Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Artificial Intelligence 154 (2004) 43–93

Artificial Intelligence

www.elsevier.com/locate/artint

Stable repeated strategies for information exchange between two autonomous agents ☆

Rina Azoulay-Schwartz [a], Sarit Kraus [a,b,*]

[a] *Department of Computer Science, Bar-Ilan University, Ramat-Gan, 52900 Israel*
[b] *Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*

Received 16 November 2001

Abstract

This paper deals with the problem of designing a strategy profile which will enable collaborative interaction between agents. In particular, we consider the problem of information sharing among agents. Providing information in a single interaction as a response to queries is often nonbeneficial. But there are stable strategy profiles that make sharing information beneficial in the long run. This paper presents these types of 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.

Keywords: Autonomous agents; Multiagent systems; Repeated encounters; Cooperative equilibrium

1. Introduction

In this paper, we consider the problem of enabling helpful behavior of agents, for situations where helpful behavior is not beneficial in the short run. We consider situations where an agent can ask for help from another agent. However, if no help was received, the agent cannot observe whether the other agent attempted to help or not. The asking agent (B) needs the help of the other agent (A), but this help is costly for A. Furthermore, we

☆ This material is based upon work supported in part by the NSF under Grants No. IIS-9820657 and IIS-0208608. Preliminary results were reported in IJCAI-01.
* Corresponding author.
E-mail addresses: schwart@macs.biu.ac.il (R. Azoulay-Schwartz), sarit@macs.biu.ac.il (S. Kraus).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2003.04.002

consider situations where no payment mechanism exists to compensate an agent for its help (for example, information agents that do not charge money for using them, or information agents of private persons). Thus, each agent will be motivated to get help, but has no motivation to provide any help to its opponent. This type of problem appears in 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, reviewing papers, 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 receiving 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 [10,14,22]. This assumption does not hold in multi-agent environments, where each agent belongs to another owner, who wants to maximize its own utility. When answering a query an agent 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 [16], 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. Payment for answers may reduce the efficiency of the overall system since an agent may give up on sending queries, only because its estimation about the benefits from them is too low.

Given the above domain, each agent would like to receive answers to its own queries, while ignoring queries directed to it. Thus, as we show in Section 3.2 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. In these strategies an agent that does not answer a query is punished by the inquiring agent. The punishment is implemented by ignoring queries of the punished agents.

To simplify the problem, we analyze a model of repeated interactions in which two agents contact each other and repeatedly ask queries. We check under which conditions it will be worthwhile for an agent to answer the queries of its opponent. In order to consider the general case where several agents are connected, each pair of agents can be analyzed separately. Furthermore, the benefits of an answer obtained by a certain agent should be evaluated given the fact that answers can also be obtained by other agents.

The problem of enabling cooperation in answering queries is different from the classical prisoner's dilemma [8] with respect to two main issues. First, the agents do not make their decisions simultaneously: in each interaction, one agent asks a query, and a 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. The agent which has to answer may consider to send a message indicating that it failed to 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 immediately revealed. For example, in the commerce domain, information about a seller cannot be given if the informer does not know actual details about it. Similarly, technical help which is not useful will immediately be found to be worthless, etc.

The paper is organized as follows. Section 2 provides a survey of related work on artificial society and on reputation mechanisms. Section 3 describes the basic model of information sharing. First, we consider the case in which an agent punishes the other agent each time the other agent has not responded to its query, which is presented in Section 4. In Section 5 we consider the general case, in which the n last queries presented to agent A are observed in order to decide whether to punish agent A. Finally, Section 6 provides the conclusions and suggests directions for future research. The paper's proofs and a table of symbols appear in appendices.

2. Related work

This paper deals with enabling information sharing among two agents. The motivation behind this issue is in the formation of societies of agents which share information. We claim that agents share information with each other due to the fact that interactions are repeated, and equilibrium can be based on punishment which is implemented by ignoring queries of the punished agents. In this section, we survey related work on two issues. We present previous work on interactions of agents in repeated games in Section 2.1, and in Section 2.2 we discuss the previous work on gathering and sharing information among self interested agents.

2.1. Cooperation among self interested entities

Research conducted in DAI concerning cooperation and coordination in repeated games deals mainly with learning the best strategy to play in this kind of game. The typical approach is to assume that the adversary's strategy is a member of some natural class of computationally bounded strategies. In this section we describe related work on strategies and learning techniques used by agents in repeated games.

Axelrod's model [2] of the evolution of cooperation was based on the iterated Prisoner's Dilemma. He considered a group of players playing the prisoner's dilemma repeatedly, thereby permitting partial time histories of behavior to guide future decisions. He found that a very simple strategy called "tit for tat" was the winning strategy. This strategy simply cooperates on the first move and then does whatever the opponent has done in the previous move. In our problem, we use a variation of tit-for-tat, adapted to the queries answering domain when actions are not formed simultaneously.

Sandholm and Lesser [15] suggest that agents use Q-learning [9] in repeated games, in order to learn how to play optimally against an unknown opponent. In their simulations, agents using a Q-learning algorithm succeed in learning to play optimally against tit-for-tat agents, but they face difficulties when playing against other learners. The agents

which fared best among the Q-learning agents, in the iterative prisoners dilemma, were agents with learning based on lookup table memories, with long history windows and long exploration schedules. In our work, we study a game different from the prisoners dilemma, and we study agents using deterministic strategies rather than learning methods. Nonetheless, in our research we also found that agents observing longer histories obtain higher expected utilities.

Carmel and Markovitch [5] described a *model-based* approach for learning in multi-agent systems, which split the learning process into two separate stages. In the first stage, the learning agent infers a model of the other agents based on past interaction. In the second stage, the agent utilizes the learned model for designing effective interaction strategy for the future. In their simulation, the model-based agents outperform the Q-agents significantly in learning to play against random opponents.

Freund et al. [7] present efficient algorithms for learning to play two types of repeated games: penny-matching and contract games. They consider two new types of adversaries: *recent history adversaries*, whose current action is determined by some simple Boolean formula of the recent history of the game, and *statistical adversaries*, whose current action is determined by some simple function of the statistics of the entire history of the game. For both classes of adversaries, they developed efficient algorithms for learning to play contract games. Finally, they consider the classical finite automata adversaries, and present an efficient algorithm for learning to play any game against any finite automata.

Sen and Sekaran [18] consider the problem of coordinating automated agents, both in cooperative and non-cooperative domains. They investigate a robot navigation problem and a resource sharing problem, and apply the Q-learning algorithm to both domains. They reveal that agents can learn to achieve their goals in both cooperative and adversarial domains. They also reveal that classifier systems achieve near-optimal solutions quicker than Q-learning, but for more rigid convergence criteria, they achieve a better solution than Q-learning, only when using a larger number of trials.

Parkes and Ungar [12] review possible models of learning in multi-agent systems. They show the influence of learning on the *compensation mechanism*, which is a mechanism for an efficient coordination of actions within a multi-agent system.

Sen and Arora [17] propose a scheme for learning to identify and exploit the weakness of a particular opponent by repeatedly playing against it over several games. They propose an expected utility maximization strategy which allows players to benefit by taking calculated risks that are avoided by the traditional min-max strategy. Their proposed mechanism improves the ability of the computer player to play more effectively against a weaker opponent.

The research described above considers learning about your opponent in a simple game which repeats itself. Our research also deals with repeated games, but we take the classic game theory approach of finding whenever a pair of strategies is an equilibrium. Thus, we do not deal with learning the strategy of the opponent, but rather with identifying stable strategies and finding the best strategy to be taken by each agent given its opponent's strategy and vice versa.

Chalasani et al. [6] 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 also consider information agents, but the strategy profiles considered are proved to be in equilibrium. We combine theoretical proofs with particular examples that demonstrate the behavior of the strategy profiles for particular parameters.

2.2. Sharing information in an artificial society

Certain research in economics and in DAI concerns the operation of gathering information about several topics. In this section, we survey related work on this issue. The common attribute of the research below is the fact that the agents are connected in a distributed network (or society) and they learn from each other about external issues. This learning may be done explicitly, by sharing information, or implicitly, by observing actions.

Bala and Goyal [4] theoretically analyzed a model in which payoffs from different actions are unknown, and the agents decide which action to use according to their own and their neighbors' past experience. They prove that in the long run, agents belonging to the same connected society will choose actions with the same payoff. They also prove that if a 'Royal family' (a small set of agents who are observed by every agent) exists then there is a positive probability that the society will eventually choose a sub-optimal action. However, in the absence of a 'royal family', in the long run the society will choose the optimal action. This result demonstrates that distribution of information sharing is important, and the existence of a central knowledge source may cause sub-optimal results. Bala and Goyal also studied the conditions for different groups of agents to decide to take different actions (having the same payoffs) in the long run. Finally, they simulate a group of farmers learning the productivity of a new crop, in order to study the temporal and spatial patterns of diffusion.

Mor [10] developed a theoretical reputation model. In his model, there is one agent α, that plays against agents from group A. Agent α plays the Prisoner's Dilemma against a player from A, and it may defect or cooperate with the agent. An agent in A informs other agents in A when it is damaged by α. Mor proves that in such a system, beneficial defection by α is intractable, i.e., it is an NPC problem for α to find a game sequence in which it receives a higher payoff than its payoff when it always cooperates. Mor also specifies a scheme of behavior of the agents in group A in which beneficial defection by agent α is intractable. Mor assumes that the agents in A are cooperative. In our research we deal with the stability of information sharing, and this includes the case of cooperation inside group A, assuming that each of the agents in the group is self motivated.

Seredynski [21] studies a model of N-person repeated games in which the interaction between agents can be represented by a ring. It is assumed that each player acts in the game independently and selects his action to maximize his payoff. To develop a global behavior in the system Seredynski applied two Approaches; the first is a loosely coupled genetic algorithm, and the second is a loosely coupled classifier system. Seredynski applied the developed evolutionary system to solve two problems: the dynamic mapping problem and the scheduling problem.

Sen [20] developed an adaptive probabilistic policy for agents in open environments. He developed a probabilistic reciprocity scheme of strategies to be used by self-interested

agents to decide on whether or not to help other agents. Experiments show that agents can use reciprocal behavior to adapt to the environment, and improve individual performance. He showed that if the group composition changes only slowly, and there is interaction between the agents, probabilistic reciprocity based strategies can maximize the utility of each of the agents. He also found by simulations that in the long run, selfish agents perform worse than reciprocative agents in a mixed group.

Sen et al. [19] considered agents that share their opinions concerning other agents. Using simulations they showed that sharing information on experiences with other agents among reciprocative agents will limit the exploitative gains of selfish agents. They provided a trust-based evaluation function and showed that this function resists both individual and group deception on the part of selfish agents.

In our research, we consider a similar problem of agents that require the help of each other. We develop strategies for stable cooperation, and find under which conditions cooperation is possible. The agents' strategies, in contrast to the work of Sen, are deterministic, since we found that the performance is better than that of mixed strategies. Moreover, we prove theoretically that the strategies are in equilibrium, i.e., no agent can gain from deviation.

Aoyagi [1] studies a model of a two armed bandit process played by several players, where they can observe the actions of other players, but not the outcome of these actions. He proved that under a certain restriction on the probability of Distribution of the arms, the players will settle on the same arm in any Nash equilibrium of the game. This shows that each agent learns from the behavior of the other agents, even if explicit information is not delivered. In our research, we do not assume that the players can observe the actions of each other, so explicit information is required in order to learn from the other players' experiences.

Zacharia [22] investigates a mechanism called *Histos*, which is based on information sharing among human societies. The mechanism is founded on the idea that *I am probably willing to trust my friend's opinion about the (unknown) user more than the opinion of a few people I have never interacted with*. Histos uses a pairwise rating of the users. The rating is represented by a directed weighted graph. Nodes represent users and weighted edges represent reputation ratings. When a user A_0 submits a query to Histos asking about the reputation value of user A_1, the system finds all directed paths connecting A_0 to A_1, of a length less or equal to N. Then a personalized reputation value for A_1 is computed. The reputation of A_1 is a weighted average of the reputation values given to A_1 by users which are directly connected to it. The weight given to each user is its own reputation. This evaluation is based on the assumption that if somebody trusts user x as a buyer or a seller, he will also trust it as an information supplier about other users. One of the limitations of this model is the fact that people need to have incentives in order to send their evaluation of others, and a traditional reputation mechanism does not provide such incentives. In this paper, we consider the stability of sharing such information between the software agents.

In the above research, the issue of learning from each other is considered. In most of the research surveyed, it is assumed that agents explicitly share information with each other and use information from each other in order to learn common issues. In this research, we study the issue of stability of the sharing information process itself, in environments where

providing information to another agent is costly, but there is no payment mechanism and no valuations of the obtained answers.

3. Methodology

In this section, we present definitions of notions and concepts which will be used in the models that we developed in this work. We describe the basic costs and benefits due to queries answering. In addition, we present the stage game, which considers one period of the game, in which one agent asks another agent a single query, and no future encounters are considered. Then, we suggest a trigger strategy equilibrium [8] to be used by the agents in the repeated game. This kind of strategy is appropriate for cases where the action performed by one agent is unobserved by the other one, but the action yields an outcome observed by both players. However, the same outcome may be the result of different actions.

The trigger strategies are based on the ability to "punish" an agent that does not answer queries and a trigger equilibrium is based on trigger strategies. In this type of equilibrium, the agent will use the outcome of its opponent's action in order to decide whether or not to punish its opponent.

We suggest that agent B punish agent A, by ignoring a future query (or queries) of agent A. However, in order to find out the cost of such a punishment to agent A, we will first consider the discount factor of the punishment. Thus, in this section we also develop the structure of the discount factor over time, which will be used in the different models we will develop.

3.1. Notation conventions

As this paper involves several kinds of models, it includes extensive notation. In Appendix A we present a table which contains the basic notation used in this paper. In the following paragraphs, we describe some of the basic criteria used to choose these notations.

In general, we consider a pair of agents and denote them agent A and agent B. When considering an arbitrary agent, we denote it as agent i, communicating with agent $j \neq i$. Any terminology related to the expected utility of an agent is described in uppercase letters (E, U, V or F), with a subscript denoting the index of the agent for whom the expected utility is calculated, and a superscript describing the relevant model. For example, U_A denotes the expected utility of agent A in the alternating model, when it sends a query to agent B.

