Stable Strategies for Sharing Information among Agents *

Rina Azoulay-Schwartz[1] **Sarit Kraus**[1,2]
[1]Department of Mathematics and Computer Science
Bar-Ilan University, Ramat-Gan, 52900 Israel
{schwart, sarit}@macs.biu.ac.il
[2]Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742

Abstract

Information sharing is important for different goals, such as sharing reputations of sellers among potential buyers, load balancing, solving technical problems, etc. In the short run, providing information as a response to queries is often unbeneficial. In the long run, mechanisms that enable beneficial stable strategies for information exchange can be found. This paper presents such mechanisms and specifies under which conditions it is beneficial to the agents to answer queries. We analyze a model of repeated encounters in which two agents ask each other queries over time. We present different strategies that enable information exchange, and compare them according to the expected utility for the agents, and the conditions required for the cooperative equilibrium to exist.

1 Introduction

In this paper, we consider the problem of information sharing among self motivated agents. Information sharing is necessary in environments where autonomous agents are required to solve problems, and additional information may improve their performance, i.e., reputation systems, load balancing, solving problems which require specialization, etc. Information sharing among agents in such environments is supposed to increase their average utility, since the cost of one agent to find an answer to a query is usually less than the utility derived by the agent who receives the response.

Research on information sharing among agents usually assumes that the agents are motivated to share information with each other and to help each other to find the best solution to their problems [Mor, 1996; Raub and Weesie, 1990; Zacharia, 1999]. This assumption does not hold in multi-agents environments, where each agent belongs to another owner, and wants to maximize its own utility. An agent when answering a query bears the costs of searching for the answer, and sending it to the questioner, and it may also bear indirect costs. For example, if the query is about the resource with the lowest load [Schaerf *et al.*, 1995], answering it may increase the load of the resource, and this can harm the responding agent that publicized this information. The responding agent does not receive any payment for its answer, since there is no mechanism to enable such a payment. Moreover, the value of an answer cannot objectively be evaluated, and payment for answers may cause the queries flow to be damaged only as a result of evaluation problems.

In fact, each agent would like to receive answers to its own queries, while ignoring queries directed to it. Thus, as we show in Section 1.1 below, it is clear that in equilibrium of a single interaction, no agent will answer any query. However, if the interactions are repeated, strategy profiles exist in which it is worthwhile for the agents to attempt to answer queries, since their long term utility will increase.

To simplify the problem, we analyze a model of repeated interactions in which two agents contact each other and ask queries repeatedly. This problem is different from the classical prisoner's dilemma [Fudenberg and Tirole, 1991] in respect to two main issues. First, the agents do not make their decisions simultaneously: in each interaction, one agent asks a query, and the decision is made by the second agent. Second, an agent, when attempting to answer a query, may fail to find an answer, and the questioner cannot know whether it did not receive an answer because the other agent ignored its query, or because the other agent failed to find an answer. In fact, the agent which has to answer may also return a negative message, saying it cannot find an answer. However, such a response is strategically equal in our model to not responding, since in that case, the questioner cannot know whether the agent really attempted to answer its query or not. We also assume that an agent cannot send a fictive answer, since such an answer will be revealed immediately. (e.g., information about a seller cannot be given if the informer does not know actual details about it. Technical help which is not useful will immediately be found to be worthless, etc.)

Other research conducted in DAI concerning repeated interactions, deals mostly with learning the best strategy to use in repeated interactions [Sen and Arora, 1997; Sandholm and Crites, 1995; Carmel and Markovitch, 1996; Freund *et al.*, 1995]. Sen and Arora developed and analyzed probabilistic *reciprocity schemes* as strategies to be used by self interested agents to decide whether or not to help other agents. The principle of reciprocity is that agents only help those agents

*This material is based upon work supported in part by the NSF under Grant No. IIS-9820657. Rina Azoulay-Schwartz is supported in part by the Israeli Ministry of Science.

who helped them in the past or can help them in the future. Their analysis and experiments show that reciprocal behavior improves the individual agent's performance in the long run over the selfish behavior. Our research also deals with repeated interactions among self interested agents, but we take the classic game theory approach of finding strategies that are in equilibrium. Chalasani at el [Chalasani *et al.*, 1998] developed a model where querying agents send queries to information agents. They designed a randomized symmetric strategy which minimizes the expected completion time of a query. However, they do not explain the motivation of an agent to use the symmetric strategy. In our research, we consider strategy profiles, which the agents are motivated to follow. We evaluate the expected utility of the strategy profiles, and the conditions required for this profile to be an equilibrium. We combine theoretical proofs with particular examples that demonstrate the behavior of the strategy profiles for particular parameters.

