DISTRIBUTED GAMES[+]

DOV MONDERER AND MOSHE TENNENHOLTZ

Faculty of Industrial Engineering and Management,
Technion–Israel Institute of
Technology, Haifa 32000, Israel
e–mail:
dov@ie.technion.ac.il
moshet@ie.technion.ac.il

Abstract. The Internet exhibits forms of interactions which are not captured by existing models in economics, artificial intelligence and game theory. New models are needed to deal with these multi-agent interactions. In this paper we present a new model – distributed games. In such a model each player controls a number of agents which participate in asynchronous parallel multi-agent interactions (games). The agents jointly and strategically (partially) control the level of information monitoring and the level of recall by broadcasting messages. As an application, we show that the cooperative outcome of the Prisoner's Dilemma game can be obtained in equilibrium in such a setting.

1. Introduction. The Internet introduces new challenges in artificial intelligence, economics and game theory.[1] It exhibits both parallel and sequential interactions. While sequential interactions have been extensively discussed in the literature, the study of parallel interactions has been neglected so far. New models of economies and games are needed in order to effectively deal with parallel interactions. In this paper we present one such new model – distributed games. Our model captures several features of distributed systems, such as the Internet. Such systems are usually assumed to be asynchronous systems where agents communicate by broadcasting messages (Tanenbaum 1988). To motivate our particular definition, think of several

[+] First Version: April 1997. This work has been influenced by discussions made by both authors with the participants of the artificial intelligence (AAAI) Spring Symposium, Stanford University 1997, and with the participants of the international conference of game theory, University of New York at Stony Brook, 1997. We thank Avraham Neyman, Haim Reisman, Ariel Rubinstein, and Doron Sonsino for helpful discussions and suggestions.

[1] See e.g., Varian [1995], Boutilier, Shoham and Wellman (1997).

users (players) who send software agents[2] to participate in auctions which are held in various locations of the net, and are conducted in a randomly chosen order. This random order of play models a situation where all auctions are supposed to take place in parallel but, due to the fact that real distributed systems are asynchronous, are actually held at different time intervals[3].

The agents can communicate by broadcasting messages. A message sent by an agent of a particular user is not considered to be private information of the agents of this user.[4] In a general distributed game the agents may be engaged in any type of interaction at each of the locations. More precisely, a distributed game is defined by the following elements:

(1) A set of Players.
(2) A set of locations.
(3) A set of agents for each player, one agent for each location.
(4) A set of games in strategic form with the given set of players, one for each location.
(5) A set of messages for each player.
(6) A probability distribution over the set of permutations of locations.

At the initial stage, Nature chooses a permutation according to the given probability distribution. In the following stages the games in the locations are played (by the agents) according to the order prescribed by the chosen permutation. After a game at a location is played, the agents at this location can send messages. The agents do not appear in the formal definition of the distributed game (which can be described formally as an extensive game with simultaneous moves (see e.g., Osborne and Rubinstein (1994)) with the given set of players), however its information structure is motivated by their existence: When an agent is called to choose an action and right after that to send a message, it is aware of its location. Any additional information it can use must be extracted from the sequence of messages it receives. Thus, two histories which generate the same sequence of messages are in the same information set. We assume that each message has a location and sender stamps,

[2] Roughly speaking, software agents are programs which are designed to serve the goals of a particular user. These agents may navigate in a computerized network, while transmitting messages among themselves, and interacting with other agents (which might be controlled by other users). The design of software agents is one of the most important research and application directions of the computer industry (CACM (1994)).

[3] It may also reflect a random decision of the auctions' organizers.

[4] This assumption reflects the current state of Internet security. One can relax this assumption by introducing probability of detection, which may be player-specific.

and that an agent can record the order of arrival of the messages. For example, consider a particular location. An agent at this location cannot distinguish between a history in which this location is the first location to a history in which this location is the last one and no agent in all other locations sent a message.

Hence, a distributed game is an extensive game with simultaneous moves, with incomplete information (e.g., about the order of stages) and with a specific player-symmetric information structure as described above. Such games have imperfect recall[5] because the particular information structure (motivated by the use of agents) implies that a player does not necessarily remember what were the previous locations at which he has already played, and even if he knows that he has already played at a particular location, he does not necessarily remember what were the actions that were selected at this location.