Parameters of the model are described in lowercase letters, English (most of the parameters) or Greek (δ, and ω), sometimes with a subscript describing the agent the parameter is associated with. Finally, macro notations, which are used only for readability purposes, are denoted by calligraphic letters, sometimes with subscripts.

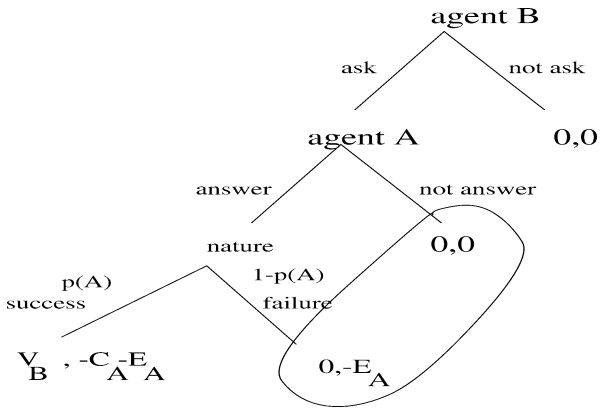

Fig. 1. Stage game: agent B sends a query, agent A decides whether to answer or not.

3.2. 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 $0 \leqslant p_i \leqslant 1$. (This probability can be calculated as the proportion of past successful queries to the agent w.r.t. all the past queries to it.) 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 incurs an obligatory cost o_i for searching for an answer when attempting to answer a query. 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 to the query. In any other case, its utility will be 0. The stage game is described in Fig. 1. As mentioned above, a list with the notations used here as well as other notations used in the rest of this paper appear in Appendix A.

Consider the one-period interaction in which agent B is ready 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.

This problem can be stated more generally. Agent j can ask agent i to perform any arbitrary action, rather than answer a query. The action is costly to agent i, and it may succeed or fail. However, if the required results of the action are not achieved, agent j cannot observe whether this happened because of a failure of the action taken by agent i,

or since agent i did not even attempt to perform the action. This problem is different from the classical *repeated principal-agent* problem [13] since each agent takes the role of *a principal* in part of the interactions, and a role of an *agent* in the other interactions. In the rest of the paper we refer to query answering, although our results are also appropriate for the general problem. In most of the paper, we consider the stage game in which agent B has a query and agent A has to decide whether to answer it or not. Of course, symmetric definitions and conclusions are appropriate for the symmetric case, where agent A has a query for agent B.

3.3. 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. We assume that if agent A is ready to ask a query when it is agent B's turn, it will ask the query somewhere else, and not wait until its next turn. In [3] we relax this assumption and assume that each agent may have a query to be asked at each time period.

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 when exactly it will be ready to send its next query, or the time its opponent will send its next query. If a query is ready, 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 $0 \leqslant \delta < 1$. We assume that interactions continue indefinitely. Our model also suits situations where in each interaction, there is a positive probability $1 - p$ that no more interactions will occur, as described in [11]. In this case, if the probability for a next interaction to occur is p, then $\delta = p$.

The configuration vector ω includes all the relevant parameters. The contents of the configuration vector are as follows:

$$\omega = (p_A, p_B, q_A, q_B, \delta, v_A, v_B, c_A, c_B, o_A, o_B),$$

where p_A is the probability for agent A to succeed in answering a query if it attempts to, p_B is the probability for agent B to succeed, q_A and q_B are the probability for agent A and agent B, respectively, to have a query in a particular time period, δ is the discount factor, v_A and v_B are the utilities of agent A and agent B from receiving an answer to a query, o_A and o_B are the cost for agent A and B when attempting to answer a query, and c_A and c_B are the additional costs whenever they succeed in answering the query. The set Ω denotes the set of all possible configuration vectors.

In this paper, we suggest a trigger strategy equilibrium [8] 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. Trigger strategies surveyed in the literature are for simultaneous games, and in most of the cases for the prisoner's dilemma. We apply this type of strategies to the queries answering problem, which is a nonsimultaneous game.

4. A one-period observation model

In this section, we consider a strategy profile where punishment is performed after each time an answer is not obtained from the opponent, though in some cases the outcome is not deliberately caused by the opponent. 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.

4.1. 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 $j \neq i$ 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 provided each agent answers its opponent's query. Given phase *Normal*, when an agent i does not answer a query, the agents switch to $Punish_i$. This punishment phase holds until agent i answers a query of agent j, in which case, the agents return to phase *Normal*. This strategy profile promotes cooperation and information sharing.

4.2. Expected discount over time

In this section, we discuss the discount factor of the expected utility over time. This discussion will be valid for the one-period observation model, and also for the n-period observation and the k-model of punishment after k unanswered queries from n. $\mathcal{D}_i$ is the expected discount ratio from the time agent j asks a query, until the time agent i will be ready to ask a query. With a probability of q_i the delay will be for one time period, in which case, a discount of δ should be considered. With a probability of $q_i(1-q_i)$ there will be a delay of two time periods (a discount of δ^2), and with a probability of $(1-q_i)^k q_i$ there will be a delay of $k+1$ periods, and the discount ratio will be δ^{k+1}. $\mathcal{D}_i$ is the sum of the above infinite geometric series. Thus, the value of $\mathcal{D}_i$ is

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

We proceed by evaluating the discount ratio in a case of punishment. Consider a case in which agent A has to be punished. This means that the last query was sent from agent B,

and no response was obtained from agent A. Since the agents *alternate* their queries, the next query will be sent by agent A, but it will be ignored by agent B. The consequent query will be sent by agent B, and agent A will attempt to answer it, as defined in Section 4.1.

Denote the present time when agent A does not send an answer t_0, the time when agent A has a query t_A, and the time after t_A in which agent B sends a query to agent A, t_B. Denote the overall expected discount ratio from t_0 until t_B, $\mathcal{D}$. The minimal value of $\mathcal{D}$ is in the case where $t_B = t_A + 1$. Namely, immediately after the period in which agent A asks a query which is ignored, agent B asks a query back. This event occurs with a probability of q_B, and in this case, the total discount factor will be $\mathcal{D} = \mathcal{D}_A\delta$. With a probability of $q_B(1-q_B)$, the delay will be for two time periods. In this case, $\mathcal{D} = \mathcal{D}_A\delta^2$. In the general case, with a probability of $q_B(1-q_B)^i$, the delay will be for $i+1$ time periods, and the discount factor will be $\mathcal{D} = \mathcal{D}_A\delta^{i+1}$.

Again, we obtain a geometric series of the expected discount ratio:

$$\begin{aligned}\mathcal{D} &= \mathcal{D}_A\delta q_B + \mathcal{D}_A\delta^2(1-q_B)q_B + \mathcal{D}_A\delta^3(1-q_B)^2q_B + \cdots \\ &= \mathcal{D}_A\frac{\delta q_B}{1-\delta(1-q_B)}\end{aligned}$$

and according to Eq. (1), this is equal to

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

Symmetrically, $\mathcal{D}$ is also the expected discount in the case of punishing agent B. In the following lemma, we prove that $0 \leqslant \mathcal{D}_i < 1$, and also that $0 \leqslant \mathcal{D} < 1$.

Lemma 4.1. *Given* $0 \leqslant \delta < 1$, *for each* $i \in \{A, B\}$, *if* $0 \leqslant q_i \leqslant 1$, *then* $0 \leqslant \mathcal{D}_i < 1$, *and if* $0 \leqslant q_A, q_B \leqslant 1$, *then* $0 \leqslant \mathcal{D} < 1$.

The proof of this lemma, as well as the proofs of the lemmas and theorems in the rest of this paper, appear in Appendix A. The terms $\mathcal{D}$, $\mathcal{D}_A$ and $\mathcal{D}_B$ will be used in the rest of this paper, to calculate the expected utility from the next interaction. The expected utility of a value v obtained in the next interaction will be v multiplied by the appropriate $\mathcal{D}$ or $\mathcal{D}_i$, since $0 \leqslant \mathcal{D} \leqslant 1$, $\mathcal{D}$ and $\mathcal{D}_i$ can be considered standard discount factors.

4.3. Expected utility

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 in equilibrium. The expected utility of each agent is based on the fact that the agents follow the equilibrium strategies. In the following, we define the terms that will be used for these specifications.

Definition 4.1. 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 4.1. *The values of V_A, U_A, and F_A are computed as follows.*

$$V_A = -o_A + p_A(-c_A + \mathcal{D}_A U_A) + (1 - p_A)F_A, \tag{3}$$

$$U_A = p_B(v_A + \mathcal{D}_B V_A) + (1 - p_B)\mathcal{D}U_A, \tag{4}$$

$$F_A = \mathcal{D}V_A. \tag{5}$$

V_A is the expected utility of agent A from attempting to answer a query. It consists of the obligatory cost o_A, and the expected future utility when the attempt to answer succeeds, and the expected utility when it fails, with the corresponding probabilities for both events. 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.

4.4. Equilibrium conditions

We proceed by identifying the conditions under which the trigger equilibrium exists. In particular, we use the strategy profile defined in the beginning of Section 4.1, and we specify the condition under which an agent prefers the trigger strategy over deviation and ignoring its opponent's queries, given that the second agent uses the trigger strategy. If the condition of each agent holds, then the trigger strategy profile is in equilibrium.

In order to prove that the strategy profile is in equilibrium, we have to prove that whenever agent i follows its equilibrium strategy, it is worthwhile for agent j to keep its equilibrium strategy too. In the following lemma, we prove under which condition this holds.

Lemma 4.2. *The trigger equilibrium of the one-period observation strategy profile is in equilibrium if* $V_i \geqslant F_i$, *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 4.3. *The expected utility of agent A when attempting to answer a query, can be formalized as follows.*

$$V_A = \frac{-o_A - p_A c_A + p_A \mathcal{D}_A \mathcal{A}}{1 - p_A \mathcal{D}_A \mathcal{B} - (1 - p_A)\mathcal{D}}, \tag{6}$$

where

$$\mathcal{A} = \frac{p_B * v_A}{1 + \mathcal{D}(p_B - 1)} \tag{7}$$

and

$$\mathcal{B} = \frac{\mathcal{D}_B * p_B}{1 + \mathcal{D}(p_B - 1)}. \tag{8}$$

After manipulating, the expected utility can be formalized also as follows:

$$V_A = \frac{(1 + \mathcal{D}p_B - \mathcal{D})(-o_A - p_A c_A) + p_A \mathcal{D}_A p_B v_A}{(1 + \mathcal{D}p_B - \mathcal{D})(1 - \mathcal{D} + \mathcal{D}p_A) - p_A \mathcal{D} p_B}. \tag{9}$$

Using Lemma 4.2, and the definitions of F_A and F_B, we can progress by finding the explicit conditions for the existence of the trigger equilibrium. The condition of agent i can be displayed as a required ratio between v_i and $c_i + o_i/p(i)$ for agent $i \in \{A, B\}$. We start with two propositions. First, we prove that the denominator of V_A is positive.

Proposition 4.1. *The term* $1 - p_A \mathcal{D}_A \mathcal{B} - \mathcal{D}(1 - p_A)$ *is positive whenever* $0 \leqslant \mathcal{D}_A, p_A \leqslant 1$.

We proceed by proving the conditions required for the equilibrium to hold.

Lemma 4.4. *If the agents are risk-neutral, then the one-period observation strategy is in equilibrium for agents A and B, if the following condition holds for both* $i = A$, $j = B$ *and* $i = B$, $j = A$.

$$\frac{v_i}{c_i + o_i/p_i} \geqslant \left(\frac{1 - \delta + \delta q_i}{\delta q_i p_j} + \frac{\delta q_j (p_j - 1)}{p_j (1 - \delta + \delta q_j)} \right). \tag{10}$$

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.

4.5. 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. According to Lemma 4.2, the value $V_A - F_A$ 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 our conclusions are proved formally, while others are demonstrated for a particular configuration of parameters $\omega = (q_i = 0.1, p_i = 0.5, \delta = 0.9, c_i = 1, o_i = 0.1, v_i = 10)$. In the following lemma, we prove the influence direction of p_i and of the cost and benefit parameters of the utilities and of the equilibrium conditions.

Lemma 4.5. *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. That is, as v_A, the benefits an agent obtains from answering queries, increases, the utility of agent A increases, and it is more worthwhile for it to answer queries. It is also expected that the directions of the 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 in answering more queries, its utility increases. As p_B increases, agent B succeeds in answering 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.

In the following lemma, we prove the influence direction of δ, q_A and q_B, both on the expected utility of the agent and on its tendency to use the trigger strategy.

Lemma 4.6. *As δ, q_A or q_B increases, the expected utility of each agent increases, and the trigger equilibrium exists for more configurations.*

As δ increases, the expected utility of agent A also increases, as well as its tendency to attempt to answer agent B's query, because 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.

As q_A increases, agent A is supposed to ask queries more frequently, consequently its utility from receiving answers increases. Thus, it is more beneficial for it to answer other agents' 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. However, as we proved in Lemma 4.6, the alternating effect is stronger, and as q_B increases, the expected utility increases and the equilibrium exists more frequently.

In [3] we also 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. 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 that is presented in Section 5, with the change in the length of the history that is taken into consideration.

5. A model with n observation periods

The equilibrium structures described in the previous section enable the agents to share information with each other, due to the fact that an agent that did not respond to a query, will be punished by the sender of the query. However, an agent is not punished only when it ignores a query. Any event of a query with no response yields a punishment, regardless if this was caused by its ignoring the query, or because of a failure to answer the query after a costly attempt.

One can suggest a refined strategy profile, where punishment is done only after a given number of queries with no response. The benefits of such a protocol are based on the fact that punishment will be done more rarely. Such a protocol is more fair, since it reduces the probability of punishing an agent that attempts to answer all the queries it receives. Increasing the number of failures required for punishment also increases the average utility of the agents. However, since the threat to punish is weakened, the equilibrium based on this approach is weaker than the equilibrium based on punishment after each missing response. This means that as the number of periods required in order to decide about a punishment increases, there are more situations where an agent can beneficially ignore queries.