1.1 The one period interaction

Consider the following interaction of two agents, i and j: Agent j is ready to ask a query, and it can either send it to agent i or not. If it sends the query, then agent i can either attempt to answer the query or not. If agent i attempts to answer the query, then with a probability of p_i it will succeed in answering the query, but with a probability of $1 - p_i$ it will fail, where $0 \leq p_i \leq 1$. If agent j does not receive an answer, it does not know whether agent i attempted to answer it and failed, or whether it even tried.

Agent i, when attempting to answer a query, incurs an obligatory cost o_i, for searching for an answer. If it succeeds in finding an answer, then it incurs an additional cost of c_i, which contains the expenses of retrieving the answer (i.e., its total cost is $o_i + c_i$). If agent i does not attempt to answer the query at all, then it will have a utility of 0. The asking agent (agent j) obtains a utility of v_j only if it receives an answer. In any other case, its utility will be 0.

Consider the one period interaction in which agent B is ready to to send agent A a query. There are two pure equilibria for this interaction: in the first, agent B will send the query to agent A, but agent A will not attempt to answer it. Note that agent B still sends its query, since we assume that it incurs no cost for sending queries. In the second, agent B will not send the query at all. In both equilibria, the utility of both agents is 0. In this paper, we present strategy profiles to be used by agents participating in the repeated version of the above interaction. We prove that under certain conditions, responding to queries is in equilibrium, and improves the agents' expected utility.

In fact, the problem can be stated more generally. Agent i can ask agent j to perform any arbitrary action, rather than answer a query. The action is costly to agent j, and it may succeed or fail. However, if the required results of the action are not achieved, agent i cannot observe whether this happened because of a failure of the action taken by agent j, or since agent j did not even attempt to perform the action. The problem is different from the classical *repeated principal-agent* problem [Radner, 1985], since each agent has a role of *a principal* in a part of the interactions, and has a role of an *agent* in the other. In the rest of the paper we refer to query answering, although our results are also appropriate for the general problem.

1.2 The repeated interaction

In the repeated interaction, there are several occurrences over time of the single interaction described above. We consider an *alternating* queries model, in which agent A asks a query, then agent B, and vice versa. Time is discrete and is indexed by $t = 1, 2, \ldots$. If it is agent i's turn to ask a query, then the probability for it to have a query at a given time period is q_i, and this probability is known to both agents. Although the agents know the probability distribution of the queries appearance, they do not know the actual time when queries will appear. This means that at a given time, each agent does not know the exact time it will be ready to send its next query, or the time its opponent will send its next a query. If a query was asked, then the one period interaction occurs, and we assume that it takes one time period. We consider a discounted utility function, and denote the discount factor of the utility function δ, where $0 \leq \delta < 1$. We assume that interactions continue indefinitely. In fact, our model also suits situations where in each interaction, there is a positive probability that no more interactions will occur, as described in [Osborne and Rubinstein, 1990]. Finally, ω denotes a configuration vector: $\omega = (p_A, p_B, q_A, q_B, \delta, v_A, v_B, c_A, c_B, o_A, o_B)$ and Ω denotes the set of all possible configuration vectors.

In this paper, we suggest a trigger strategy equilibrium [Fudenberg and Tirole, 1991] to be used by the agents in the repeated interaction. Trigger strategies are appropriate for cases where the action performed by one agent is unobserved by the other one, and it yields an outcome that is observed by both agents. However, the same outcome may be the result of different actions, with different probabilities. In this type of equilibrium, an agent uses the outcome of its opponent's action in order to decide whether to behave cooperatively, or to punish its opponent, and apply the non-cooperative strategy.

2 A one-period observation model

In this section, we consider a punishment for each time an answer is not obtained from the opponent, though there are cases in which this was not deliberately caused by the opponent. In fact, using a trigger strategy profile causes the agents to attempt to answer each other's queries, thus increasing the agents expected utility with regards to the case where the equilibrium of the one period interaction is implemented. However, there are cases where agents are punished due to failure in answering queries. We begin this section by defining the trigger strategy profile.

We suggest that the agents use a trigger strategy profile which is based on three possible "phases": $Normal$, $Punish_A$ and $Punish_B$. In phase $Normal$, each agent attempts to answer the query of the other agent. In phase $Punish_i$, agent j ignores the queries of agent i, but if agent j asks a query, agent i will attempt to answer it. At the beginning, the agents are in phase $Normal$, and remain there providing each agent answers its opponent's query. Given phase $Normal$, whenever an agent i does not answer a query, they switch to $Punish_i$. This punishment phase holds until agent i answers a query of agent j, in which case, they return to phase $Normal$. The above strategy profile promotes cooperation and information sharing. In the next section we reveal under which conditions it is an equilibrium.