A particular type of a distributed game is a parallel game which is defined by the additional condition that each permutation is equally likely[6]. This definition reflects the asynchronous nature of distributed systems. In a parallel game the interactions are supposed to be completely parallel, but due to the asynchronous nature of distributed systems there is a stochastic noise caused mainly by asynchronous clocks, implying sequential interactions. The players are ignorant about the precise nature of the noise and therefore all they can do is use the principle of indifference (Keynes (1963)). That is, they assign equal probabilities to all permutations of locations[7]. A distributed game is location-symmetric if the same stage game in strategic form is played at all locations.

As an application, we analyze a location-symmetric parallel game in which the same Prisoner's Dilemma game is played at all locations. We show that if the number of locations is sufficiently large, the outcome of cooperation can be obtained in equilibrium. In this equilibrium, an agent cooperates if and only if it has not gotten a message. If it detects a deviation from cooperation it immediately broadcasts an alarm message, otherwise it does not send any message. Note that this equilibrium survives due to the joint decision of agents not to reveal to their partners and opponents any information regarding the number of future interactions, as long as their

[5] See the discussion at the end of Section 2.

[6] That is, its probability is $\frac{1}{n!}$, where n denotes the number of locations.

[7] Parallel games model multi-agent interactions in computerized distributed systems with asynchronous clocks (which cause the stochastic noise on order of stages); Each location is identified with a processor, and each agent which is active at this location, is identified with a process that runs in the processor. The assumption of asynchronous clocks is typical in distributed computing (see e.g., Fagin, Halpern, Moses, and Vardi (1995) for a discussion of this issue in the context of the theory of knowledge).

opponents cooperate with them. Hence, in equilibrium the agents never know how many more games are to be played. This lack of knowledge enables cooperation. This result is extended to a general folk theorem for location-symmetric parallel games.

2. Distributed Games.

A *distributed game* DG is defined by the following elements:

$DG1$ A finite set of players $F = \{1, 2, \ldots, m\}$, where $m \geq 2$.

$DG2$ A finite set of locations $L = \{1, \ldots, n\}$, $n \geq 1$.

$DG3$ For each Location i a game G_i in strategic form, with the set of players F. The action set of Player j at the game G_i is denoted by S_i^j. We assume that S_i^j contains at least two actions. The payoff function of Player j at G_i is $u_i^j : S_i \to R$, where $S_i = \times_{j=1}^{m} S_i^j$.

$DG4$ To each Player j, a set of agents $\{i_j\}_{i \in L}$; i_j is referred to as the agent of j at Location i.

$DG5$ For every player j a finite set M^j of messages. We assume that M^j contains the no-message option, denoted by φ, and contains at least one real message. Thus for every Player j, $M^j = \{\varphi, a_1^j \ldots, a_{k^j}^j\}$, where $k^j \geq 1$. The agents of Player j can broadcast messages from M^j.

$DG6$ A probability distribution λ on the set PE of $n!$ permutations of $\{1, 2, \ldots, n\}$, where a permutation is a one to one function $r : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$.

We need the following definition. Consider a distributed game described by $DG1 - DG6$. We define the message game MG to be the game in strategic form whose players are F. The strategy set of Player j is M^j and the payoff functions of the players are defined on $M = \times_{j \in F} M^j$, and they are constantly zero. We now define the rules of move, the payoffs, and the information structure of the distributed game DG defined by $DG1 - DG6$. Recall that the agents do not appear in the formal definition, but their existence and abilities motivate the definition.

The rules of move: The game has $n + 1$ stages. At the initial stage (Stage 0), Nature chooses a permutation $r \in PE$ according to the probability distribution λ. Each additional stage $1 \leq t \leq n$ is divided into two sub-stages, the action stage $t[a]$ and the message stage $t[m]$. At stage $t[a]$ the players play the game $G_{r(t)}$, that is, they choose simultaneously an action profile $x_t \in S_{r(t)}$. At Stage $t[m]$, which comes after stage $t[a]$, the players play the game MG. That is, they choose simultaneously a message profile $m_t \in M$.

The payoff structure: If Nature chooses r, and the players choose the action profiles $x_t \in S_{r(t)}$, $1 \leq t \leq n$, then Player j receives the accumulated payoffs in the stage games, that is, he receives $\sum_{t=1}^{n} u^j_{r(t)}(x_t)$.