We consider three variations of the n-period observation model. First, for demonstration, in Section 5.1 we present a model where $n = 2$, and we compare this model to the one-period model ($n = 1$) discussed in Section 4. Second, in Section 5.2, an agent is punished after n consequent queries with no answer by this agent. Finally, in Section 5.3, punishment is implemented after k unanswered queries, from the n last queries to that agent.

In this paper, we consider only pure trigger strategies for these cases, whereas in [3] we considered a mixed strategy profile. In the mixed strategy an agent i randomly decides whether or not to attempt to answer a query for some histories. There are two situations in which mixed actions can be used: (a) in a punishment phase, where agent i is allowed to punish agent j; (b) in the normal phase, where 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 the conditions for the existence of the equilibrium 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 corresponding pure strategies. Thus, mixed strategies are not recommended for use in our model and throughout the rest of this paper we refer only to the pure strategies.

In order to consider the last n periods when deciding on a punishment, the agent should save the history of the last $n-1$ results (answered or ignored) of queries sent to its opponent, and add the result of the present queries. We denote by h_i the results of the $n-1$ last events when agent i was required to answer queries. The history of agent i is composed as follows: $h_i = (h_i(n-1), \ldots, h_i(2), h_i(1))$, where $h_i(1)$ represents the last event of a query sent to agent i. $h_i(k) = 0$ if k's last query to agent i received no answer, and $h_i(k) = 1$ if 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 $((1, \ldots, 1), (0, \ldots, 0))$, 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, $h_i \ll \textit{new\_event}$, means deleting the oldest event in h_i, and adding a new event to h_i. Finally, the function $\textit{zero}(h_i)$ returns true if all the events in h_i are unanswered queries. Formally, $\textit{zero}(h_i) = \textit{true}$ if $\sum_{k=1}^{n-1} h_i(k) = 0$. Using these notations, we proceed with describing and analyzing the two variations of the n-period model.

5.1. Punishment after two unanswered queries

In this section, we consider a special case of punishment after two consequent periods of no response from an agent. If a two-period strategy profile is in use, the agent will be concerned with both its own history, and the history of its opponent, when deciding whether to answer a query or not. If no punishment mode is reached, then it will only be concerned with the last event in the history of both. If its last event was an unanswered query, then if it will also not answer in the current period (either because of a failure or because of ignoring the query), an immediate punishment will be implemented. In contrast, if the last event in its history contains a successful response, then an unanswered query at this time may be forgiven, and in some cases, it may be beneficial for the agent to ignore queries in this state.

We denote a successful event 1 and an unanswered query event 0. Since we are interested in the last event of each of the servers, we denote the current state (a, b). $a \in \{0, 1\}$ is the last event of agent i (1 if agent A succeeds in answering, 0 if it did not send an answer). Similarly, $b \in \{0, 1\}$ is the last event of agent B.

5.1.1. Structure of the utility function

In order to analyze the two-period model, we will use the same notation as used in Section 4. However, we use additional terms with respect to the expected utility of the agents, since there are four different possible states, as described below, and for each of them, the expected utility will be slightly different.

Denote by V_A^{xy} the expected utility of agent A, when the last event performed by agent A is x, and the last event performed by agent B is y, and it is now agent A's turn to answer a query. However, none of the agents are in a punishment phase. Since the agents

are assumed to hold the strategies defined above, agent A will attempt to answer the query. Denote by U_A^{xy} the expected utility of agent A, when the last event performed by agent A is x, the last event performed by agent B is y, and it is now agent B's turn to answer agent A's query. Again, both agents are not in a punishment phase. Thus, agent B will attempt to answer agent A's query.

In the following formulas we specify the expected utility for each of the 4 cases. V_A^{11} is the utility of agent A when a query was sent to it, and the last event of both agents was a success. With a probability of p_A, agent A will succeed in answering, and with a probability of $1 - p_A$, it will fail. In both cases it incurs a cost of o_A, and in the case of success, it incurs a cost of c_A. The actual event, success or failure, determines agent A's history when continuing to the next time period, where agent A sends a query.

$$V_A^{11} = -o_A + p_A\left(-c_A + \mathcal{D}_A U_A^{11}\right) + (1 - p_A)\mathcal{D}_A U_A^{01}. \tag{11}$$

V_A^{10} is the expected utility of agent A when it has to answer a query, and its last event was a success in answering agent B's query, but it did not receive an answer to its last query from agent B. The explanation is similar to that of V_A^{11}.

$$V_A^{10} = -o_A + p_A\left(-c_A + \mathcal{D}_A U_A^{10}\right) + (1 - p_A)\mathcal{D}_A U_A^{00}. \tag{12}$$

In states $(0, 0)$ and $(0, 1)$, the last query sent to agent A is not answered. Thus, if for the current query it will also not receive an answer, a punishment is implemented. The punishment is skipping its turn to obtain an answer to a query, as in the one stage alternating model.

$$V_A^{00} = -o_A + p_A\left(-c_A + \mathcal{D}_A U_A^{10}\right) + (1 - p_A)\mathcal{D}_A\mathcal{D}_B V_A^{00} \tag{13}$$

and finally,

$$V_A^{01} = -o_A + p_A\left(-c_A + \mathcal{D}_A U_A^{11}\right) + (1 - p_A)\mathcal{D}_A\mathcal{D}_B V_A^{01}. \tag{14}$$

The following formulas specify the utility of agent A when it is agent B's turn to answer its queries. Again, the utility depends on the history of the last events of both agents.

$$U_A^{00} = p_B\left(v_A + \mathcal{D}_B V_A^{01}\right) + (1 - p_B)\mathcal{D}_A\mathcal{D}_B U_A^{00}, \tag{15}$$

$$U_A^{10} = p_B\left(v_A + \mathcal{D}_B V_A^{11}\right) + (1 - p_B)\mathcal{D}_A\mathcal{D}_B U_A^{10}, \tag{16}$$

$$U_A^{01} = p_B\left(v_A + \mathcal{D}_B V_A^{01}\right) + (1 - p_B)\mathcal{D}_B V_A^{00}, \tag{17}$$

$$U_A^{11} = p_B\left(v_A + \mathcal{D}_B V_A^{11}\right) + (1 - p_B)\mathcal{D}_B V_A^{10}. \tag{18}$$

The above recursive equations can be solved in order to find a solution which gives the complete description of the utility in each state. We solved the equations using Maple, and found the explicit expected utility, and the conditions for an equilibrium to exist. However, since the results contain a very complex structure of the solution we refrain from presenting them here.[1] In the next section we demonstrate the behavior of the expected utility and the conditions for the equilibrium, and compare them with that of the one-period model.

[1] A full version including the explicit expected utility is available in http://www.cs.biu.ac.il/~schwart/articles/rep-full.ps.

5.1.2. Properties of the model

Intuitively, for most of the possible configurations, if punishment is done after two periods of observation instead of after one period, then the expected utility of an agent is higher, assuming that both agents follow the trigger strategies. This is because punishment is caused only in situations that an agent fails to answer queries in two consecutive periods (under the assumption that it always attempts to answer). The probability for such an event is much lower than the probability for one failure, which causes punishment to the agent in the one-period model. It is also intuitive that in the current model the motivation to follow the trigger strategy is much lower, since deviation from equilibrium does not always yield immediate punishment.

In order to compare the equilibrium conditions in the two-period model with that of the one-period model, we can focus on state $(1,0)$, where, assuming that in the past both agents used the trigger strategies, agent A succeeds in answering the last query of agent B, while agent B fails to answer the last query of agent A. We check whether it is worthwhile for agent A to answer agent B in this turn. In particular, we verify whether

$$V_A^{10} - \mathcal{D}_A U_A^{00} \geqslant 0$$

i.e., whether the utility of attempting to answer a query in state $(1,0)$ is higher than the utility of ignoring the query.[2] We denote the equilibrium existence index $V_A^{10} - \mathcal{D}_A U_A^{00}$. Whenever the index value is positive, an equilibrium exists for this set of parameters. As the value of the index increases, an equilibrium exists for more sets of parameters. We also define the equilibrium existence index of the one-period observation model as $V_A - \mathcal{D} * V_A$. Whenever this index is positive, an equilibrium exists. As the index value increases, the equilibrium becomes stronger, and holds for more values of the other parameters.

In order to evaluate the expected utility, we use V_A for the one-period observation model, and we use V_A^{11} for the two-period model, since we consider a situation with no failure or deviating in the past. In the following paragraphs we demonstrate the influence of several parameters on the expected utility and on the equilibrium conditions for a particular configuration of parameters. The basic configuration is $\omega = (q_i = 0.5, p_i = 0.5, \delta = 0.9, c_i = 0.1, o_i = 0.1, v_i = 10)$.

Influence of v_A, c_A and o_A. First, we consider the influence of c_A and v_A on the expected utility. As explained above, we assess the influence on V_A^{11}. As expected, the increase of v_A and the decrease of c_A cause an increase of V_A^{11}. We illustrate this influence in Fig. 2.

In Fig. 3, we present the values of both V_A^{11} and V_A of the one-period model, for different values of v_A, assuming $c_A = 1$. We revealed that V_A^{11} is higher than V_A, and the difference increases as v_A increases. We can see that for v_A near zero, the curves are crossed. For the above set of parameters, the expected utility in both models is equal for both models when $c_A = 3.296299245 * v_A$. However, if such a relation exists between c_A and v_A, then equilibria of both types will not exist. If the cost of answering a query is more than the benefit of a possible answer in the future, then an agent has no motivation to answer queries.

[2] Proof of why it is sufficient to consider state $(1,0)$ in order to prove the existence of an equilibrium is provided for the n-period case in Lemma 5.1.

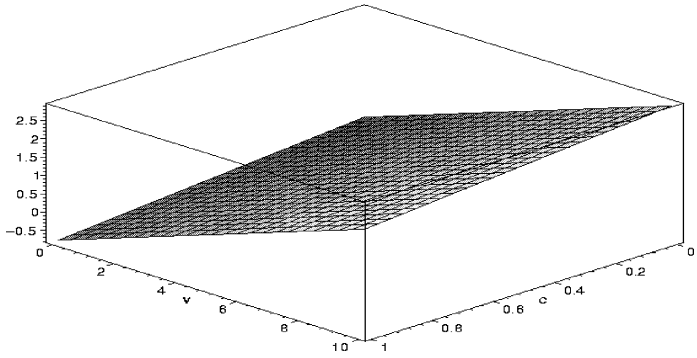

Fig. 2. Two-period observation model: V_A^{11} as a function of v and c.

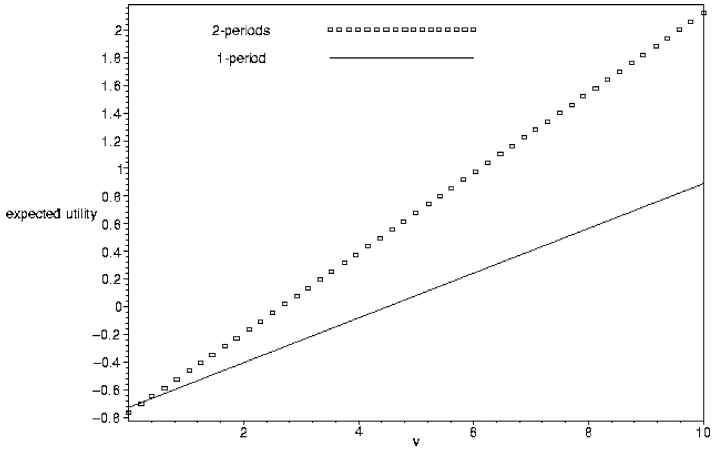

Fig. 3. Two-period observation model: V_A^{11} and V_A as functions of v_A (given $c_A = 1$).

Next, we checked the influence of c_A and v_A on the equilibrium existence index for the two-period model. The results are presented in Fig. 4. It is clear that, as in the one-period model, as c_A and o_A increase, and as v_A decreases, the equilibrium exists for a larger set of configurations. In Fig. 5 we compare the equilibrium existence index of the two-period model with that of the one-period model, for different values of v_A, and for $c_A = 1$. We can see that for these values, equilibrium of the two-period model does not exist, while equilibrium of the one-period model exists from a given value of v_A. For the above sets of parameters, we checked for which values of v_A and c_A equilibrium exists, and we found that the equilibrium of one-period model exists for $v_A \geqslant 4.498245616 * c_A$, while the equilibrium of the two-period model exists only for $v_A \geqslant 80.15503155 * c_A$.

Influence of δ. In order to demonstrate the influence of δ on the expected utility of the agents and on the strength of the equilibrium, we set c_A as 0.1. This enabled equilibrium

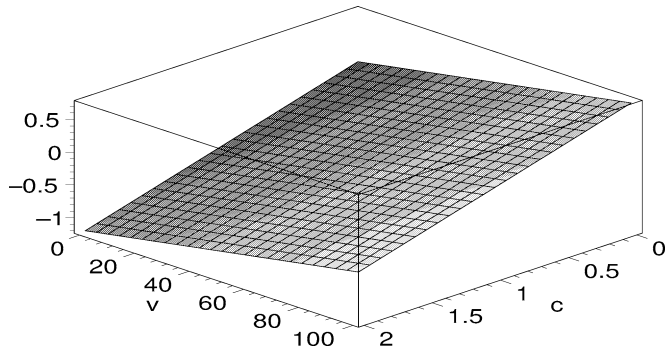

Fig. 4. Two-period observation model: the influence of v_A and c_A on the equilibrium existence index $V_A^{10} - \mathcal{D}_A U_A^{00}$.

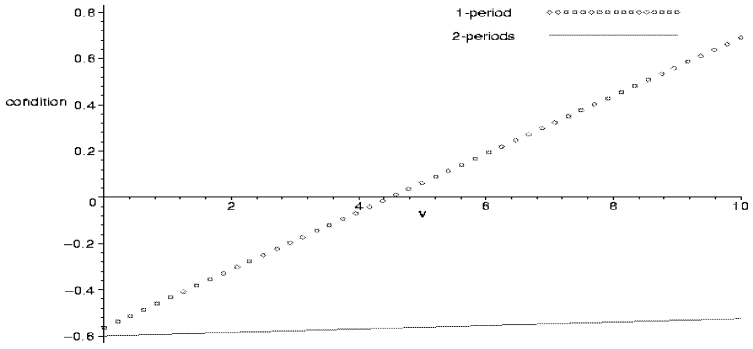

Fig. 5. Two-period observation model: the influence of v_A on the equilibrium existence index $V_A - \mathcal{D}V_A$ (one-period model) and on $V_A^{10} - \mathcal{D}_A U_A^{00}$ (two-period model).

to exist for different values of δ also for the two-period model, though for $c_A = 1$, as previously determined, equilibrium does not hold.