$\mathcal{D}_i$ is the expected discount ratio from the time agent i asks a query, until the time agent j will be ready to ask a query. We proved that

$$\mathcal{D}_i = \delta q_i + \delta^2(1-q_i)q_i + ... = \frac{\delta q_i}{1-\delta(1-q_i)}. \quad (1)$$

Denote the present time t_0, the time when agent B asks a query t_B, and the time after t_B in which agent A sends a query to B t_A. Denote the overall expected discount ratio from t_0 until t_A, $\mathcal{D}$. We proved that

$$\mathcal{D} = \frac{\delta^2 q_A q_B}{(1-\delta(1-q_A))(1-\delta(1-q_B))} = \mathcal{D}_A\mathcal{D}_B \quad (2)$$

Symmetrically, we proved that $\mathcal{D}$ is also the expected discount in the case of punishing agent B. We proved that $0 \leq \mathcal{D}_i < 1$, and also that $0 \leq \mathcal{D} < 1$. The proofs of this proposition, as well as the other proofs, can be found in [Azoulay-Schwartz, 2001]. For space limitation, we do not present them here.

2.1 Expected Utility and Equilibrium Conditions

In this section, we specify the expected utility of the agents when they follow the strategies profile described above, and the conditions under which this profile is an equilibrium. First, we define the terms that will be used for these specifications.

Definition 2.1 *The following terms express the expected utility of the agents, from the present until infinity.*

- V_i *: the expected utility of agent i if it attempts to answer the query of agent j (whether it succeeds or not).*
- U_i*: the expected utility of agent i when it is agent j's turn to answer i's query (whether j succeeds or not).*
- F_i*: the expected utility of agent i as the agents move to phase $Punish_i$ (either since agent i ignored the last query, or because of agent i's failure to answer it).*

Generally, we consider the expected utility and the trigger equilibrium condition of agent A. B's specifications can be detailed symmetrically. We consider an unrestricted horizon model, so the utility terms are defined recursively.

Attribute 2.1 *The values of V_A, U_A, and F_A are computed as follows.*

$$V_A = \quad -o_A + p_A(-c_A + \mathcal{D}_A U_A) + (1-p_A)F_A \quad (3)$$
$$U_A = \quad p_B(v_A + \mathcal{D}_B V_A) + (1-p_B)\mathcal{D}U_A \quad (4)$$
$$F_A = \quad \mathcal{D}V_A \quad (5)$$

V_A is the expected utility of agent A from attempting to answer a query. It consists of the expected future utility when the attempt to answer succeeds, and the expected utility when it fails, with the corresponding probabilities for both events, and the obligatory cost o_A. In case of success, the utility of agent A consists of the cost of c_A, and of its utility from asking agent B a query (U_A), after an expected discount of $\mathcal{D}_A$. In case of failure, agent A's utility is F_A.

U_A is the expected utility of agent A when it asks a query. If agent B succeeds to answer agent A's query, then agent A receives an immediate utility of v_A, and the agent stays in state $Normal$, i.e., after a delay of $\mathcal{D}_B$, agent A will be required to answer agent B's query, with an expected utility of V_A. If agent B fails to answer the query, then after a delay of $\mathcal{D}$, it will be required to answer the next query of agent A, i.e., agent A's expected utility is U_A.

F_A is the utility of agent A when it does not answer the query of agent B. Agent A will be punished, and after an expected discount ratio of $\mathcal{D}$, again, it will be its turn to answer agent B's query, i.e., its expected utility will be V_A.

We proceed with identifying the conditions under which the trigger equilibrium exists. In particular, we use the strategy profile defined in the beginning of Section 2, and we specify the condition under which each agent prefers the trigger strategy over deviation and ignoring the queries, given that the second agent uses its trigger strategy. If the condition of each agent holds, then the trigger strategy profile is an equilibrium.

Lemma 2.1 *The trigger equilibrium of the one period observation strategy profile is an equilibrium if $V_i \geq F_i$, for $i \in \{A, B\}$.*

The above condition claims that whenever the utility of answering a query is higher for the agent than its utility from ignoring the query, a trigger equilibrium exists. In the following lemma, we found an explicit formula which defines V_A, by using formulas 3- 5.

Lemma 2.2 *The expected utility of agent A when attempting to answer a query, can be formalized as follows.*

$$V_A = \frac{(1+\mathcal{D}p_B - \mathcal{D})(-o_A - p_A c_A) + p_A \mathcal{D}_A p_B v_A}{(1+\mathcal{D}p_B - \mathcal{D})(1-\mathcal{D}+\mathcal{D}p_A) - p_A\mathcal{D}p_B} \quad (6)$$

Using lemma 2.1, and the definitions of F_A and F_B, we can progress with finding the explicit conditions for the existence of the trigger equilibrium. In fact, the condition of agent i can be displayed as a required ratio between v_i and $c_i + \frac{o_i}{p(i)}$ for agent $i \in \{A, B\}$.