The information structure: For each $r \in PE$ and for each $2 \leq T \leq n$, let $H^a_r[T] = \times_{t=1}^{T-1}(S_{r(t)} \times M)$ be the set of histories that can be generated in the game after the permutation r is chosen at stage 0 and just before the action selection stage $T[a]$. When $T = 1$ the agents at Location $r(1)$ face the empty history denoted by e_r. Hence, $H^a_r[1] = \{e_r\}$. Let $H^a_r = \cup_{T=1}^{n} H^a_r[T]$. Similarly, let $H^m_r[T] = H^a_r[T] \times S_{r(T)}$ be the set of histories that can be generated in DG after the permutation r is chosen and just before the message selection stage $T[m]$. Let $H^m_r = \cup_{T=1}^{n} H^m_r[T]$, and let $H_r = H^a_r \cup H^m_r$.

Let $H = \{(r, h_r) : r \in PE, h_r \in H_r\}$. Every $h \in H$ is a non-null history in DG. An *information structure* of a player is a partition of H. Elements in this partition are called *information sets.* As all agents at a particular location receive the same information, the structure of information in a distributed game is player-symmetric. That is, we have to define only one partition Π which is common to all players. We define this partition by using the concept of signals. With each $h \in H$ we associate a signal $I(h)$. Two histories are in the same information set if and only if they are associated with the same signal. We refer to I as the *signaling function.* We first define the set Z_i of possible signals at location i. Consider an agent at location i just before an action stage. This agent knows its location. If it does not receive any message we denote the associated signal by (i, NM) ("no message at location i"). Otherwise the agent receives a signal of the form

$$z = (i, ((i_1, m_1), \ldots, (i_T, m_T))) \qquad (2.1)$$

of non-null message profiles with a location and sender stamps. Thus, an agent that receives the signal z given in (2.1), knows that it is located at i, and that in the chosen permutation r, i_t comes before i_{t+1}, $1 \leq t < T$ and i_T comes before i (that is, $r^{-1}(i_t) < r^{-1}(i_{t+1}) < r^{-1}(i)$). This agent further knows that for $1 \leq t \leq T$, the agent of Player j at Location i_t broadcasted the message m^j_t. Let Z^a_i be the set of all signals that are feasible for an agent at location i just before an action stage. That is, Z^a_i contains the signal (i, NM) and every signal that is described by (2.1). Similarly, an agent at Location i, at a message selection stage receives a signal in Z^a_i attached with an action profile from S_i. Let $Z^m_i = Z^a_i \times S_i$ be the set of all signals that are feasible for an agent at Location i just before a message selection stage. Let $Z_i = Z^a_i \cup Z^m_i$ and let $Z = \cup_{i=1}^{n} Z_i$. Z is the set of signals. We now define $I : H \to Z$

as follows: Let $h = (r, h_r) \in H$. Therefore, for some $1 \leq T \leq n$, $h_r \in H^a_r[T]$ or $h_r \in H^m_r[T]$. Denote $i = r(T)$. Assume $h_r \in H^a_r[T]$. If $h_r = e_r$, $I(h) = (i, NM)$. Otherwise, $h_r = (x_1, m_1), \ldots, (x_{T-1}, m_{T-1})$. Let $i_1 = r(t_1), \ldots, i_k = r(t_k)$, $1 \leq k \leq T-1$, be the $r-$ordered sequence of locations from which a non-null message profile is broadcasted. That is, let t_1 be the first t for which m_t is non-null, then $i_1 = r(t_1)$, and for $1 < q \leq k$, t_q is the first $t > t_{q-1}$ for which m_t is non-null, and $i_q = r(t_q)$. $I(h)$ is defined to be $(i, ((i_1, m_{t_1}), \ldots, (i_k, m_{t_k}))) \in Z^a_i$. If $h_r \in H^m_r[T]$, $h_r = (\bar{h}_r, x_T)$ with $\bar{h}_r \in H^a_r[T]$ and $x_T \in S_i$. In this case we define $I(h) = I(r, h_r) = (I(r, \bar{h}_r), x_T) \in Z^m_i$.