Thus, the set of parameters we used in the following figures is $q_A = q_B = 0.1$, $p_A = p_B = 0.5$, $c = 0.1$, $o = 0.1$ and $v = 10$.

The influence direction of δ on the expected utility and on the equilibrium existence index of the two-period model is, as expected, similar to the influence of the direction in the one-period model: As δ increases, the expected utility of the agent also increases, as well as the set of configurations for which the equilibrium conditions hold. (The interesting phenomena, as demonstrated below, is the clear influence of δ on the difference between the utilities of both models, and on the difference between the equilibrium existence indices of both models.) Fig. 6 shows V_A^{11}, V_A^{01}, and V_A, as functions of δ.

The expected utility for $\delta = 0$ is the same for the three functions. This can be explained, since the expected utility for $\delta = 0$, where the agents do not regard the future expected utility, contains only the current costs of c_A and o_A. As δ increases, the future becomes more important, so the difference between the curves also increases.

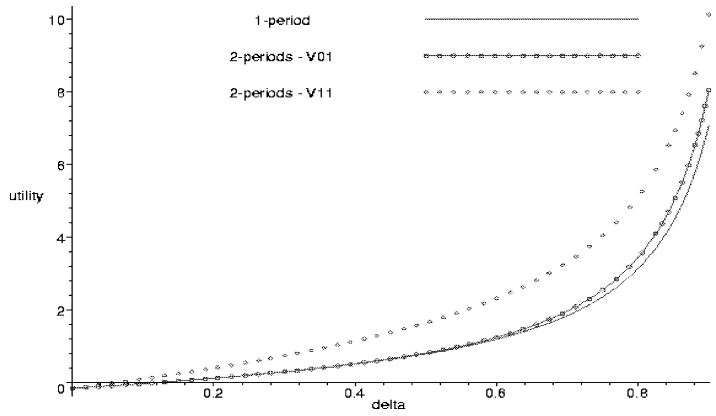

Fig. 6. Two-period observation model: the influence of δ on the expected utility in the two-period model and in the one-period model.

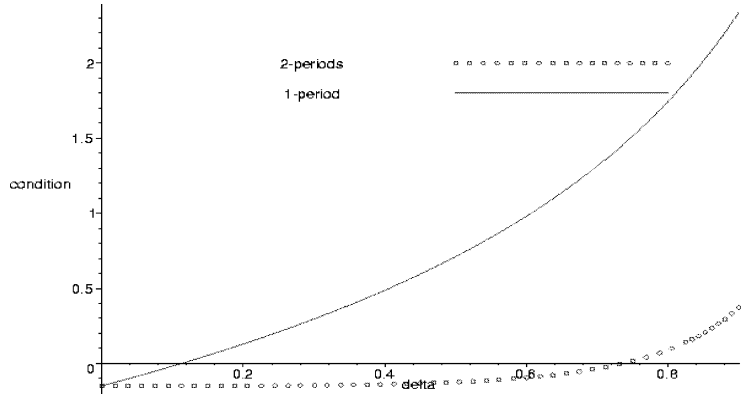

Fig. 7. Two-period observation model: the influence of δ on the equilibrium existence index in state $(1, 0)$.

Our next set of figures demonstrates the influence of δ on the value of the equilibrium existence index. Fig. 7 demonstrates that as δ increases, the equilibrium becomes stronger. This is intuitively clear, since as δ increases, the future is more important, so an equilibrium exists more frequently. For small values of δ, the value of the equilibrium existence index is negative, i.e., the trigger equilibrium does not exist for the considered parameters profile. The equilibrium existence index in the one-period model and the index in the two-period model are shown in Fig. 7. They give the same (negative) value for $\delta = 0$, but as δ increases, the condition tends to hold more for the one-period model, and the difference between the equilibrium existence indices increases as δ increases. The reason being that in the one-period model, punishment is performed more often in the future, so as δ increases, and the future becomes more important, the punishment becomes more threatening.

Influence of q_A and q_B. Fig. 8 demonstrates the influence of q_A and q_B on the expected utility, and Fig. 9 shows their influence on the tendency to answer queries. We found that increasing the tendency to ask queries (q_A) or to receive queries (q_B) increases the util-

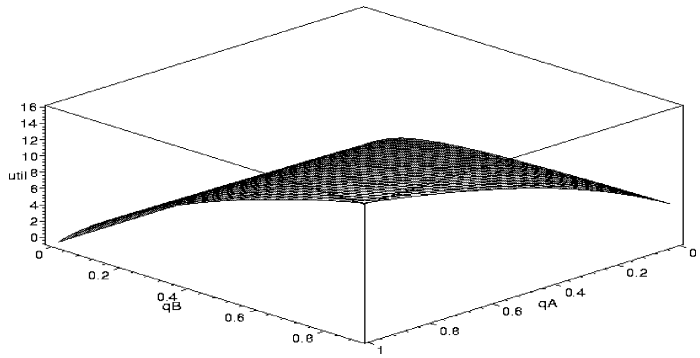

Fig. 8. Two-period observation model: the influence of q_A and q_B on the expected utility in the two-period model.

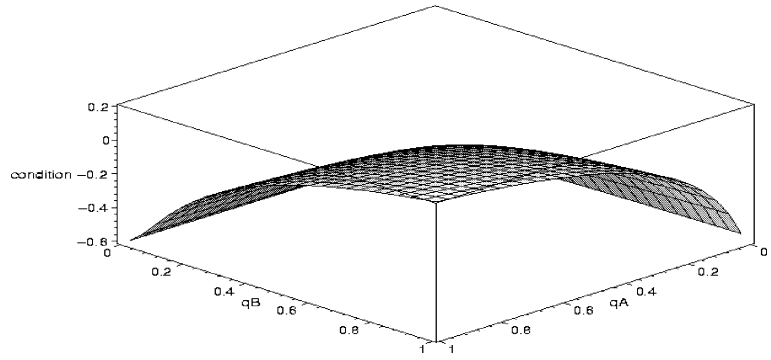

Fig. 9. Two-period observation model: the influence of q_A and q_B on the equilibrium existence index in state $(1, 0)$.

ity of the agents and its tendency to answer queries. These results are intuitively clear, as explained in the one-period model.

We also found that as q_A or q_B increases, the difference between the expected utility in the two-period model and the expected utility in the one-period model increases. In other words, as the frequency of queries increases, the expected utility in the two-period model is influenced more than the expected utility in the one-period model. The intuition for this result is similar to the intuition for the influence of δ on the difference between these two models. As q_A or q_B increases, the future obtains a higher weight since it depends on the difference in punishments in future interactions. Thus, the difference between the one-period model and the two-period model increases.

Influence of p_A and p_B. The influence of p_A and p_B on the expected utility is demonstrated in Fig. 10. The influence is clearly positive, for the same reasons as explained in the one-period model. Fig. 11 shows the influence of p_A and p_B on the conditions. It is clear from the figure that as p_B increases, agent A will tend to attempt to answer queries, since the probability of its own queries to be answered increases.

However, the influence of p_A on the equilibrium existence index depends on the entire environment configuration, and this is also demonstrated in Fig. 12. In this figure, the

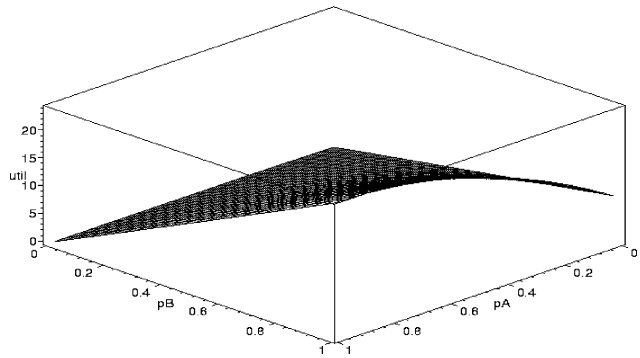

Fig. 10. Two-period observation model: the influence of p_A and p_B on the expected utility in the two-period model.

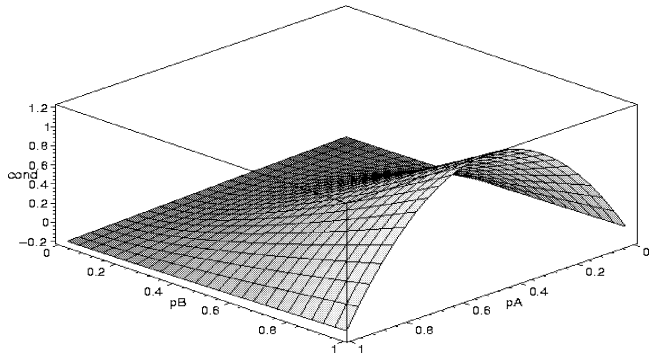

Fig. 11. Two-period observation model: the influence of p_A and p_B on the equilibrium existence index in state $(1, 0)$.

equilibrium existence index is shown as a function of p_A for both the two-period model (the descending line) and the one-period model (the ascending line), while p_B is fixed at 0.5.

We can see that while in the one-period model, as p_A increases, an equilibrium exists more often, the influence of p_A on the equilibrium in the two-period model depends on the environment. For $p_B = 0.5$, an increase of p_A causes the equilibrium existence index to increase until $p_A = 0.472$, but from this maximum point, the influence of p_A becomes negative. The explanation for this phenomenon is that for higher values of p_A, the threat of punishment due to ignoring one query decreases, given state $(1, 0)$. The reason for this reduction is that punishment of agent A will be performed only if the next query to agent A will not be answered, and this probability decreases as p_A increases. Thus, for high values of p_A, as p_A increases, there may be more environments in which agent A can allow itself to ignore queries, given that no immediate punishment will be performed.

When comparing V_A^{11} with V_A in the examples we checked, as predicted, we found that the expected utility when using the two-period observation model is higher than the expected utility in the one-period alternating model. This remains true even when comparing V_A^{01} and V_A.

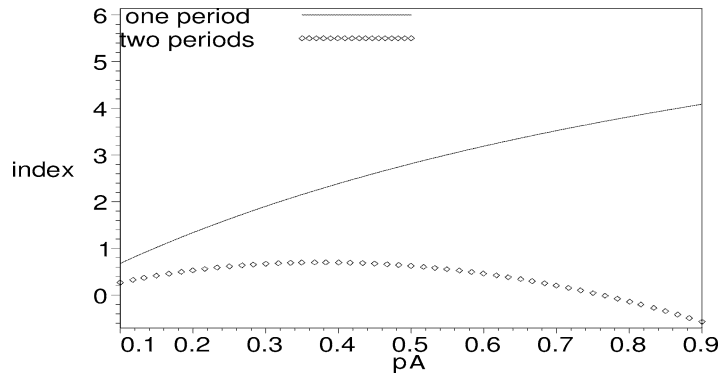

Fig. 12. The influence of p_A on the equilibrium existence index in the two-period model, where $p_B = 0.5$.

Nevertheless, in our testing we found that $V_A^{01} \geqslant V_A$ for different values of parameters. In other words, if we compare the expected utility of an agent after one failure in the two-period model, the expected utility is still higher than in the one-period model, for the values of parameters where equilibrium exists.

However, in both cases failure of agent A to answer the current query will yield an immediate punishment. This phenomenon can be explained by the fact that the utility function of the agent is recursive and it also includes future events: even if in the next period, success of agent A will cause agent B to attempt to answer it, and failure to answer will cause agent B to ignore its query, the utility function also includes future events, in which the two-period model yields a higher expected utility than the one-period observation model.

5.1.3. Summary

Nevertheless, the two-period strategy profile, although yielding a higher expected utility, has a significant disadvantage. There are much more situations where it is not in equilibrium. In fact, the basic profile we used for demonstration in the previous sections of this paper, $\delta = 0.9$, $q_A = q_B = 0.1$, $p_A = p_B = 0.5$, $v_A = 10$, $c_A = 1$, $o_A = 0.1$, was appropriate for the one-period profile, but the two-period profile was not in equilibrium for these parameters.

The main conclusion of the above observation is that the agents' designers, when designing information sharing agents, should consider the environment parameters and decide which model to use according to the configuration parameters. This conclusion can be generalized for the n-period model presented below.

5.2. Equilibrium with punishment after n unanswered queries

In this section, we analyze a model in which punishment of an agent is performed after n consecutive events of queries with no responses, assuming an alternating queries model. The strategies and phases of the n-period model are defined as in Section 4.1, but moving from phase *Normal* to phase *Punish*$_i$ will occur only after n consecutive queries with no response from agent i. A strategy is an *n-period 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 to a query from agent j. Denote by $\Omega_n \subseteq \Omega$ the set of all $\omega \in \Omega$, such that the pair of n-period strategies is in equilibrium given configuration ω.

5.2.1. Expected utility

Assuming that both agents use their n-period strategies, $V_A^{n,h}$ is the expected utility of agent A when it obtains a query from agent B given history h. 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 the *Normal* phase:

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

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

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

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 consecutive failures. Finally,

$$pun_A(h) = -o_A + \mathcal{D} V_A^{n,(h_A \ll 0, h_B)}$$

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:

$$V_A^{n,h} = \begin{cases} p_A \cdot suc_A(h) + (1 - p_A) pun_A(h) & zero(h_A) = true, \\ p_A \cdot suc_A(h) + (1 - p_A) fail_A(h) & \text{otherwise.} \end{cases} \tag{19}$$

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, which 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,

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

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

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

is A's utility when agent B fails to answer A's query, but punishment of B is not required, and $pun_B(h) = \mathcal{D} U_A^{n,(h_A, h_B \ll 0)}$ is A's utility when punishing agent B is required. Using the above, the expected utility of agent A, when it forwards a query to agent B, given n and h, denoted $U_A^{n,h}$, is defined as follows:

$$U_A^{n,h} = \begin{cases} p_B \cdot suc_B(h) + (1 - p_B)pun_B(h) & zero(h_B) = true, \\ p_B \cdot suc_B(h) + (1 - p_B)fail_B(h) & \text{otherwise.} \end{cases} \tag{20}$$

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 $U_A^{n,h}$. 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. The dynamic programming method is based on filling the value of $V_A^{n,h}$ and $U_A^{n,h}$ for each possible history. In the n-period model, all histories with a last success are equivalent (since punishment will be performed after n unanswered queries), and all the histories with last k failures are equivalent (since punishment will be performed after an additional $n - k$ unanswered queries). Thus, the state, for which the utilities and the equilibrium conditions are defined, is the number of last consecutive failures. Thus, only $n + 1$ different states have to be observed, for each possible number of last subsequent failures, from 0 to n. Since there are two agents, there are $(n + 1)^2$ different states to be considered. The utility of both agents will be evaluated for each state.