Lemma 2.3 *If the agents are risk-neutral, then the one period observation strategy is an equilibrium for agents A and B, if the following condition holds for both $i = A, j = B$ and $i = B, j = A$.*

$$\frac{v_i}{c_i + \frac{o_i}{p_i}} \geq (\frac{1-\delta+\delta q_i}{\delta q_i p_j} + \frac{\delta q_j(p_j - 1)}{p_j(1-\delta+\delta q_j)}) \quad (7)$$

Using the above lemmas, important properties can be identified, concerning the strength of the equilibrium and the influence of the configuration parameters on the conditions of the equilibrium and on the agents expected utility. We present our conclusions in the following section.

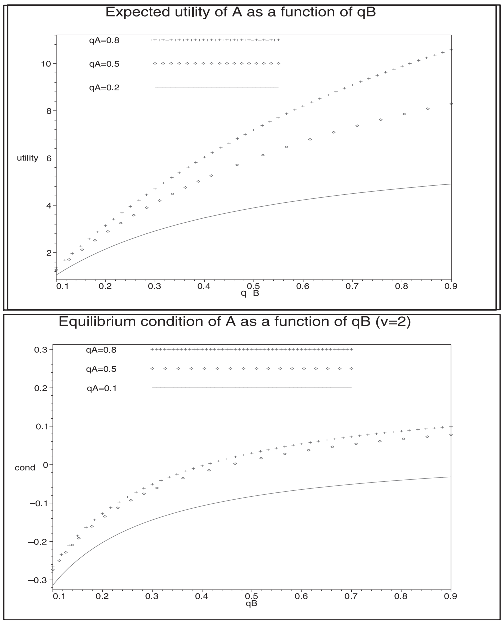

Figure 1: The influence of q_A and q_B: Up: the expected utility V_A. Down: the trigger equilibrium condition of agent A ($v_A = 2$; the equilibrium does not hold for all cases indicated by $y < 0$.)

2.2 Properties of the Expected Utility and of the Equilibrium Conditions

In this section, we study the existence of the trigger equilibrium and the agents' expected utility V_A. The value $V_A - F_A$ of lemma 2.1 should be non-negative for the equilibrium to exist. As this value increases, the trigger equilibrium exists for a larger set of configurations. Some of the conclusions are proved formally, while others are demonstrated for a particular configuration of parameters $\omega = (q_i = 0.1, p_i = 0.5, \delta = 0.9, c_i = 1, o_i = 0.1, v_i = 10)$.

Lemma 2.4 *As v_A, p_A or p_B increases, and as o_A or c_A decreases, the expected utility of each agent increases, and the trigger equilibrium holds for more configurations.*

The above conclusion is intuitive, since as the benefits an agent obtains from answering queries, v_A, increases, the utility of agent A increases, and it is more worthwhile for it to answer queries. It is also expected that the direction of influence of o_A and c_A will be opposite: as they increase, attempting to answer queries is more costly, so the utility, as well as the tendency of agent A to answer queries, decreases. Similarly, as p_A increases, V_A increases, since if agent A succeeds to answer more queries, its utility increases. As p_B increases, agent B succeeds to answer more queries of agent A, and agent A's utility increases (more cases where a utility of v_A is obtained), as well as its willingness to answer B's queries.

As δ increases, the expected utility of agent A also increases, as well as its tendency to attempt to answer agent B's query. The reason being that agent A bears present costs in order to achieve future benefits. Thus, as the discount of time decreases, the weight of the future benefits increases, and this causes the utility to increase, and the tendency to answer queries to increase too.

Figures 1-up and 1-down show that as q_A or q_B increase, the expected utility of agent A increases, as well as its tendency to follow the trigger strategy. As q_A increases, agent A is supposed to ask queries more frequently, so its utility from receiving answers increases. Thus, it is more beneficial for it to answer others' queries, since this will enable it to receive answers to its own queries. Thus, its tendency to attempt to answer queries, and its expected utility in this situation, increase with q_A. The influence of q_B is not intuitively clear. On the one hand, as q_B increases, agent B will ask a query more often, and this causes future costs for agent A. On the other hand, since the agents alternate in asking their queries, more frequent queries of agent B will cause agent A to also ask queries more often, and this may improve its utility, and its motivation to answer agent B's queries. In fact, Figure 1-up demonstrates that the influence of q_B is positive, and is similar to the influence of q_A. In order to check this phenomenon, we created 50,000,000 random configurations in which equilibria existed. In all these configurations the influence of q_B was positive. In [Azoulay-Schwartz, 2001] we considered the influence of q_B in situations where queries are not alternating, but at any given time, each agent can send a query. In these situations the influence of q_B is negative: as agent B is supposed to have more queries, the expected utility of agent A decreases.