In the sequel we make use of the following additional notations and definitions: Every signal $z \in Z^a_i$ is a pair (i, y), where $i \in L$ and y is a sequence of messages with location and sender stamps. We denote the set of all such sequences by Y^a. That is, $Z^a_i = L \times Y^a$. An element of Y^a is called *a location-free signal at an action stage.*

An intuitive demonstration of the information structure. It is useful to demonstrate our definition of the information structure in distributed games by a simple example. Suppose there are four locations ($n = 4$), $L = \{1, 2, 3, 4\}$, and two players ($m = 2$). Denote by i_j the agent of Player j at Location i. Let $M^1 = M^2 = \{\varphi, A, B, \ldots Z\}$ be the message sets. Assume $i = 4$ and that the agents at Location 4 are about to choose their actions in S_4 (Agent 4_j, $j \in \{1, 2\}$ has to choose an action in S^j_4). Agent 4_1 (as well as Agent 4_2) receives the signal

$$z = (i, ((i_1, m_1), (i_2, m_2))) = (4, ((3, (A, B)), (1, (B, \varphi)))).$$

That is, $i_1 = 3$, $i_2 = 1$, $m_1 = (A, B)$, and $m_2 = (B, \varphi)$. This agent (as well as the other agent at Location 4) deduces that in the chosen permutation r, Location 3 comes before Location 1, and Location 1 comes before Location 4. That is, it deduces that the chosen permutation is one of the 4 permutations (out of 24 possible permutations): $r^1 = 2314$, $r^2 = 3214$, $r^3 = 3124$, or $r^4 = 3142$. In addition, it deduces that the stage games at Locations 3 and 1 have already been played. Note that if $r = r^1, r^2$ or r^3, then it means that the game at location 2 is over, but neither of the two agents 2_1 or 2_2 sent a message. If the true r is r^4, the game at Location 2 will be played after the current stage is over. In addition, Agent 4_1 knows the messages from Locations 3 and 1, that is, it knows that Agent 3_1 sent the message A, agent 3_2 sent the message B, Agent 1_1 sent the message B, and Agent 1_2 did not send a message. Note that the agents at location 4 may deduce nothing about the actions chosen at previously played locations. To conclude, these agents extract the following information about the true history (r, h_r):

(1) $r = 2314$ and $h_r = ((x_1, (\varphi, \varphi)), (x_2, (A, B)), (x_3, (B, \varphi)))$, for some x_1, x_2, x_3 in S_2, S_3, S_1 respectively, or

(2) $r = 3214$ and $h_r = ((x_1, (A, B)), (x_2, (\varphi, \varphi)), (x_3, (B, \varphi)))$, for some x_1, x_2, x_3 in S_3, S_2, S_1 respectively, or

(3) $r = 3124$ and $h_r = ((x_1, (A, B)), (x_2, (B, \varphi)), (x_3, (\varphi, \varphi)))$, for some x_1, x_2, x_3 in S_3, S_1, S_2 respectively, or

(4) $r = 3142$ and $h_r = ((x_1, (A, B)), (x_2, (B, \varphi)))$, for some x_1, x_2 in S_3, S_1 respectively.

We now define strategies in distributed games.

Strategies: A *strategy* for Player j in DG is a pair $\sigma^j = (f^j, g^j)$, where $f^j = (f_i^j)_{i=1}^n$ and $g^j = (g_i^j)_{i=1}^n$, where $f_i^j : Z_i^a \to S_i^j$ and $g_i^j : Z_i^m \to M^j$. We refer to (f_i^j, g_i^j) as the strategy of Agent i_j. That is, for every information signal $z \in Z_i^a$ Agent i_j chooses the action $f_i^j(z)$, and for every information signal $z \in Z_i^a$ and $x_i \in S_i$ Agent i_j broadcasts the message $g_i^j(z, x_i)$.

Equilibrium: Let H_n be the the set of all histories of the form (r, h_r), where h_r is a sequence of pairs of action and message profiles of length n. That is, H_n is the set of all histories $\psi \in H$ of the form

$$\psi = (r, ((x_1, m_1), \ldots, (x_n, m_n))),$$

where $x_t \in S_{r(t)}$ and $m_t \in M$ for all $1 \leq t \leq n$. Every tuple of strategies $\sigma = (\sigma^j)_{j=1}^m$ defines a probability distribution μ_σ over H_n. The expectation operator with respect to μ_σ is denoted by E_σ. Let $\tilde{r}$ be the random variable on H_n defined by $\tilde{r}(\psi) = r$, and let X_i, $i \in L$ be the random variables defined by $X_i(\psi) = x_{r^{-1}(i)}$.