5.2.2. Properties of the n-period model

In the following paragraphs, we analyze important properties of the n-period 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-period strategy equilibrium.

In this section we use the following notations. Denote by $exp\_u_i(\omega, n, history)$ the expected utility of agent i from a model in which punishment is imposed after n periods, given *history* of n last periods for each agent. Given configuration ω, $history = (hist_A, hist_B)$, and the n-period observation model, the agent can evaluate the difference between its utility after answering a query, and its utility after ignoring a query. Formally, denote by $ignr\_loss_i(\omega, n, history)$ the loss of agent i when it does not answer a query, w.r.t. the case when it does answer. The definition of $ignr\_loss_A(\omega, n, history)$ for the model of punishment after n consecutive failures, is as follows:

$$ignr\_loss_A\big(\omega, n, (hist_A, hist_B)\big) = \begin{cases} \mathcal{D}_A(v_A + exp\_u_A(\omega, n, (hist_A \ll 1, hist_B)) \\ \quad - exp\_u_A(\omega, n, (hist_A \ll 0, hist_B))) & zero(hist_A) \neq true, \\ \mathcal{D}_A(exp\_u_A(\omega, n, (hist_A \ll 1, hist_B)) \\ \quad - exp\_u_A(\omega, n, (hist_A \ll 0, hist_B))) & \text{otherwise.} \end{cases} \tag{21}$$

This means that the loss of agent A from avoiding answering a query is an immediate punishment, if it is required, but it also contains the future losses due to the effect of this unanswered query in the future. In order to prove our claims about the n-period strategy profile we start with an auxiliary lemma that will help us reveal when the equilibrium conditions hold.

Lemma 5.1. *Consider a trigger equilibrium based on the n-period strategy profile. The equilibrium will exist, if it is worthwhile for agent A to attempt to answer agent B after a history of* $((1,\ldots,1),(0,\ldots,0))$, *and if it is worthwhile for agent B to attempt to answer agent A after a history of* $((0,\ldots,0),(1,\ldots,1))$.

After a history of $((1,\ldots,1),(0,\ldots,0))$, 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 $((0,\ldots,0),(1,\ldots,1))$, then it will be worthwhile for it to do so after any other history. Based on Lemma 5.1, in order to determine whether or not an n-period equilibrium exists, we only need to consider the $((1,\ldots,1),(0,\ldots,0))$ history of agent A, and the $((0,\ldots,0),(1,\ldots,1))$ history of agent B. Based on this reasoning, in order to check whether an equilibrium exists or not, we only have to check the $((1,\ldots,1),(0,\ldots,0))$ condition. We will use this attribute in order to prove that $\mathit{ignr\_loss}_A(\omega,n,((1,\ldots,1),(0,\ldots,0)))$ increases as n decreases. When we consider an $n+1$-period model, the history includes $2n$ events instead of $2(n-1)$.

However, in order to compare $\mathit{ignr\_loss}_A$ of the n-period observation model with that of the $n+1$, we check when an equilibrium exists given a history of $((x,1,\ldots,1),(1,0,\ldots,0))$ (at least $n-1$ successful events of agent A, and exactly $n-1$ failure events of agent B) when considering the $n+1$-period model. This is because a history with n consequence failures will cause an immediate punishment of agent B. In the following lemma, we prove that as n increases, the loss of an agent from ignoring a query decreases.

Lemma 5.2. *Given a history of exactly $n-1$ consecutive queries to agent B with no answer, and at least $n-1$ consecutive successes of agent A,*

$$\mathit{ignr\_loss}_A\big(\omega,n+1,((x,1,\ldots,1),(1,0,\ldots,0))\big) < \mathit{ignr\_loss}_A\big(\omega,n,((1,\ldots,1),(0,\ldots,0))\big).$$

In addition,

$$\mathit{ignr\_loss}_A\big(\omega,n+1,((x,1,\ldots,1),(1,0,\ldots,0))\big) \leqslant (1-p_A)\mathit{ignr\_loss}_A\big(\omega,n,((1,\ldots,1),(0,\ldots,0))\big).$$

In the following lemma we prove that there are more configurations in equilibrium in the n-period observation strategies than in the $n+1$-period observation strategies. In other words, as we observed when comparing the models of $n=1$ and $n=2$, as n increases the equilibrium conditions become more restrictive. This will be proven in the following lemma.

Theorem 5.1. *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 lemma 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 implemented 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. If an n-period equilibrium does not exist, the agents should reduce n, until they obtain n' for which n'-period equilibrium does exist. They should find the largest possible n', since, as proven in Theorem 5.2, increasing n increases the expected utility of the agents. In the theorem below, we prove our main claim about the change in the agents' utilities: as n, the number of periods considered to decide about a punishment, increases the agent's average utility also increases.

More generally, we can show that for each configuration of parameters we can find the largest n, such that for any smaller $n' < n$, answering will be in equilibrium for any history, given this configuration. This is important, since it will be possible to determine the largest n given a configuration, and, as we prove later, this will attain the optimal expected utility for the agents.

Lemma 5.3. *$n \in N$ exists for each $\omega \in \Omega$ such that $\omega \notin \Omega_n$, but for each $n' < n$, $\omega \in \Omega_{n'}$.*

In the above lemma we show that a smaller n enables more configurations to be stable. However, we will now show that as we observed when comparing the models of $n = 1$ and $n = 2$, a larger n increases the expected average utility of the agents. Thus, given a configuration, it is beneficial to find the largest n for which answering is in equilibrium for this configuration. We start with an auxiliary lemma, and then we proceed with our conclusions.

Lemma 5.4. *Given $n \in N$, for each ω, such that $\omega \in \Omega_n$, for each history, the expected utility of agent A from receiving an answer from agent B, is higher than the expected utility if the query receives no answer.*

In the following lemma we prove our main claim about the change in the agents' utilities: as n, the number of periods considered to decide about a punishment increases, the agent's average utility increases too. This was demonstrated in Section 5.1.2 for the case of changing n from one-period to two periods. In the following theorem, we prove that the same direction of influence exists for any positive n.

Theorem 5.2. *For each agent i, for each $n \in N$, for each ω, such that $\omega \in \Omega_n \cap \Omega_{n+1}$, and for each history h, $V_i^{n+1,h} > V_i^{n,h}$.*

We demonstrate our main conclusions in Fig. 13, for a particular configuration of parameters ($c_i = 1$, $o_i = 0.1$, $v_i = 100$, $p_i = 0.5$, $\mathcal{D}_i = 0.9$).[3] The figure demonstrates that as n increases $V_A^{n,h}$ also increases, as proven in Theorem 5.2, but the increase is not linear: the increment level decreases as n increases. However, as proven in Lemma 5.1, as

[3] The value v_i is much larger than its value in the one-period model since we consider models with a more restricted set of configurations for which the equilibrium exists.

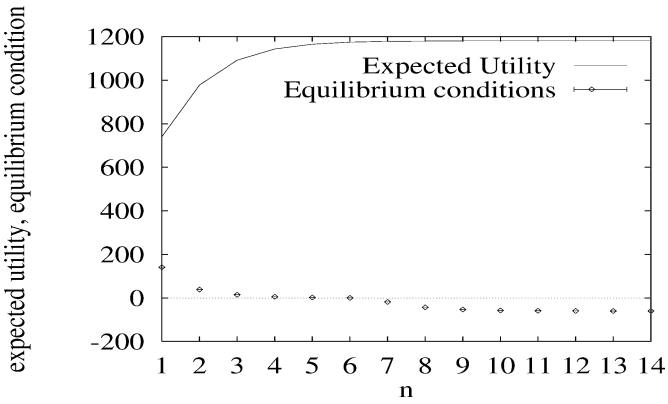

Fig. 13. n-period model: expected utility and equilibrium existence index as a function of n.

n grows, the set of appropriate configuration values becomes smaller. 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 a history of $((1,\ldots,1),(0,\ldots,0))$, and its utility if it ignores the query. If the difference is positive, then an n-period equilibrium exists, as was proven in Lemma 5.1. It is also clear that as the difference increases, the n-period 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 the agents increases, while the trigger equilibrium exists for a smaller set of configurations.

The conclusion from this section is that given a configuration of parameters, ω, the agents can decide about the optimal n to be used. Thus, 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 a history of $((1,\ldots,1),(0,\ldots,0))$, and it is still beneficial for agent B to answer queries given a history of $((0,\ldots,0),(1,\ldots,1))$. 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 5.2.

5.3. Punishment after k unanswered queries from n

In the previous sections, we considered strategies in which punishment is done after one, two or any other predefined number of subsequent unanswered queries. We found that as the number of unanswered queries required for punishment increases, the expected utility of the agents increases, but the number of configurations for which equilibrium exists decreases. In this section, we consider a new type of strategy profiles. As in the previous section, n periods of history are considered by the agent when it has to decide when to punish its opponent. The difference in this strategy is in the fact that it is ample to observe $k \leqslant n$ queries with no answer in order to decide about a punishment. The model considered in Section 5.2 is a special case of this model, with the restriction of $k = n$.

In fact, our model includes more possible combinations of strategies that may yield a higher expected utility. In particular, there are configurations, and values of n, for which the equilibrium of the n-period model does not exist, but there are strategy profiles in equilibrium, in which punishment is done after k unanswered queries from n, for the same n. Such a strategy profile may often be more beneficial than choosing a smaller n and using the n-period observation strategies. In this section, we will consider the k from n-period observation model. First, we present important notations and we discuss some of the properties of the k/n-model. Finally we suggest how to choose the best value of k and n given a particular configuration.

We denote $\Omega_{k,n}$ to be the set of configurations for which the strategy profile of punishment after k unanswered queries from n, is in equilibrium. We denote by $h = (h_A, h_B)$ the history of the n last results of queries sent to agent A and the n last queries sent to agent B. We denote $V_i^{k,n,h}$ the utility of agent i when it is its turn to answer, given history h, and assuming a model of punishment after k unanswered queries from n queries. Similarly, $U_i^{k,n,h}$ is the utility of agent i when it is its turn to ask a query, given history h, for a model of punishment after k unanswered queries from n queries.

Similar to Section 5.2, the expected utility $V_A^{k,n,h}$ of agent A when required to answer a query, is defined as follows:

$$V_A^{k,n,h} = \begin{cases} (\mathcal{D}_A U_A^{k,n,(h_A,h_B)}) & \textit{sum\_zeroes}(h_B) \geqslant k, \\ -o_A + p_A \cdot (-c_A + \mathcal{D}_A U_A^{k,n,(h_A \ll 1,h_B)}) & \\ \quad + (1-p_A)(\mathcal{D} V_A^{k,n,(h_A \ll 0,h_B)}) & \textit{sum\_zeroes}(h_A \ll 0) \\ & \quad \geqslant k, \\ -o_A + p_A \cdot (-c_A + \mathcal{D}_A U_A^{k,n,(h_A \ll 1,h_B)}) & \\ \quad + (1-p_A)(\mathcal{D}_A U_A^{k,n,(h_A \ll 0,h_B)}) & \text{otherwise} \end{cases} \tag{22}$$

and the expected utility $U_A^{k,n,h}$ of agent A when it asks a query, is defined as follows:

$$U_A^{k,n,h} = \begin{cases} (\mathcal{D}_B U_B^{k,n,(h_A,h_B)}) & \textit{sum\_zeroes}(h_A) \geqslant k, \\ p_B \cdot (v_A + \mathcal{D}_B V_A^{k,n,(h_A,h_B \ll 1)}) & \\ \quad + (1-p_B)(\mathcal{D} U_A^{k,n,(h_A,h_B \ll 0)}) & \textit{sum\_zeroes}(h_B \ll 0) + 1 \geqslant k, \\ p_B \cdot (v_A + \mathcal{D}_B V_A^{k,n,(h_A,h_B \ll 1)}) & \\ \quad + (1-p_B)(\mathcal{D}_B V_A^{k,n,(h_A,h_B \ll 0)}) & \text{otherwise.} \end{cases} \tag{23}$$

The intuition behind these formulas is the same as in the model of punishing after n consequent failures. If it is agent A's turn to answer a query, then it may just ignore it, if there are at least k failures from n last queries to agent B. If it has to attempt to answer the query, then with a probability of p_A it will succeed and with a probability of $(1-p_A)$ it will fail. In a case of a failure, a punishment mode can be reached, if there are at least k unanswered queries from n last queries to A, including the current failure. (If agent A punishes agent B and ignores its query, this punishment will not be saved in the history.)

Similarly, if it is agent A's turn to receive an answer to a query, and agent A has at least k unanswered queries, i.e., it is agent A's punishment phase, then its query will be

ignored. Otherwise, agent B will attempt to answer the query. It will receive an answer with a probability of p_B and with a probability of $(1 - p_B)$ agent B will not find an answer. In this case, if there are at least k failures of agent B, then agent A will punish agent B.

The difference between the two models is the condition for a punishment event to take place. Punishment in the n-period model is implemented after n unanswered queries, i.e., if all the $n - 1$ events in the history of an agent are zeroes. Punishment in the k/n-period model is implemented when there are k unanswered queries from the history of length n, i.e., if the sum of unanswered queries is greater than or equal to k.