To summarize, we have shown the influence of several parameters on the expected utility of agent A, and on its willingness to attempt to answer queries. Symmetric conclusions hold for agent B's utility and its trigger equilibrium condition. In fact, we can see that as the factors change in a direction that increases the utility of the agent, it will be more motivated to attempt to answer its opponent's queries. This conclusion does not hold for the situation of Section 3, as we change the length of the history that is taken into consideration.

3 A model with n periods observation

The trigger strategy is composed of punishment also in situations when the punished agent behaves cooperatively and follows its trigger strategy. This is unfair, and reduces the expected utility of the agents. Thus, we tested different strategy profiles where punishment is used, but more rarely. In this section, we consider a model, in which the n last periods are observed by the agent in order to decide whether to answer a query of its opponent or not. We consider two variations of the n periods observation model. In Section 3.1- 3.2, an agent is punished after n consequent queries with no answer by this agent. In Section 3.3, punishment is implemented after k unanswered queries, out of the n last queries to that agent.

In addition, in [Azoulay-Schwartz, 2001] we also considered a mixed strategy profile in which for some histories, an agent i will randomly decide whether or not to attempt to answer a query. There are two situations in which a mixed strategy can be considered. (a) in a punishment phase, where agent i is allowed to punish agent j; (b) in the Normal phase, when agent i is supposed to answer j's query. We proved that a mixed strategy profile in a punishment phase (case (a)) is not stable. A mixed strategy profile may be stable in the *Normal* phase (case (b)), but we proved that its conditions are equivalent to those of the corresponding pure strategy, while the expected utility of the agents when using a mixed strategy profile, is lower than their expected utility when using the equivalent pure strategies. Thus, mixed strategies are not recommended for use in our model. In [Azoulay-Schwartz, 2001] we specify the model details and we prove our claims, but for space limitation, we do not specify these details here.

The results of the $n-1$ last events when agent i was required to answer queries is denoted h_i. The history of agent i is composed as follows: $h_i = (h_i(n-1), ..., h_i(2), h_i(1))$, where $h_i(1)$ represents the last event of a query sent to agent i. $h_i(k) = 0$ if the k's last query to agent i received no answer, and $h_i(k) = 1$ if the k's last query was answered by agent i. The term $h = (h_A, h_B)$ contains the $n-1$ last events with respect to the queries that agent A received, and the $n-1$ last events with respect to queries that agent B received. In particular, the notation $((1, ..., 1), (0, ..., 0))$, indicates a history of $n-1$ consequent successful answers of agent A, and $n-1$ consequent queries to agent B, with no response. Concatenating a new event to h_i, $h_i << new\_event$, means deleting the oldest event in h_i, and adding a new event to h_i. Finally, the function $zero(h_i)$ returns true if all the events in h_i are unanswered queries. Using these notations, we proceed with describing and analyzing both variations of the n-periods model.

3.1 Equilibrium with punishment after n failures

In this section, we analyze a model in which punishment of an agent is performed after n consecutive events of queries with no responses. The strategies and phases of the n periods model are defined as in Section 2, but moving from phase *Normal* to phase $Punish_i$ will occur only after n consequent queries with no response from agent i. A strategy is an ***n periods trigger strategy*** if it tells each agent i to answer queries of its opponent j, unless the last n queries sent to j received no answer. In this case, agent i ignores the queries of agent j, until it receives an answer from agent j to a query. Denote by $\Omega_n \subseteq \Omega$ the set of all $\omega \in \Omega$, such that the pair of n-periods strategies is an equilibrium given combination ω.

Assuming that both agents use their n-periods strategies, $V_A^{n,h}$ is the expected utility of agent A, when it obtains a query from agent B. Similarly, $U_A^{n,h}$ is the expected utility of agent A, when it waits for an answer from agent B. Suppose that the agents are in state $Normal$.