Let $E_i^j(\sigma) = E_\sigma(u_i^j(X_i))$, be the expected payoff of Player j from Location i when all players use the strategy profile σ. The expected payoff of Player j when all players use the strategy profile σ is denoted by $u^j(\sigma)$. That is,

$$u^j(\sigma) = \sum_{i=1}^{n} E_i^j(\sigma).$$

Consider a distributed game DG given by $DG1 - DG6$. We can associate with it a game in strategic form in which the player set is F, the action set of Player j is the set Σ^j of the strategies of Player j in DG, and the payoff function of j is $u^j : \Sigma \to R$, where $\Sigma = \times_{j \in F} \Sigma^j$. We say that a strategy profile σ is an equilibrium in DG if and only if it is an equilibrium in the associated game in strategic form.

Special types: A distributed game is called a *parallel game* if each permutation is equally likely, that is, $\lambda(r) = \frac{1}{n!}$ for every permutation r, where n denotes the number of locations. This definition reflects the asynchronous nature of distributed systems: In a parallel game the interactions are supposed to be completely parallel, but due to the asynchronous nature of distributed systems there is a stochastic noise, implying sequential interactions. The players are ignorant about the precise nature of the noise and therefore, because all locations are symmetric, all they can do is use the principle of indifference (Keynes (1963)). That is, they assign equal probabilities to all permutations of locations. Using distributed computing terminology one can associate each location with a processor, and each agent with a process of a particular user. Each process has access to the clock of the processor where it runs, while the clocks of the different processors are not synchronized, as in typical asynchronous distributed systems. A distributed game is called *location-symmetric*, if all stage games are identical. That is, $G_i = G_l$ for all locations $1 \leq i, l \leq n$. In a location-symmetric distributed game it is useful to define the concept of location-free strategies. Let DG be a location-symmetric distributed game. Denote by S_c ("c" for "common") the set of action profiles in any location. A strategy σ^j is *location-free* if $\sigma^j_i(i, y)$, $y \in Y^a$, does not depend on i. That is, σ^j is location-free if there exist functions $f : Y^a \to S_c$ and $g : Y^a \times S_c \to M^j$ such that for every location i, $f^j_i(i, y) = f(y)$ and $g^j_i((i, y), x_c) = g(y, x_c)$. In our software agents terminology, a player who uses a location-free strategy sends n copies of the same software program rather than sending n distinct programs. In a location-symmetric distributed game, we say that σ is a *location-free equilibrium* if it is an equilibrium and σ^j is a location-free strategy for every player j.

Game theoretic features: It may be useful to summarize some special game theoretic features of distributed games. Such games can be described as extensive games with simultaneous moves (see e.g., Osborne and Rubinstein (1994)) with incomplete information. They possess the additional feature that the information monitoring is partially controlled by the players via the messages: Classical repeated games in which the players observe at each stage the whole history of previous moves are referred to as games with perfect monitoring. Repeated games with imperfect monitoring are of two types: At the first type, the structure of the information released to the players after each stage does not depend on their actions. This is the case e.g., in repeated games with observable payoffs and unobserved actions in which after each stage the players observe the history of all previous payoffs.

Another example is games with bounded recall (Lehrer 1988), in which after each stage the player observe the last k moves. In the other type of imperfect monitoring, the structure of monitoring depends on the actions. Such is the case in distributed games in which the structure of information release regarding a previously active location depends on whether one of the agents at this location sent a message.[8]

A distributed game (with at least two locations) has imperfect recall[9]: An agent of Player j may not remember the actions chosen at previous locations, though the agent of Player j at such previous location has observed some of these actions. Like the concept of monitoring, there are two types of recall structure. In one of them (like in games with bounded recall) the level of recall is action-independent. In the other type (e.g., in a distributed game) the level of recall depends on the actions. In particular, in distributed games the use of agents enable the players to jointly commit themselves (by not sending messages) not to remember that they have already been in a particular location.