For the expected utility calculation the agent has to use an algorithm based on the formulas $V_A^{k,n,h}$ and $U_A^{k,n,h}$. As in the n-period model, the formulas depend on each other and can be solved using a divide and conquer algorithm or a dynamic programming algorithm. However, in this model, all the possibilities of histories with a length of n have to be considered since the utility of one history depends on the utility of other histories (with an additional success, or with an additional failure). A failure can cause a future punishment even if there was a later success after this failure. (Only the number of consequent last failures has to be considered in the n-period model, since a failure with a later success has no meaning.) Thus, all the possible combinations of histories of length n have to be considered in our model, i.e., there are 2^{2n} possible combinations of histories to be evaluated.[4] Given that this number is exponential in n, the best algorithm for evaluating $V_A^{k,n,h}$ or for checking the existence of the equilibrium should take at least exponential time.

5.3.1. Properties of the model

Given the ability to use a trigger strategy profile in which punishment is imposed after k from n unanswered queries, we would like to suggest how the values of k and n should be chosen by the agents. This is an important decision that influences the utilities of the agents, as well as their motivation to use the equilibrium strategies. In this section, we will check the influence of k and n in order to suggest how the agents should choose them. We start by formally proving certain properties, and we proceed by testing other important properties via simulations.

5.3.2. The influence of k

In order to prove the influence of k on the equilibrium existence, we start with two auxiliary lemmas. First, we prove that an equilibrium exists whenever it is worthwhile for each agent i to answer the query of its opponent even with the best history for i. Denote by $best\_case_i(k, n)$ the best history for agent i. This history includes no failures of i, but the maximum number of allowed failures of j. In other words, in $best\_case_A(k, n)$ there are $n - 1$ last successes of agent A, and $k - 1$ recent failures (from n) at the end of the history of agent B. If it is worthwhile for agent i to answer the query of agent j even in this history, it will be worthwhile for i to answer the queries for any other history.

[4] During agent A's punishment mode queries to agent or from agent A are ignored until it succeeds in answering a query. Thus, a history of n length for agent A and n length for agent B is sufficient to represent all the required information for future decisions.

Lemma 5.5. *Consider an equilibrium of a strategy profile where each agent answers the queries of its opponent, unless there was no response to at least k queries from the last n queries sent to this opponent. An equilibrium will exist, if it is worthwhile for each agent i to attempt to answer agent j after the best_case$_i(k,n)$ history of $n-1$ consecutive answered queries by agent i, and $k-1$ consecutive unanswered queries of agent j.*

According to the above lemma, the equilibrium of the k from n-period observation strategies is stable if, and only if, these strategies are stable after the history of $n-1$ consecutive answered queries by agent i, and $k-1$ consecutive unanswered queries of agent j, for $i=A$, $j=B$ and for $i=B$, $j=A$. In order to check whether an equilibrium exists or not, based on this reasoning we only have to check the condition for the history of *best_case*$_A(k,n)$ and *best_case*$_B(k,n)$.

Lemma 5.6. *Given the best_case$_i(k,n)$ history of exactly $k-1$ consecutive last failures of agent j and $n-1$ consecutive successes of agent i,*

$$\begin{aligned} &\textit{ignr\_loss}_i\big(\omega,k,n,\textit{best\_case}_i(k,n)\big)\\ &\quad>\textit{ignr\_loss}_i\big(\omega,k+1,n,\textit{best\_case}_i(k+1,n)\big). \end{aligned}$$

Based on the above lemmas, the next theorem summarizes our results concerning the influence of k on the conditions of the equilibrium. We prove that as k increases an equilibrium holds for a smaller set of configurations.

Theorem 5.3. *If $k1<k2\leqslant n$, then $\Omega_{k2,n}\subseteq\Omega_{k1,n}$.*

According to the above theorems, the equilibrium tends to be weaker as k increases when keeping n fixed. This result is intuitively clear, since as k increases punishment becomes rarer. This causes the threat of punishment to decrease, and it may also harm the stability of the strategy profiles. Thus, we can assume that as k increases, the expected utility of the agents will increase, but the equilibrium will hold more rarely.

5.3.3. Simulation results

In order to check the influence of different values of k and n on particular configurations and histories, we developed an algorithm based on dynamic-programming, as explained above. This algorithm is able to run for different strategy profiles, based on different values of k and n.

After implementing this algorithm, we ran a simulation which randomly generated configurations ω and histories h, and checked how different values of k and n influence the utility, and the equilibrium existence, given ω and h. The values of the configurations were generated as follows: q_A, q_B, p_A and p_B were drawn randomly from 0 to 1. δ was drawn randomly from 0.9 to 1. The value of δ is near 1, since the loss from a delay of one-period should be small. v_A was drawn from 0 to 100, c_A was drawn from 0 to 0.1, and o_A was drawn from 0 to 0.01. We chose these cost parameter values since we wanted the equilibrium to exist for different values of k and n. This type of relation between the cost of acquiring information and the benefits from it often appear in knowledge sharing, where

Table 1
Influence of n and k on: (a) the average utility; (b) the ratio of configurations for which an equilibrium exists. The format is (a)/(b)%

n	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
1	74.6/99.59%					
2	59.7/99.72%	98.2/71.3%				
3	53.5/99.74%	84.8/81.2%	110.1/50.0%			
4	50.5/99.74%	78.3/82.8%	99.2/57.7%	117.0/33.4%		
5	48.9/99.74%	74.9/83.2%	93.2/58.5%	108.3/42.5%	121.4/22.3%	
6	47.9/99.74%	72.9/83.7%	89.8/58.1%	103.2/43.1%	114.5/30.7%	124.4/14.7%

the agent which asks a query can greatly benefit from an answer, but the agent which has to answer it incurs a cost. Though the cost may be low it may still be significant.

For simplification, we checked only the expected utility of agent A, and the equilibrium existence from the point of view of agent A. The influence on agent B should be, on average, identical, since it has the same utility function and equilibrium existence index, and the configuration values are randomly generated.

We first ran a simulation with 10,000 sets of parameters in order to check the influence of k and n on the average utility and on the equilibrium existence index. Table 1 presents the average values of the expected utility of agent A, and the ratio of the configuration in equilibrium from all the configurations checked, given different configuration values.

In Table 1, we can see that as k increases, and as n decreases, the average expected utility increases, and the ratio of the configurations in equilibrium decreases. The intuition behind this result is that as the number of unanswered queries required for punishing, k, increases, then punishment is inflicted more rarely. Thus, in general, the expected utility of the agents increases while their tendency to follow the equilibrium decreases. As the number of periods tested in order to punish, n, increases, punishment will be performed more often, and for a longer time. Consequently, the expected utility of the agents decreases but the threat of a punishment is stronger showing that the equilibrium exists more frequently.

The above table presents the average direction of the influence of k and n. However, we also wanted to test whether this direction of influence exists for *all* configurations, and not only on the average. In order to do so, we ran a simulation for each of the above changes in k or in n, and tried to find a counter example with the opposite direction of the average change. First, we checked how the increase of k influences each configuration tested. As we explained above, as more unanswered queries are required in order to punish, agent i's punishment is performed more rarely, increasing the expected utility of agent i. However, as k increases, the punishment of the opponent j is also performed more rarely, meaning that agent i has to attempt to answer more queries.

We first ran an additional set of simulations for varied histories. In most of our simulations, results show that as k increases the expected utility of an agent increases. But, there were also examples, given varied histories, where increasing k, causes the expected utility $V_A^{k,n,h}$ to decrease. We observed that all the cases where the expected utility decreases when k increases were always in situations where the history includes at least one failure of one of the agents.

However, the process of determining the values of k and n is performed before the repeated interaction starts. Thus, the agents decide to prefer $(k1, n1)$ over strategy $(k2, n2)$, if strategy $(k1, n1)$ has a higher expected utility, given the history before the repeated interaction starts. Since the interaction starts with no unanswered queries, the agents should consider only a history of $((1, \ldots, 1), (1, \ldots, 1))$ when they decide about their strategy, given that this decision is done once and before any query was even sent: the choice of n and k cannot to be taken during the interaction itself. We ran our simulation for 10,000 configuration values, given the history of $((1, \ldots, 1), (1, \ldots, 1))$. In all the configurations we tested, the expected utility of the agents increases as k increases. (We did not prove this formally, because of the complexity of the model.) This result is interesting, since it shows that the direction of the influence on the utility also depends on the history of a given sequence of interactions and not only on the configuration. The intuition behind this effect is that there are other histories, for example, $best\_case_A(k, n)$, where the punishment of agent B will almost certainly be performed. In such situations, agent A may sometimes be motivated to use a more threatening mechanism, since this will almost surely affect agent B and increase the utility of agent A. However, as we explained above, the history given when the agents decide about their strategies is $((1, \ldots, 1), (1, \ldots, 1))$. So only this history has to be considered.

Recall from Theorem 5.3 that as k increases, the equilibrium holds for a more restricted set of configurations. We can summarize these two conclusions and say that given a value of n, as k increases the utility of the agents increases, but the equilibrium holds more rarely. Thus, we can conclude that given the value of n, the agents will be motivated to choose the highest possible k for which the (k, n) pair is in equilibrium.

However, there is no uniform rule of how to choose n. In our simulations we found that on the average, as n increases the utility decreases while the equilibrium exists for more cases. However, there were also counter examples. Even when we check the influence on the utility function given a history of $((1, \ldots, 1), (1, \ldots, 1))$, there still are cases in which the increase of n causes $V_A^{k,n,h}$ to increase.

We ran our simulation for $k = 2$ and n increases from 5 to 6, and we ran 10,000 configurations. For 83.7% of the configurations that are in equilibrium, we found that as n increases the utility decreases. For 3.3% of the configurations that are not in equilibrium for $n = 5$ or for $n = 6$, equilibrium starts to exist as n increases to 6, and for 1.1% of these configurations, the equilibrium starts to exist as n decreases to 5. So we can conclude that given a configuration ω, the direction of the influence of n on the utility and on the existence of equilibrium, depends on ω, although the average direction is clear. Thus, if the agents would like to be certain that they are taking the best value of n for a given k, they should check all the possible values of n, given a configuration ω.

Table 2 summarizes our conclusions considering the change of n in the n-model, and the change of k or n in the k/n-model. The results are based on a simulation of over 10,000 configurations of parameters. There are cells with proven directions, and according to our simulations there are cells with and without uniform directions.

A typical demonstration is presented in Fig. 14, for a particular configuration $\omega = (c_i = 1, o_i = 0.1, v_i = 20, p_i = 0.5, \mathcal{D}_i = 0.9)$. In this example, we can see that as k increases, and as n decreases, the expected utility of agent A increases, while the equilibrium exists more rarely. In this example, the pair $k = n = 3$ maximizes the expected utility

Table 2
The influence of changing n or k, on the expected utility and on the existence of an equilibrium, as found theoretically or by simulations. The influence direction of cells with the indication (simulation), was obtained by simulation. The influence direction of cells with the indication (proven), was proven

	Utility any history	Utility history $((1,\ldots,1),(1,\ldots,1))$	Number of configurations in equilibrium
n-model, n increases	↑ (proven)	↑ (proven)	↓ (proven)
k/n-model k increases	on average ↑ (simulation)	always ↑ (simulation)	↓ (proven)
k/n-model n increases	on average ↓ (simulation)	on average ↓ (simulation)	on average, ↑ (simulation)

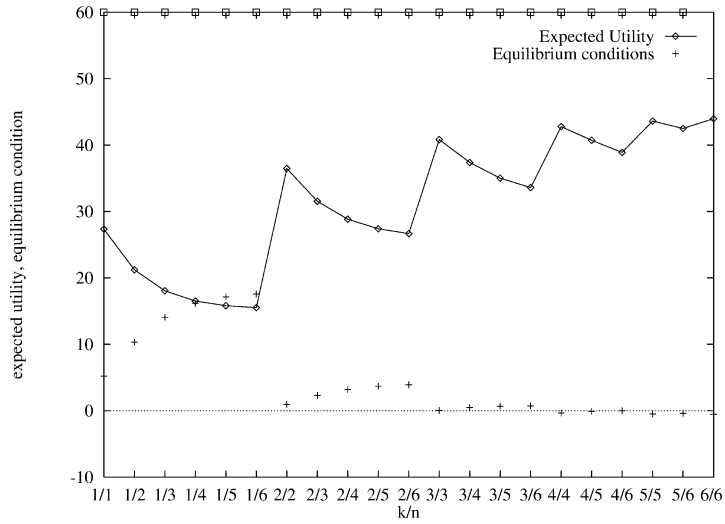

Fig. 14. Punishment after k failures from n: expected utility and trigger equilibrium conditions as a function of k/n.

of both agents, while the trigger equilibrium still exists. Thus, the agents should choose the equilibrium based on this pair. However, for other examples, the optimal pair is different, and the optimal value of k is often different from n.

We can see that different values of k and n may yield different values of the expected utility and their particular value determines the existence of the trigger equilibrium. Recall that given a configuration of parameters, the agents have to find the pair of k and n for which the trigger equilibrium exists, and to choose the optimal pair (k,n) from among them. As we explain above, in order to evaluate the expected utility and the equilibrium conditions, the agents should examine all the possible histories, since the utility given a particular history depends on the utilities for other histories (with an additional success, or with an additional failure). There are 2^{2n} possible values of histories. Thus, the evaluation of one pair (k,n) requires exponential time, in view of the fact that all the possible histories

can occur. Accordingly, only feasible values of n should be examined, and the agents should determine a value max_n, which is the largest value of n that is feasible and check all the values of $n = 1, 2, \ldots, max_n$.

Given a particular value of n, according to our conclusions, as k increases, the utility increases while the conditions hold less frequently. Thus, we can conclude that for each feasible value of n, the agents will run a binary search in order to find $best\_k(n)$, which is the largest value of k for which the equilibrium holds, for a given n. Then, they will compare the values of the different pairs $(n, best\_k(n))$, and choose the optimal value among them.

There may be situations in which each agent will prefer a different pair (k, n) due to different parameters' values of the different agents. For example, suppose that there are two possible pairs in equilibrium: $k = 2, n = 4$ and $k = 3, n = 5$. Suppose also that one agent prefers the pair $k = 2, n = 4$, while the other agent prefers the pair $k = 3, n = 5$. In such cases, the agents can determine a rule of how to choose (k, n), such as, maximizing their average expected utility or maximizing the product of the expected utility.