$$suc_A(h) = (-c_A + \mathcal{D}_A U_A^{n,(h_A<<1,h_B)})$$

is the expected utility of agent A from successfully answering a query of agent B. It includes the cost c_A, and the expected utility of asking a query after a delay of $\mathcal{D}_A$. Denote by

$$fail_A(h) = \mathcal{D}_A U_A^{n,(h_A<<0,h_B)}$$

the expected utility of agent A from a failure to answer agent B's query, if this didn't cause an immediate punishment. It includes an expected utility of asking a query after a delay of $\mathcal{D}_A$, but the failure is noted in h_A, and may cause a future punishment, if there will be future consequence failures. Finally,

$$pun_A(h) = \mathcal{D} V_A^{n,(h_A<<0,h_B)}$$

is the expected utility of agent A from a punishment. After a delay of $\mathcal{D}$, agent A will be expected to answer agent B's query. The expected utility of agent A when required to answer a query, denoted $V_A^{n,h}$, is defined as follows:

$$\begin{cases} -o_A + p_A \cdot suc_A(h) + (1-p_A) pun_A(h) & zero(h_A) = true. \\ -o_A + p_A \cdot suc_A(h) + (1-p_A) fail_A(h) & otherwise \end{cases}$$

Since agent A attempts to answer the query, it bears a cost of o_A. With a probability of p_A it will succeed in answering the query, and then its expected utility is $suc_A(h)$. With a probability of $1-p_A$, it will fail, and this will be noted in its history. If the current history of agent A includes only zeroes, then $Punish_A$ is reached, and the expected utility of agent A is $pun_A(h)$. Otherwise, its expected utility is $fail_A(h)$.

Similarly,

$$suc_B(h) = v_A + \mathcal{D}_B V_A^{n,(h_A,h_B<<1)}$$

is the expected utility of agent A when agent B succeeds answering its query,

$$fail_B(h) = \mathcal{D}_B V_A^{n,(h_A,h_B<<0)}$$

is A's utility when agent B fails to answer A's query, but punishment of B is not required, and

$$pun_B(h) = \mathcal{D} U_A^{n,(h_A,h_B<<0)}$$

is A's utility when punishing agent B is required. Using the above, the expected utility of agent A, when it forwarded a query to agent B, given n and h, denoted $U_A^{n,h}$, is defined as follows:

$$\begin{cases} p_B \cdot suc_B(h) + (1-p_B) pun_B(h) & zero(h_B) = true. \\ p_B \cdot suc_B(h) + (1-p_B) fail_B(h) & otherwise \end{cases}$$

With a probability of p_B, agent B will succeed in answering, and agent A's expected utility will be $suc_B(h)$. With a probability of $1-p_B$, agent B will fail to answer agent A's query. In this case, if punishment is required, then the expected utility of agent A is $pun_B(h)$. Otherwise, its expected utility is $fail_B(h)$.

For the expected utility calculation, the agent has to use an algorithm, based on the formulas of $V_A^{n,h}$ and $U_A^{n,h}$. In fact, these formulas depend on each other. In order to implement the calculation, a predefined depth (number of future periods) should be taken into consideration. A divide and conquer algorithm, or a dynamic programming algorithm, can be used in order to calculate the values of the formulas, given the required number of future periods.

3.2 Properties of the n-periods model

In the following, we analyze some important properties of the n-periods history model. In particular, we test the influence of n on the expected utility of the agents and on the conditions required for the existence of an n-periods strategy equilibrium. We start with an auxiliary claim.

Lemma 3.1 *Consider a trigger equilibrium based on the n periods strategy profile. The equilibrium will exist, if it is worthwhile for agent A to attempt to answer agent B after a history of $((1, ..., 1), (0, ..., 0))$, and it is worthwhile for agent B to attempt to answer agent A after a history of $((0, ..., 0), (1, ..., 1))$.*

After a history of $((1, ..., 1), (0, ..., 0))$, a future punishment of agent A due to current ignorance of a query has the lowest probability after the longest delay. Thus, if it is still worthwhile for A to hold the equilibrium strategy given this history, it will be worthwhile for it to do so after any other history. Similarly, if it is worthwhile for agent B to hold the equilibrium strategy given a history of $((0, ..., 0), (1, ..., 1))$, then it will be worthwhile for it to do so after any other equilibrium. Based on lemma 3.1, in order to determine whether an n-periods equilibrium exists or not, we only need to consider the history $((1, ..., 1), (0, ..., 0))$ of agent A, and the history $((0, ..., 0), (1, ..., 1))$ of agent B. In the following theorem, we prove that the set of configurations for which the n-periods equilibrium exists reduces as n increases, i.e., the trigger equilibrium exists more rarely as n increases.

Theorem 3.1 *For each $n \in N$, $\Omega_{n+1} \subset \Omega_n$. Moreover, for each $\omega \in \Omega$, $n \in N$ exists, such that $\omega \notin \Omega_n$, but for each $0 < n' < n$, $\omega \in \Omega_{n'}$.*

The motivation in the above theorem is that as n increases, the probability of punishment because of a present disregard of a query, becomes lower, and the time when this punishment will be used becomes more distant. Thus, there are more combinations for which the threat on an agent is not strong enough. The above theorem provides a simple rule for finding the optimal strategy profile for a given configuration. In fact, if an n-periods equilibrium does not exist, the agents should reduce n, until they obtain n' for which n'-periods equilibrium does exist. They should find the largest possible n', since, as proven in the following theorem, increasing n increases the expected utility of the agents.