Recently, following Piccione and Rubinstein (1997), equilibrium concepts in games with imperfect recall have been extensively discussed in the literature. See e.g., the special issue on imperfect recall, Kalai (1997). In particular, there is no agreement about the "right" concept of equilibrium in such games. Moreover, even the concept of "optimal" strategy in one person game with imperfect recall is not agreed upon. The debate however concerns abstract games. Our definition of equilibrium seems to us natural in distributed games in which the players send their agents (programs) to the locations before the game begins and they cannot replace them after the game begins.

Our definition of distributed games can be modified in several natural ways in order to deal with other types of distributed interactions. We discuss such

[8] Note that every player has a strategy which guarantees that at every history which is consistent with this strategy, each of his agents knows the sequence of previous active locations. In particular, each agent knows the true stage at every stage. In such a strategy every agent of this player at every location broadcasts some message for every information signal it receives. However, since all players have symmetric information structure, using such a strategy implies that at every location, the sequence of past active locations is commonly known by the agents of *all* players. Note moreover that if the message space of Player j contains a sufficient number of distinct messages (that is, if the language of Player j's agents is sufficiently rich), then Player j has a strategy which guarantees that it deduces along every consistent history, the locations that have already been played and the actions and messages chosen at these locations.

[9] The concept of imperfect recall and related topics were already discussed in the early game theory literature, e.g., in Kuhn (1953), Dalkey (1953), (1957), and Aumann (1964). The original definition of perfect and imperfect recall (see e.g., Mertens,Sorin and Zamir (1994)) applies to extensive games in which every decision node belongs to a single player. This definition is naturally generalized to extensive games with simultaneous moves and,in particular, to distributed games.

modifications in Section 6.

3. Cooperation in the Parallel Prisoner's Dilemma Game. In this section we analyze a location-symmetric parallel game with two players and n locations, $L = \{1, \dots, n\}$, in which at each location the agents of the players play the Prisoner's Dilemma game described below:

$$\begin{array}{cc} & \begin{array}{cc} D & C \end{array} \\ \begin{array}{c} D \\ C \end{array} & \begin{pmatrix} (a,a) & (b,0) \\ (0,b) & (c,c) \end{pmatrix} \end{array},$$

where $b > c > a > 0$.

As we deal with a parallel game, the probability distribution λ assigns probability $\frac{1}{n!}$ to each order of locations. The message sets are M^j, $j = 1,2$. We denote the empty message[10] by φ. We assume that each message space contains at least one message (hence, each M^j contains at least two elements, φ and a real message).

Consider the following inequality:

$$(EC) \qquad\qquad b \leq c + \frac{n-1}{2}(c-a).$$

We show that if (EC) is satisfied, then there exists a location-free equilibrium $\sigma = (\sigma^1, \sigma^2)$ which induces the outcome (C,C) at each location. The proof of this result appears in the full paper. We now describe the related equilibrium strategies. Let a^1, a^2 be nonempty messages in M^1, M^2 respectively. For $j = 1,2$, define $\sigma^j = (\sigma_i^j)_{i=1}^n = ((f_i^j, g_i^j))_{i=1}^n$ as follows: f_i^j assigns the action C to the information signal (i, NM) of no previous messages and it assigns the action D to any other information signal in Z_i^a. In addition, independently of the information signal $z \in Z_i^a$, $g_i^j(z,(C,C)) = \varphi$, and $g_i^j(z,(C,D)) = g_i^j(z,(D,C)) = g_i^j(z,(D,D)) = a^j$. Obviously if all agents obey the strategy profile σ, then the outcome (C,C) is played in every location. In the full paper we show that Player j $(i = 1,2)$ does not have a profitable deviation from σ^j.

4. Cooperation in the Prisoner's Dilemma Game: Related Literature. It is instructive to compare our result with other cooperation results for the Prisoner's Dilemma game, obtained by assuming finite sequentiality and bounded rationality[11]. Radner (1986) showed that when we are ready to settle for ε-equilibrium,

[10] recall that the empty message is not a message, it is just a way to describe the event that an agent does not send a message.