6. 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. Mainly, we considered an alternating model, where at each time period each agent may have a query. First, we introduced the *one-period strategy profile*, 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 from n queries, and we checked the influence of changing k and n.

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.

Appendix A. Table of symbols

Table A.1
Table of symbols

Symbol	Explanation	Appears
$\mathcal{A}, \mathcal{B}$	components of U_i in the alternating model	Lemma 4.4
v_i	the utility of agent i for answering its query	4
c_i	the cost of agent i in the case of success	4
δ	the discount factor of the utility function of the agents	4.2

Table A.1 (*continued*)

Symbol	Explanation	Appears
$\mathcal{D}$	the expected discount ratio from the current time, when it is agent i's turn to answer until the next time when it is its turn to answer	4.2
$\mathcal{D}_i$	the expected discount ratio from the current time until t_i	4.2
$history$	the histories of both agents. $(history_A, history_B)$	5
$history_i$	the results of the $n-1$ last events of queries sent to agent i (each result is 1 for an answered query, 0 for a query with no answer) (n-period model)	5
n	the number of periods observed in order to decide about a punishment	5
k	the number of failures from n observed periods, for which punishment is done	5.3
o_i	the cost of agent i when it tries to answer a query	4
$ignr\_loss_i(\omega, n, history)$	the expected loss of agent i due to failure or ignoring a query	5
$\{Normal, Punish_A, Punish_B\}$	the three phases included in the strategy profile	Lemma 4.2
p_i	the probability of agent i to succeed in answering, if it tries to.	4
q_i	the probability for agent i to have a query in a given time period (in the stochastic mode)	4.2
t_0	the current time	4.2
t_i	the time when agent i will send a query	4.2
U_i	the expected utility for agent i when it is its turn to receive an answer from agent $j \neq i$ (whether agent i succeeds or not) (alternating model)	4.3
U_i^{xy}	the expected utility of agent i in the two-period model, when the last event of agent A was x, the last event of agent B was y, and it is i's turn to receive an answer from agent $j \neq i$ (two-period model)	5.1
$U_i^{history,n}$	the expected utility for agent i when it is its turn to receive an answer from agent $j \neq i$ given $history$, in the n-period model	5.2
V_i	the expected utility for agent i if it decides to answer the query of $j \neq i$ (whether it succeeds or not) (alternating)	4.3
V_i^{xy}	the expected utility of agent i in the two-period model, when the last event of agent A was x, the last event of agent B was y, and it is now agent i's turn to answer a query	5.1
$V_i^{n,history}$	the expected utility for agent i when it is its turn to answer the query of $j \neq i$, in the n-period model	5.2
F_i	the utility of agent i from deviating to the one-stage equilibrium, and ignoring the query of agent $j \neq i$ (alternating)	4.3
ω	a combination of the parameters of the model	5
Ω	a set of all the combinations of model parameters	5
Ω_n	a set of all the combinations of the model parameters which are stable for a strategy profile based on punishment after n consequent failures	5

Appendix B. Proofs

Proof of Lemma 4.1. Based on equation (1), $\mathcal{D}_A$ is defined as $\delta q_A/(1-\delta(1-q_A))$. In order to prove that $\mathcal{D}_A \geqslant 0$, we have to prove that the numerator is greater than or equal to 0, and that the denominator is positive. Since $\delta \geqslant 0$ and $q_A \geqslant 0$, it is clear that the numerator is greater than or equal to 0. The denominator is positive whenever $1-\delta(1-q_A) > 0$. This holds whenever $\delta(1-q_A) < 1$, and this is true, since $\delta < 1$ and $0 \leqslant q_A \leqslant 1$.

In order to prove that $\mathcal{D}_A < 1$, we have to prove that $\delta q_A < 1-\delta(1-q_A)$, i.e., $\delta < 1$, and this is true by definition. Symmetrically, also $\mathcal{D}_B$ is between 0 and 1, and the proof is based on Eq. (1).

As proven, $0 \leqslant \mathcal{D}_A < 1$. Symmetrically, $0 \leqslant \mathcal{D}_B < 1$. According to Eq. (2), $\mathcal{D} = \mathcal{D}_A\mathcal{D}_B$. Thus, $0 \leqslant \mathcal{D} < 1$. □

Proof of Lemma 4.2. Consider the strategy profile defined in Section 4.1. If $V_A \geqslant F_A$, and agent B uses the above strategy, then any attempt of agent A to deviate from any history, will reduce its utility.

Suppose agent B sends a query to agent A; There are three possible phases of the game:

Punish$_B$: In this phase, if agent A punishes agent B, then it saves the cost of c_A. Whether it answers or not, the next state will be *Punish$_B$*. Thus, it is clear that A prefers to punish B in state *Punish$_B$*.

Normal and Punish$_A$: In both phases, if agent A will attempt to answer agent B's query, then its utility will be V_A, and if it decides to deviate and ignore the query, its utility will be F_A. Thus, the condition for answering is $V_A \geqslant F_A$.

The same arguments also apply to the case when agent A sends a query to agent B. So the conditions needed for the equilibrium to hold are $V_A \geqslant F_A$, and $V_B \geqslant F_B$. □

Proof of Lemma 4.3. By manipulating the formulas of Attribute 4.1, and solving the recursive formulas, we obtain

$$U_A = \frac{p_B * v_A}{1+\mathcal{D}(p_B-1)} + \frac{\mathcal{D}_B * p_B V_A}{1+\mathcal{D}(p_B-1)}.$$

We use the notation

$$\mathcal{A} = \frac{p_B * v_A}{1+\mathcal{D}(p_B-1)}, \tag{B.1}$$

and

$$\mathcal{B} = \frac{\mathcal{D}_B * p_B}{1+\mathcal{D}(p_B-1)} \tag{B.2}$$

so $U_A = \mathcal{A} + \mathcal{B}V_A$.

Recall from Attribute 4.1, that $V_A = -o_A + p_A(-c_A + \mathcal{D}_A U_A) + (1-p_A)F_A$.

By substituting F_A, we obtain,

$$V_A = -o_A + p_A(-c_A + \mathcal{D}_A U_A) + (1-p_A)\mathcal{D}V_A.$$

By substituting U_A, we obtain,

$$V_A = -o_A + p_A(-c_A + \mathcal{D}_A(\mathcal{A} + \mathcal{B}V_A) + (1 - p_A)\mathcal{D}V_A.$$

By eliminating V_A, we obtain,

$$V_A(1 - p_A\mathcal{D}_A\mathcal{B} - (1 - p_A)\mathcal{D}) = -o_A + p_A(-c_A + \mathcal{D}_A\mathcal{A})$$

and the value of V_A is

$$V_A = \frac{-o_A - p_A c_A + p_A\mathcal{D}_A\mathcal{A}}{1 - p_A\mathcal{D}_A\mathcal{B} - (1 - p_A)\mathcal{D}}$$

and when opening the formula, we finally obtain

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

Proof of Proposition 4.1. Substituting $\mathcal{B}$ in Eq. (8), we obtain

$$1 - p_A\mathcal{D}_A\frac{\mathcal{D}_B * p_B}{1 + \mathcal{D}(p_B - 1)} - \mathcal{D}(1 - p_A) > 0.$$

Since it is clear that $1 + \mathcal{D}(p_B - 1) > 0$, we only have to prove that

$$1 + \mathcal{D}(p_B - 1) - p_A\mathcal{D}_A\mathcal{D}_B * p_B + \mathcal{D}(p_A - 1)\big(1 + \mathcal{D}(p_B - 1)\big) > 0$$

and since $\mathcal{D} = \mathcal{D}_A + \mathcal{D}_B$,

$$1 + \mathcal{D}(p_B - 1 - p_A p_B - 1 + p_A) + \mathcal{D}^2(p_A p_B - p_A - p_B + 1)$$

and

$$1 + \mathcal{D}(-2 + p_A + p_B - p_A p_B) + \mathcal{D}^2(1 - p_A - p_B + p_A p_B)$$

and this is equal to

$$1 + \big(\mathcal{D}^2 - 2\mathcal{D}\big) + p_A p_B\big(\mathcal{D}^2 - \mathcal{D}\big) + (p_A + p_B)\big(\mathcal{D} - \mathcal{D}^2\big).$$

$(\mathcal{D}^2 - 2\mathcal{D})$ is minimized when $\mathcal{D}$ approaches one, thus $(\mathcal{D}^2 - 2\mathcal{D}) \to -1$, so we are left with

$$p_A p_B\big(\mathcal{D}^2 - \mathcal{D}\big) + (p_A + p_B)\big(\mathcal{D} - \mathcal{D}^2\big)$$

which is equal to $(\mathcal{D} - \mathcal{D}^2)(p_A + p_B - p_A p_B)$ which is positive. $\square$

Proof of Lemma 4.4. According to Lemma 4.3, the expected utility of an agent when attempting to answer a query, is

$$V_A = \frac{-o_A - p_A c_A + p_A\mathcal{D}_A\mathcal{A}}{1 - p_A\mathcal{D}_A\mathcal{B} - \mathcal{D}(1 - p_A)}.$$

According to Lemma 4.2, there is an equilibrium when $V_A \geqslant F_A$ and $V_B \geqslant F_B$. According to Attribute 4.1, $F_A = \mathcal{D}V_A$. As we proved in Lemma 4.1, $\mathcal{D} \in [0, \ldots, 1)$. Using this is a necessary and sufficient condition to prove that $F_A < V_A$ is $V_A \geqslant 0$.

According to Lemma 4.2, there is an equilibrium whenever $V_A \geqslant F_A$, and $V_B \geqslant F_B$. However, according to Attribute 4.1, F_A is defined to be $\mathcal{D}V_A$, and $\mathcal{D}$ was proved in

Lemma 4.1 to be between 0 and 1. Thus, $V_A \geqslant F_A$ whenever $V_A \geqslant 0$, since in any case when $V_A \geqslant 0$, $F_A = \mathcal{D}V_A < V_A$

According to Proposition 4.1, the denominator of V_A is positive. In the following formula, we will find under which conditions the numerator will also be positive. In other words, we have to find whenever $p_A\mathcal{D}_A\mathcal{A} \geqslant p_A c_A + o_A$. $p_A \in (0,1)$, so $p_A > 0$. Thus, we can maintain that $\mathcal{D}_A\mathcal{A} > c_A + o_A/p_A$. Substituting $\mathcal{A}$ with its formula, we receive,

$$\mathcal{D}_A \frac{p_B v_A}{1+\mathcal{D}(p_B-1)} \geqslant c_A + \frac{o_A}{p_A}$$

and,

$$v_A \geqslant c_A \frac{(1+\mathcal{D}(p_B-1))}{\mathcal{D}_A p_B} + o_A \frac{(1+\mathcal{D}(p_B-1))}{p_A p_B \mathcal{D}_A}$$

and,

$$v_A \geqslant c_A\left(\frac{1}{\mathcal{D}_A p_B} + \frac{\mathcal{D}_B(p_B-1)}{p_B}\right) + E_A\left(\frac{1}{p_A p_B \mathcal{D}_A} + \frac{\mathcal{D}_B(p_B-1)}{p_A p_B}\right).$$

Substituting $\mathcal{D}_A$ with $\delta q_A/(1-\delta+\delta q_A)$ and $\mathcal{D}_B$ with $\delta q_B/(1-\delta+\delta q_B)$, and $\mathcal{D}$ with $\mathcal{D}_A\mathcal{D}_B$, we obtain

$$\begin{aligned} v_A \geqslant{}& c_A\left(\frac{1-\delta+\delta q_A}{\delta q_A p_B} + \frac{\delta q_B(p_B-1)}{p_B(1-\delta+\delta q_B)}\right) \\ &+ o_A\left(\frac{1-\delta+\delta q_A}{p_A p_B \delta q_A} + \frac{\delta q_B(p_B-1)}{(1-\delta+\delta q_B)p_A p_B}\right). \end{aligned}$$

In other words,

$$v_A \geqslant \left(c_A + \frac{o_A}{p_A}\right)\left(\frac{1-\delta+\delta q_A}{\delta q_A p_B} + \frac{\delta q_B(p_B-1)}{p_B(1-\delta+\delta q_B)}\right). \tag{B.3}$$

Symmetrical arguments will lead to the symmetrical formula for the case where agent B has to answer. □

Proof of Lemma 4.5. First, in Eq. (10) of Lemma 4.4, the equilibrium condition is presented as a ratio between $v_A p_A$ and $c_A p_A + o_A$, and this ratio should be larger than another term, that is a function of the other parameters. So it is clear that as v_A increases, the condition is inclined to hold, and as c_A or o_A increases, the condition may be violated.

The influence of c_A, o_A and v_A on the utility function can be shown using the formula of V_A. According to Lemma 4.3,

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

and we already proved in Lemma 4.1 that the denominator is positive. Thus, it is clear that the influence direction of o_A and c_A is linearly negative, with less influence of o_A, while the influence direction of v_A is linearly positive (since it is easy to show that its coefficient is positive).

We proceed by proving that as p_A or p_B increases, the expected utility of each agent increases, and the set of parameter values for which equilibrium exists, increases too.

We start with proving that p_A and p_B positively influence V_A. Manipulating Eq. (6) of Lemma 4.3,

$$V_A = \frac{(1+\mathcal{D}p_B-\mathcal{D})(-o_A-p_Ac_A)+p_A\mathcal{D}_Ap_Bv_A}{(1+\mathcal{D}p_B-\mathcal{D})(1-\mathcal{D}+\mathcal{D}p_A)-p_A\mathcal{D}p_B}.$$

When isolating p_A, we obtain

$$V_A = \frac{-o_A(1+\mathcal{D}p_Bc_A-\mathcal{D})+p_A(-c_A-\mathcal{D}p_B-\mathcal{D}c_A+\mathcal{D}_Ap_Bv_A)}{(1-\mathcal{D})(1+\mathcal{D}p_B-\mathcal{D})+p_A(\mathcal{D}+\mathcal{D}^2p_B-\mathcal{D}^2-\mathcal{D}p_B)}.$$

Since $\mathcal{D}<1$, $-o_A(1+\mathcal{D}p_Bc_A-\mathcal{D})<0$, if the equilibrium holds, then the numerator is positive. So $(-c_A-\mathcal{D}p_B-\mathcal{D}c_A+\mathcal{D}_Ap_Bv_A)>0$. The denominator includes two parts, but since $\mathcal{D}<1$, $(1-\mathcal{D})(1+\mathcal{D}p_B-\mathcal{D})>0$. Thus, as p_A is multiplied, the numerator increases more than the denominator, so the value of V_A increases.