Theorem 3.2 *For each $n \in N$, for each ω, such that $\omega \in \Omega_n \bigcap \Omega_{n+1}$, and for each history h, $V^{n+1,h} > V^{n,h}$.*

We demonstrate our main conclusions in Figure 2, for a particular configuration of parameters ($c_i = 1, o_i = 0.1, v_i = 100, p_i = 0.5, \mathcal{D}_i = 0.9$). The figure demonstrates that as n increases, $V_A^{n,h}$ increases too, as proven in lemma 3.2, but the increase is not linear: the increment level decreases as n increases. However, as proven in lemma 3.1, as n grows, the set of appropriate configuration values becomes smaller, and this is demonstrated in the lower dotted curve, which shows the difference between the expected utility of agent A if it attempts to answer a query after history $((1, ..., 1), (0, ..., 0))$, and its utility if it ignores the query. If the difference is positive, then an n-periods equilibrium exists, as was proven in lemma 3.1. It is also clear that as the difference increases, the n-periods equilibrium will exist for a larger set of parameters. As can be seen in the figure, the trigger equilibrium does not exist for n values higher than 6. This limit will be different for different parameter values, but the conclusion is clear. There is a trade off between the expected utility and the existence of a trigger equilibrium: as n increases, the expected utility of

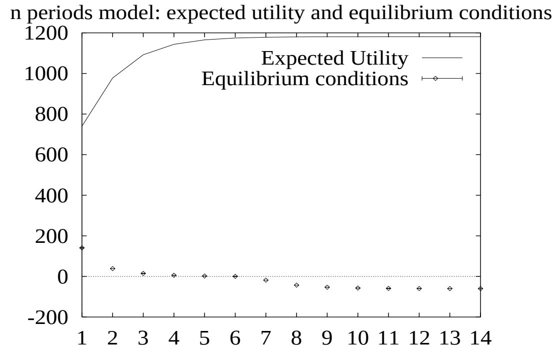

Figure 2: n-periods model: expected utility and trigger equilibrium condition as a function of n

the agents decreases, while the trigger equilibrium exists for a smaller set of configurations.

The conclusion from this section is that given a parameters configuration, ω, the agents can decide about the optimal n to be used. In fact, they will choose the largest n for which a trigger equilibrium still exists, i.e., it is still beneficial for agent A to answer queries given history $((1, ..., 1), (0, ..., 0))$, and it is still beneficial for agent B to answer queries given history $((0, ..., 0), (1, ..., 1))$. Testing these conditions can be done by using a computation method based on the formulas of $V_A^{n,h}$ and $U_A^{n,h}$, as described in Section 3.1.

3.3 Punishment after k unanswered queries out of n

In this section, we consider a model, in which n periods of history are considered, but it is enough to observe $k \leq n$ unanswered queries of an agent, in order to decide to punish this agent. The model considered in Section 3.1- 3.2 is a special case of this model, with the restriction $k = n$. We denote $\Omega_{k,n}$ to be the set of configurations for which the strategy profile of punishment after k unanswered queries out of n, is an equilibrium. The next theorem summarizes our results concerning the influence of k on the agents' expected utilities and the existence conditions of the trigger equilibrium.

Theorem 3.3 *If $k1 < k2 \leq n$, punishment after $k2$ unanswered queries over n observed queries, achieves a higher expected utility than punishment after $k1$ unanswered queries over n observed queries, but $\Omega_{k2,n} \subset \Omega_{k1,n}$.*

The above theorem is intuitively clear, since as more unanswered queries are required in order to punish, then punishment is used more rarely, and this increases the agent's expected utility, while causing deviation to be beneficial in more situations. In order to check the influence of different values of n, we developed an algorithm based on dynamic-programming to compare the expected utility of agent A and its equilibrium condition for different strategy profiles, based on different n and ks. The results are presented in Figure 3, for: $\omega = (c_i = 1, o_i = 0.1, v_i = 20, p_i = 0.5, \mathcal{D}_i = 0.9)$. As proved in lemma 3.3, we can see that as k increases, the expected utility of agent A increases, while the equilibrium condition is weaker. However, the figure also demonstrates