[11] The folk theorems (See e.g., Aumann and Shapley (1994), Rubinstein (1979)) show that cooperation is possible in equilibria of the *infinite* repeated game.

then when the number of repetitions increases the corresponding sets of ε-equilibria allow longer and longer period of cooperation. Kreps, Milgrom, Roberts, and Wilson (1982) showed that if we assume that with an arbitrary small but positive exogenous probability, one of the players is playing tit-for-tat rather than maximizing, then with sufficiently long repetition, all sequential equilibria outcomes are close to the cooperative outcome (see, however, Fudenberg and Maskin (1986)). Aumann and Sorin (1989) proved that when every player ascribes a small positive exogenous probability to his opponent being an automaton with bounded recall, then every equilibrium in sufficiently long repetition of the game is close in the payoff space to the cooperative outcome. Neyman (1985) deals with finitely repeated Prisoner's Dilemma games in which the players are restricted to use automata with a fixed number of states. When the number of stages is large relative to the number of states[12], the automata cannot effectively count the number of previous stages, and as in our case, this ignorance regarding the number of stages enables cooperation (in all stages) in equilibrium. Hence in the automaton model the players are not able to process the available information, while in our model, the players jointly and strategically decide not to obtain this information. We think that our model is more realistic in the distributed systems (e.g., Internet) setup in which it is assumed that players broadcast messages and act in a parallel asynchronous setting, while controlling software agents with a tremendous counting ability. Zemel (1989) use the finite automata model of Neyman with the additional feature of allowing the players to send and receive messages. This additional feature allows cooperation by saturating the computational resources of the players and thus preventing them from utilizing complex strategies.

The recent paper of Neyman (1996) is related to our work. It is proved there that the players in a repeated Prisoner's Dilemma can (almost) reach the cooperation outcome if they do not share common knowledge regarding the true number of stages. Roughly speaking, our model exhibits a "real life" situation where such lack of common knowledge is possible. Note however that this lack of common knowledge is not exogenous, but it is the result of the strategic decisions of the players to let their agents and opponents be ignorant about the number of (future) stages.

5. A General Folk Theorem. The results of Section 3 can be generalized to a

[12] Or relative to the number of states to the power of $\frac{1}{\varepsilon}$, when we are ready to settle for ε-equilibrium.

folk theorem for location-symmetric parallel games.

Theorem A. *Let G be a game in strategic form with the player set F and with the action sets S_c^j, $j \in F$. For every $n \geq 1$, let DG_n be the location-symmetric parallel game defined by $DG1 - DG6$ with $G_i = G$ for every $1 \leq i \leq n$. Let $b \in S_c = \times_{j \in F} S_c^j$, be an equilibrium in G. Let $x \in S_c$ satisfy:*

$u^j(x) > u^j(b)$ for every $j \in F$.

Then, there exists a positive integer N such that for every $n \geq N$ there exists a location-free equilibrium σ in DG_n, at which the outcome x is reached at each location.

The proof of Theorem A appears in the full paper.

6. Remarks.

Cryptography The assumption that messages are not private information motivates some of cryptography theory (e.g., Diffie and Hellman (1976)). In the model of distributed games considered in this paper we do not distinguish between a message (e.g., a stream of bits) and its content. In more advanced models one may wish to deal with asymmetric information setup, where only the agents of the same player can decrypt their messages. The results proved in Section 3 and 5 hold for such models too.

Extensions It is possible to extend the definition of distributed games in many ways. For example, one can replace the assumption that messages are not private information by the assumption that the messages can be received with some given probability distribution, that may be player-specific.

A more essential change in the definition of distributed game is obtained if we introduce some cost for every player who wishes his agents to receive messages sent by other players' agents.

Modifications Parallel games capture three main features of computerized distributed systems: Asynchronous clocks and activities, software agents[13], and messages. Combined together, these features, translated to a game theoretic language, make the folk theorem work. Each of these features, on its own, and any two of these features give rise to a particular type of game. For example, one may wish to analyze distributed games with synchronized clocks. In such games the probability distribution λ is concentrated at a particular permutation. Such a game captures the features of agents and broadcasting.

[13] Or communicating sequential processes (see Hoare (1985)).

Parallel Auctions Our model of parallel games can be applied to parallel auctions, in which information about one auction may leak to other auctions. The folk theorem suggests that a collusion outcome may be obtained in equilibrium in such setting.

REFERENCES.

