G, \mathbf{f}, t) - \hat{S}(G)| \leq \frac{\epsilon}{4}$ . The proof of Claim 1 is omitted due to lack of space, and will appear in the full paper.

Suppose a single player deviated from the LE-RS algorithm (we refer to him as the adversary). We show that he will be punished in an efficient time (from the beginning of the game). W.l.o.g let the adversary be player  $n$ . For the remainder of the proof by using the term player, we refer to non-adversarial players. We let  $N' = \{1, \dots, n-1\}$ . Notice that since the on-path stages ("Cooperative Exploring" or "Playing Optimal") are fully deterministic, all players will be able to simultaneously identify the adversary. Let  $g_n$  be the adversary's deviating strategy.

We distinguish between the following cases:

(i) The players identify the adversary during the "Playing optimal" stage: Hence, the players move to the "Punishing" stage after all players know all resources. Therefore they play in every iteration (in the "Punishing" stage) a Nash equilibrium of the true game with the set of players  $N'$  (constructed by the NE-Process). Hence, by Observation 1 the adversary's cost in every iteration during the "Punishing" stage will be at least  $NE^{max}(G)$ . Notice that  $NE^{max}(G) \geq \hat{S}(G)$ . Let  $\bar{T}$  be as in Claim 1. Suppose the adversary deviated before stage  $\bar{T}$ . In this case the least cost she can suffer at stage  $t \geq \bar{T}$  is  $\frac{(t-\bar{T})NE^{max}(G)}{t}$ . By Claim 1, it is enough to show that there exists  $\tilde{T} \geq \bar{T}$  such that for every  $t \geq \tilde{T}$

$$\frac{(t-\bar{T})NE^{max}(G)}{t} \geq \hat{S}(G) - \frac{\epsilon}{2}. \quad (1)$$

By solving (1) we obtain the result for  $\tilde{T} = \frac{2\bar{T}\hat{S}(G)}{\epsilon}$ .

Suppose the adversary deviated after stage  $\bar{T}$ . In this case, by Claim 1, the least cost she can suffer at every stage  $t$  is  $\frac{t(\hat{S}(G) - \frac{\epsilon}{4})}{t+1}$ . Hence, it is enough to show, again by Claim 1, that there exists  $T'$  such that for every  $t \geq T'$ ,  $\frac{t(\hat{S}(G) - \frac{\epsilon}{4})}{t+1} \geq \hat{S}(G) - \frac{\epsilon}{2}$ . Setting  $T' = \frac{2(\hat{S}(G) - \frac{\epsilon}{4})}{\epsilon}$  obtains the result. Setting  $T = \max\{\bar{T}, T'\}$  completes the proof for this case. Note that both  $\bar{T}$  and  $T'$  are polynomial in  $C_{max}, \frac{1}{\epsilon}, n, K$ .

Before the next case we give the following claim.

**Claim 2:** During the punishing stage, in every  $G \in \mathcal{M}$ , the adversary's cost can be lower than  $NE^{max}(G)$  for at most one iteration. The proof of Claim 2 is given in the full version.

(ii) The players identify the adversary during the "Cooperative Exploring" stage: Thus, they all move to the "Exploring with an adversary stage". In this stage, the probability that some player  $i$  fails to learn  $c_k(l)$ , for some  $2 \leq l \leq n-1$  is  $(\frac{1}{2})^{T_1}$ . In the "Learn Alone" stage the probability that a player  $i$  fails to learn  $c_{k_1}(1)$  or  $c_{k_2}(1)$  when randomizing between  $k_1$  and  $k_2$  is also  $(\frac{1}{2})^{T_1}$ . Since there are at most  $(n-2)(n-1)K$  entries to learn in the "Explore with an adversary stage" and at most  $(K-1)(n-1)$  entries to learn in the "Learn Alone" stage, the probability that the players do not reach the "Punishing" stage is bounded by  $[(n-1)(n-2)K + (K-1)(n-1)](\frac{1}{2})^{T_1} \leq 2n^2K(\frac{1}{2})^{T_1}$ . Therefore, the probability that the players reach the "Punishing" stage is at least  $p = 1 - 2n^2K(\frac{1}{2})^{T_1}$ .

Let  $T_2$  be the maximal number of iterations before the players reach the "Punishing" stage, given that they do not reach the "Failure" stage. Hence,  $T_2 = n^2K + [(n-1)(n-2)K + (K-1)(n-1)]T_1$ . Notice that for every  $T_1 \geq 1$ ,  $T_2 < 2n^3KT_1$ . Let  $T'_2 = AT_1$  where  $A = 2n^3K$ . Therefore, by Claim 2, for any  $t \geq T'_2$  and every  $G \in \mathcal{M}$

$C^n(G, (\mathbf{f}_{-n}, g_n), t) \geq p \frac{(t-T'_2)NE^{max}(G)}{t}$  (the probability that the players reach the "Punishing" stage is at least  $p$ , and in at least  $t - T'_2$  the adversary's cost will be at least  $NE^{max}(G)$ ).