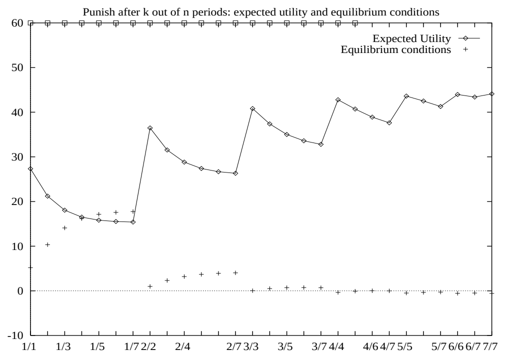

Figure 3: Punishment after k failures out of n: expected utility and trigger equilibrium conditions as a function of k/n

that as n increases, the expected utility of agent A decreases, while its condition becomes stronger. The intuition behind this result is that as the size of history observed for punishment increases from n to $n+1$, while keeping k fixed, a present ignorance of a query may cause a future punishment with a higher probability, since punishment can be utilized also due to a failure at $n+1$. Thus, the equilibrium conditions are expected to hold more frequently for larger n's. On the other hand, an increase of n will reduce the expected utility of the agent when it is required to answer a query, since failure to answer the query, will cause a punishment with a higher probability.

To summarize, we can see that different values of n and k may yield different values of expected utility and their particular value determines the existence of the trigger equilibrium. Thus, given a configuration of parameters, the agents have to find the pair of n and k for which the trigger equilibrium exists, (i.e., it is beneficial for both agents to follow the equilibrium strategies) and to choose a pair (n,k) from among them. Since there is no clear rule of choosing the optimal k and n, the agents should search all valid combinations of n and k to complete this task.

In fact, there may be situations in which each agent will prefer a different pair (n,k) due to different parameters values of the different agents. However, the agents can determine a rule of how to choose (n,k), such as, maximizing the average expected utility of them, or maximizing the product of the expected utility, etc. In the example demonstrated in Figure 3, the pair $k=n=3$ maximizes the expected utility of both agents, while the trigger equilibrium still exists. Thus, the agents should choose the equilibrium based on this pair.

4 Conclusion

In this paper, we present the problem of sharing information among self motivated agents. An agent receives queries and decides whether or not to attempt to answer them. First, we introduced the *one-period model*, in which each agent observes the last history event of its opponent in order to decide whether or not to answer it. Second, we introduced the model of punishing an agent after n unanswered queries. We found that as n increases, the expected utility of the agents increases, while there are more situations in which a trigger equilibrium does not exist. We also considered the general case, where punishment is implemented after k unanswered queries out of n queries, and we checked the influence of changing n and k.

In conclusion, we found that different punishment-based strategy profiles can be appropriate to attain responses in situations where attempting to answer queries is costly, and may result in success or failure. These profiles are stable, and increase the expected utility of the agents. Moreover, given a specific configuration, the agents may choose a strategy profile which maximizes the average, or product, of their expected utility, while a trigger equilibrium still exists. Additional variations of the model discussed in this paper are studied in [Azoulay-Schwartz, 2001]. In that paper we consider a model where at each time, each of the agents may have a query. In future work, we intend to consider situations where different parameters (n and k) are used by different agents, and also to consider a model where an agent can send its query to more than one agents.

References

R. Azoulay-Schwartz. *Protocols, Strategies and Learning Techniques for Reaching Agreements Mor e Effectively*. PhD thesis, Bar Ilan University, 2001. forthcoming.

D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Proc. of AAAI-96*, pages 62–67, 1996.

P. Chalasani, S. Jha, O. Shehory, and K. P. Sycara. Strategies for querying information agents. In *Cooperative Information Agents*, pages 94–107, 1998.

Y. Freund, M. Kearns, D. Ron Y. Mansour, R. Rubinfeld, and R. Schapire. Efficient algorithms for learning to play repeated games against computationally bounded adversaries. In *FOCS-95*, pages 332–341, 1995.

D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

Y. Mor. Computational approaches to rational choice. Master's thesis, Hebrew University, Jerusalem, Israel, 1996.

M. J. Osborne and A. Rubinstein. *Bargaining and Markets*. Academic Press Inc., San Diego, California, 1990.

R. Radner. Repeated principal agents games with discounting. *Econometrica*, 53:1173–1198, 1985.

W. Raub and J. Weesie. Reputation and efficiency in social interactions: An example of network effects. *American Journal of Sociology*, 96 (3):626–654, November 1990.

T. W. Sandholm and R. H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, pages 147–166, 1995.

A. Schaerf, Y. Shoham, and M. Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *JAIR*, 2:475–500, 1995.

S. Sen and N. Arora. Learning to take risks. In *AAAI-97 Workshop on Multiagent Learning*, pages 59–64, July 1997.

G. Zacharia. Collaborative reputation mechanisms for online communities. Master's thesis, MIT, Sept 1999.