Aumann, R.J. (1964), "Mixed and Behavior Strategies in Infinite Extensive Games," in *Annals. of Mathematics Study 52* (Dresher, M., Shapley, L.S., and Tucker, A.W., Eds.), Princeton University Press, Princeton, NJ..

Aumann, R.J. and Sorin, S. (1989), "Cooperation and Bounded Recall," *Games and Economic Behavior* **1**, 5–39.

Aumann, R. J. and Shapley, L.S. (1994), "Long Term Competition– A game Theoretic Approach," in *Essays in game Theory, In Honor of Michael Maschler* (N . Meggido, Ed.). pp. 1–15.

Boutilier, C., Shoham, Y. and Wellman, M.P. [1997], Special Issue of *Artificial Intelligence*, on *Economic Principles of Multi-Agent Systems*, Vol. 94, 1997.

CACM (1994), "Intelligent Agents," *Special Issue of CACM (Communications of the ACM)*, **37**(7).

Dalkey, N. (1953), "Equivalence of Information Patterns and Essentially Determinate Games", in *Annals. of Mathematics Study 28* (Kuhn, H.W., and Tucker, A.W.. Eds.), Princeton University Press, Princeton, NJ.

Diffie, W. and Hellman, M.E. (1976), "New Directions in Cryptography," *IEEE Transactions on Information Theory*, **22**(5), 644–654.

Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995), *Reasoning About Knowledge*, Cambridge Ma: MIT Press.

Fudenberg, D. and Maskin, E. (1986), "The Folk Theorem in Repeated games with Discounting and Incomplete Information," *Econometrica* **54**, 533–554.

Hoare, C.A.R. (1985), *Communicating Sequential Processes.* Englewood Cliffs, N.J.: Prentice Hal

Isbell, J. R. (1957), "Finitary Games," in *Annals. of Mathematics Study 39* (Dresher, M., Tucker, A.W., and Wolfe,P., Eds.), Princeton University Press, Princeton, NJ.

Kalai, E. (1997) (Ed.), "Special Issue: Imperfect Recall," *Games and Economic*

Behavior **20**, 1–130.

Keynes, J.M. (1963), *A treatise on Probability.* Macmillan, London.

Kuhn,H.W. (1953), "Extensive games and the Problem of Information," in *Annals. of Mathematics Study 28* (Kuhn, H.W., and Tucker, A.W., Eds.), Princeton University Press, Princeton, NJ.

Kreps, D., Milgrom, P., Roberts, J. and Wilson, R. (1982), "Rational Cooperation in Finitely-Repeated Prisoners' Dilemma," *Journal of Economic Theory* **27**, 245–252.

Lehrer, E. (1988), "Repeated Games with Stationary Bounded Recall," *Journal of Economic Theory* **46**, 130-144.

Mertens, J.F., Sorin, S. and Zamir, S. (1994), "Repeated games," *Core Discussion Papers 9420, 9421, and 9422,* Universite Catholique De Louvain.

Neyman, A. (1985), "Bounded Complexity Justifies Cooperation in the Finitely Repeated Prisoner's Dilemma," *Economics Letters* **19**, 227–229.

Neyman, A. (1996), "Cooperation in Repeated Games when the Number of Stages is not Commonly Known," *Center of Rationality Working Paper.*

Osborne, M., and Rubinstein, A. (1994), *A Course in Game Theory,*Cambridge Ma: MIT Press.

Piccione, M. and Rubinstein, A. (1997), "On the Interpretation of Decision Problems with Imperfect Recall," *Games and Economic Behavior* **20**, 3–24.

Radner, R. (1986), "Can Bounded Rationality Resolve the Prisoners' Dilemma," in *essays in Honor of Gerard Debreu* (A. Mas-Colell and W. Hildenbrand, Eds.), North-Holland Publ., Amsterdam.

Rubinstein, A. (1979), "Equilibrium in Supergames With the Overtaking Criterion," *Journal of Economic Theory* **21**, 1–9.

Tanenbaum, A.S. (1988), *Computer Networks.* 2nd Edition, Prentice Hall.

Varian, H.R. [1995], "Economic Mechanism Design for Computerized Agents," *Working paper, Berkeley University.*

Zemel, E. [1989], "Small talk and Cooperation: A Note on Bounded Rationality," *Journal of Economic Theory* **49**, *1-9*