Since  $\hat{S}(G) \leq NE^{max}(G)$ , then by Claim 1, it is enough to show that there exists  $T' \geq T'_2$ , such that for every  $t \geq T'$ , and for every  $G \in \mathcal{M}$

$$p \frac{(t - T'_2)NE^{max}(G)}{t} \geq NE^{max}(G) - \frac{\epsilon}{2}. \quad (2)$$

Inequality (2) is equivalent to:

$$p \geq \frac{t(NE^{max}(G) - \frac{\epsilon}{2})}{(t - T'_2)NE^{max}(G)}. \quad (3)$$

Let  $f(x) = \frac{t(x - \frac{\epsilon}{2})}{(t - T'_2)x}$ . Note that  $f(x)$  is increasing in  $x$ . Therefore, it is enough to show that there exists  $T' > T'_2$ , such that for every  $t \geq T'$ ,  $p \geq \frac{t(C_{max} - \frac{\epsilon}{2})}{(t - T'_2)C_{max}}$ . Observe that  $\frac{t(C_{max} - \frac{\epsilon}{2})}{(t - T'_2)C_{max}}$  is a decreasing function of  $t$ . Therefore, it is enough to show that there exists  $T' > T'_2$ , such that for every  $t \geq T'$ ,  $p \geq \frac{T'(C_{max} - \frac{\epsilon}{2})}{(T' - T'_2)C_{max}}$ .

The last inequality is equivalent to  $pT'_2C_{max} \leq T'[pC_{max} + \frac{\epsilon}{2} - C_{max}]$ . Observe that there exist  $T_1$  (polynomial in  $\frac{1}{\epsilon}, K, n, C_{max}$ ) such that  $pC_{max} + \frac{\epsilon}{2} - C_{max} > 0$ . Hence, setting  $T' = \frac{pT'_2C_{max}}{pC_{max} + \frac{\epsilon}{2} - C_{max}}$  for such a  $T_1$ , completes the proof.  $\square$

## References

Ashlagi, I.; Monderer, D.; and Tennenholtz, M. 2006. Robust learning equilibrium. In *Proceedings of the 22th Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 34–41. Corvallis, Oregon: AUAI Press.

Ashlagi, I.; Monderer, D.; and Tennenholtz, M. 2006. Resource Selection Games with Unknown Number of Players. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, 819–825.

Bowling, M., and Veloso, M. 2001. Rational and convergent learning in stochastic games. In *Proc. 17th IJCAI*, 1021–1026.

Brafman, R. I., and Tennenholtz, M. 2002. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3:213–231.

Brafman, R., and Tennenholtz, M. 2004. Efficient Learning Equilibrium. *Artificial Intelligence* 159(1-2):27–47.

Brafman, R., and Tennenholtz, M. 2005. Optimal Efficient Learning Equilibrium: Imperfect Monitoring. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI) 2005*.

Conitzer, V., and Sandholm, T. 2003. Awesome: a general multiagent learning algorithm that converges in self-play and learns best-response against stationary opponents. In *Proceedings of the 20th ICML*, 83–90.

Erev, I., and Roth, A. 1998. Predicting how people play games: Reinforcement learning in games with unique strategy equilibrium. *American Economic Review* 88:848–881.

Fudenberg, D., and Levine, D. 1998. *The theory of learning in games*. MIT Press.

Gairing, M.; Monien, B.; and Tiemann, K. 2005. Selfish routing with incomplete information. In *Proc. of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures table of contents, Las Vegas, Nevada, USA*, 203–212.

Garg, D., and Narahari, Y. 2005. Price of Anarchy of Network Routing Games with Incomplete Information. In *Proc. of 1st Workshop on Internet and Network Economic, Springer Verlag LNCS series 3828*, 1066–1075.

Greenwald, A.; Hall, K.; and Serrano, R. 2002. Correlated q-learning. In *NIPS workshop on multi-agent learning*.

Hu, J., and Wellman, M. 1998. Multi-agent reinforcement learning: Theoretical framework and an algorithms. In *Proc. 15th ICML*.

Koutsoupias, E., and Papadimitriou, C. 1999. Worst-Case Equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, 404–413.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th ICML*, 157–163.

Monderer, D., and Shapley, L. 1996. Potential Games. *Games and Economic Behavior* 14:124–143.

Monderer, D., and Tennenholtz, M. 2007. Learning equilibrium as a generalization of learning to optimize. *Artificial Intelligence*. to appear.

Rosenthal, R. 1973. A Class of Games Possessing Pure-Strategy Nash Equilibria. *International Journal of Game Theory* 2:65–67.