A similar proof is true for the influence of p_B on V_A. The formula of V_A can be written as

$$\frac{p_B(-o_A\mathcal{D}-p_A\mathcal{D}c_A+p_A\mathcal{D}_Av_A)+(-o_A+o_A\mathcal{D}-p_Ac_A+p_A\mathcal{D}c_A)}{p_B(\mathcal{D}-\mathcal{D}^2+p_A\mathcal{D}^2-\mathcal{D}p_A)+(1-2\mathcal{D}+\mathcal{D}^2+p_A\mathcal{D}-p_A\mathcal{D}c_A)}.$$

The second part of the numerator can also be written $(\mathcal{D}-1)(o_A+p_Ac_A)$, and this value is negative since $\mathcal{D}<1$. Thus, the first part of the numerator must be positive when there is an equilibrium. The second part of the denominator can be written as $(\mathcal{D}-1)(\mathcal{D}(1-p_A)-1)$. Both parts of the expression are negative, so the expression on the whole is positive. Again, as p_B is multiplied, there is a greater effect on the numerator (where one part depends on p_B and the other part is negative) than on the denominator (where the part which does not depend on p_B is positive). Thus, as p_B increases, V_A increases too.

Agent A's willingness to attempt to answer queries is positively influenced by increasing p_A or p_B. This influence can be easily shown from the condition $V_A(1-\mathcal{D})\geqslant 0$. Since p_A and p_B increase v_A, they also increase the equilibrium condition, given the other parameters stay constant. □

Proof of Lemma 4.6. First, we prove that as δ increases, $\mathcal{D}_i$ also increases. According to Eq. (1),

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

As δ increases, the numerator increases, while the denominator decreases. Thus, as δ increases, $\mathcal{D}_i$ increases.

We proceed by proving that as q_A or q_B increases, $\mathcal{D}_i$ increases too. The value of $\mathcal{D}_i$ can be written as

$$\mathcal{D}_i = \frac{\delta q_i}{1-\delta+\delta q_i}$$

so, as q_i increases with a ratio r, the numerator increases in the same ratio r, while part of the denominator increases in ratio r. However, the rest of the denominator $(1-\delta)$ remains unchanged. Since this part is positive ($\delta<1$), the denominator at whole will increase with a ratio less than r. Thus, the numerator increases with a ratio larger than the increased ratio of the denominator, so the value of $\mathcal{D}_i$ increases.

According to Eq. (2),

$$\mathcal{D} = \mathcal{D}_A \mathcal{D}_B.$$

We showed that as δ increases, $\mathcal{D}_A$ and $\mathcal{D}_B$ increase, $\mathcal{D}$ also increases. Similarly, as q_A increases, $\mathcal{D}_A$ increases while $\mathcal{D}_B$ remains unchanged, and as q_B increases, $\mathcal{D}_B$ increases while $\mathcal{D}_A$ remains unchanged. Thus, as q_A or q_B increases, $\mathcal{D}$ increases.

We will now show that as $\mathcal{D}_A$ or $\mathcal{D}_B$ increases, the expected utility V_A also increases. Following Lemma 4.3, the value of V_A can be written as

$$V_A = \frac{(1 + (p_B - 1)\mathcal{D})neg + p_A \mathcal{D}_A pos}{(1 + (p_B - 1)\mathcal{D})(1 + (p_A - 1)\mathcal{D} - p_A p_B \mathcal{D}},$$

where $neg = (-o_A - p_A c_A)$ is negative, and $pos = p_B v_A$ is positive. As $\mathcal{D}$ increases, the value of $(1 + (p_B - 1)\mathcal{D})$ decreases, and becomes closer to 0. Since this value is multiplied with *neg*, the numerator increases as $\mathcal{D}$ increases. Moreover, as $\mathcal{D}_A$ increases, an additional positive influence exists since $p_A \mathcal{D}_A pos$ also increases, so both parts of the numerator increase as $\mathcal{D}_A$ increases. If $\mathcal{D}_B$ increases while $\mathcal{D}_A$ is unchanged, $\mathcal{D}$ increases. As a result the numerator's left side will increase and the numerator will increase. Thus, we have shown that as $\mathcal{D}_A$ or $\mathcal{D}_B$ or both increase, the numerator increases too.

In addition, as $\mathcal{D}$ increases, the denominator decreases, since $(1 + (p_B - 1)\mathcal{D})$ decreases, $(1 + (p_A - 1)\mathcal{D})$ decreases, and $p_A p_B \mathcal{D}$, which appears as a negative factor, increases. So, we can conclude that as $\mathcal{D}$, $\mathcal{D}_A$ or $\mathcal{D}_B$ increases, the value of V_A increases. To summarize, as δ, q_A or q_B increases, $\mathcal{D}$ increases, and $\mathcal{D}_A$ or $\mathcal{D}_B$ or both increase too. Thus, as δ, q_A or q_B increases, $\mathcal{D}$ increases, and V_A increases.

We proceed by proving the influence on the equilibrium condition. Manipulating Eq. (10) of Lemma 4.4, the equilibrium holds whenever

$$\frac{v_i}{c_i + \frac{o_i}{p_i}} \geqslant \frac{1}{\mathcal{D}_i p_j} + \frac{\mathcal{D}_j (p_j - 1)}{p_j}.$$

As $\mathcal{D}$ increases, $1/(\mathcal{D}_i p_j)$ decreases. Since $(p_j - 1) < 1$, $\mathcal{D}_j (p_j - 1)/p_j$ decreases too. So, as $\mathcal{D}$ increases, the minimum value of $v_i/(c_i + o_i/p_i)$ for which the equilibrium exists decreases, and therefore an equilibrium exists for more configurations. To summarize, as δ, q_A or q_B increases, $\mathcal{D}$ increases, and the equilibrium exists for more configurations. □

Proof of Lemma 5.1. Consider the equilibrium conditions for agent A, and consider a case in which attempting to answer is beneficial given *history* $= ((1, \ldots, 1), (0, \ldots, 0))$. Suppose that agent A receives a query after that history, and suppose also that both agents follow their equilibrium strategies. In this case, if agent A doesn't answer a punishment of agent A will occur with a probability of $(1 - p_A)^{n-1}$, since $n - 1$ future failures are required in order for a punishment event to occur. If such a punishment will be imposed, this will occur only after at least $n - 1$ queries events to agent A. (A longer delay of n-period may occur if during this time, agent B was punished.)

It is beneficial for agent A to attempt to answer a query, if and only if its expected loss due to answering the query is not more than its loss due to ignoring the query.

i.e., $o_A + p_A c_A + (1 - p_A)ignr\_loss_A(\omega, n, history) \leqslant ignr\_loss_A(\omega, n, history)$. Thus, equilibrium exists if,

$$ignr\_loss_A(\omega, n, history) \geqslant c_A + \frac{o_A}{p_A}.$$

Now, consider a history that is different from $((1, \ldots, 1), (0, \ldots, 0))$. There may be three possible situations of histories:

(1) The last event of agent A was a successful answer: history $((\ldots, 1), (0, \ldots, 0))$.
(2) The end of the history of agent A contains at least one failure: history $((\ldots, 0), (0, \ldots, 0))$.
(3) The history of agent B contains at least one success: history $((\ldots), (\ldots, 1, \ldots)$.

First consider case (1). Any state with the same history of agent B, and where the last event in the history of agent A is 1, is equivalent to ours, in the model where punishing is done after n consequent failures, since a present ignorance of a query by agent A will cause a punishment of A only after additional $n - 1$ unanswered queries.

Now, consider case (2). In any state with one failure or more at the end of agent A's history, $ignr\_loss_A(\omega, n, history)$ will be higher than after a history of $((1, \ldots, 1), (0, \ldots, 0))$, since less failures of agent A are required in order to punish it. (If there are k failures at the end of agent A's history, then the probability of punishment because of a current unanswered query, is $(1 - p)^{n-1-k}$, and this may occur after a delay of at least $n - 1 - k$ queries events.)

Finally, consider case (3). In any case with one success or more in agent B's history, the probability of punishment of agent B is lower than after a history of $((1, \ldots, 1), (0, \ldots, 0))$, since more than one consequent failure of agent B is required in order to punish it. Any punishment of agent B causes a delay in the answers required by agent A. Thus, if the probability of punishing agent B decreases, the expected delay of time until agent A will be punished increases, so $ignr\_loss_A(\omega, n, history)$ also increases.

To summarize, $ignr\_loss_A(\omega, n, history)$ for any given history, is higher than or equal to

$$ignr\_loss_A\big(\omega, n, ((1, \ldots, 1), (0, \ldots, 0))\big).$$

Thus, if after a history of $((1, \ldots, 1), (0, \ldots, 0))$, still

$$ignr\_loss_A\big(\omega, n, ((1, \ldots, 1), (0, \ldots, 0))\big) > c_A + \frac{o_A}{p_A},$$

this will also hold for the other histories. In other words, if the equilibrium exists after a history of $((1, \ldots, 1), (0, \ldots, 0))$ for agent A, and after a history of $((0, \ldots, 0), (1, \ldots, 1))$ for agent B, it will also exist after any history.

Finally, if an equilibrium exists, then each agent will answer the query of its opponent. In particular, each agent will do this given the history of $((1, \ldots, 1), (0, \ldots, 0))$ and the history of $((0, \ldots, 0), (1, \ldots, 1))$. □

Proof of Lemma 5.2. Consider a history of $((x, 1, \ldots, 1), (1, 0, \ldots, 0))$, in the $n + 1$-period model. This means that at least the last $n - 1$ events of agent A are successes, and

exactly the last $n-1$ events were unanswered queries by agent B. Consider now reducing $n+1$ to n. This will increase the probability of punishing agent A after an event of not answering (p_A^{n-1} instead of p_A^n). Moreover, the possible punishment, if performed, may be done earlier (after $n-1$ consecutive failures instead of n), and this will increase the present value of the loss due to punishment.

On the other hand, reducing the number of periods observed in the strategies also causes a possible punishment of agent B to be imposed earlier and with a higher probability. There are three situations:

- If the next event of agent B is a successful answer: then no punishment is given in both models at the next n periods.
- If the next event of agent B is a failure, and the consequent event is a success: then a punishment is given to agent B in the n-period model, and no punishment is inflicted in the $n+1$-period model.
- if the next k (2 or more) consecutive events of agent B are failures: then k punishments of agent B will be performed in the n-period model, and $k-1$ punishments of agent B will be performed in the $n+1$-period model.

Thus, moving from the $n+1$-period model to the n-period model causes at least one more punishment of agent B, given a history of $n-1$ failures of agent B. This one punishment will make the future possible punishment of agent A to be one query event later. Namely, if a punishment event occurs, then agent A is exempt from answering one query, so the possible punishment of agent A is delayed.

To summarize, as $n+1$ decreases to n, punishment of agent A will occur with a probability of $(1-p_A)^n$ instead of $(1-p_A)^{n+1}$, and this possible punishment will be at the same expected future time as in the n-period model, or even earlier. (The possibility of punishing B causes the time of punishment agent A to increase by one query event or to remain unchanged. However, this time of punishment decreases by one query since less failures of agent A are required in order to punish.) Thus, as n increases, $\mathit{ignr\_loss}_A(\omega, n, ((1,\ldots,1),(0,\ldots,0)))$ decreases, and, in particular,

$$\begin{aligned}&\mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big)\\ &\quad\leqslant (1-p_A)\mathit{ignr\_loss}_A\big(\omega, n, ((1,\ldots,1),(0,\ldots,0))\big).\end{aligned}$$ □

Proof of Lemma 5.1. First, we show that for each $\omega \in \Omega_{n+1}$, $\omega \in \Omega_n$. If $\omega \in \Omega_{n+1}$, then, in particular, after a history of last $n-1$ successes of agent A and last $n-1$ failures of agent B, it is still beneficial for agent A to attempt to answer agent B's query. This means that the cost of answering is less than or equal to the expected loss from avoiding answering, i.e.,

$$\begin{aligned}&o_A + p_A c_A + (1-p_A)\mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big)\big)\\ &\quad\leqslant \mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big),\end{aligned}$$

and,

$$\frac{o_A}{p_A} + c_A \leqslant \mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big).$$

According to Lemma 5.2,

$$\begin{aligned} &\mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big) \\ &\quad < \mathit{ignr\_loss}_A\big(\omega, n, ((1,\ldots,1),(0,\ldots,0))\big). \end{aligned}$$

Thus, if

$$\frac{o_A}{p_A} + c_A \leqslant \delta_A \mathit{ignr\_loss}_A\big(\omega, n+1, ((1,\ldots,1),(0,\ldots,0))\big)$$

this holds also for the n-period model, so

$$\frac{o_A}{p_A} + c_A \leqslant \delta_A \mathit{ignr\_loss}_A\big(\omega, n, ((1,\ldots,1),(0,\ldots,0))\big).$$

Based on Lemma 5.1, if answering is beneficial for agent A given a configuration ω, and after $n-1$ consequent successes of agent A and $n-1$ consequent failures of agent B, then it is in equilibrium for each history. Thus, $\omega \in \Omega_n$.

Now, we show that $\omega \in \Omega_n$ exists such that $\omega \notin \Omega_{n+1}$. In particular, we take ω such that

$$\frac{o_A}{p_A} + c_A = \mathit{ignr\_loss}_A\big(\omega, n, ((1,\ldots,1),(0,\ldots,0))\big).$$

An equilibrium exists for this ω given the n-period model. Then, as the strategy profile is changed to be an $n+1$-strategy profile, $\mathit{ignr\_loss}_A$ decreases, so

$$\frac{o_A}{p_A} + c_A < \mathit{ignr\_loss}_A\big(\omega, n+1, ((1,\ldots,1),(0,\ldots,0))\big).$$

In other words, it is no longer beneficial for agent A to attempt to answer the query of agent B, so $\omega \notin \Omega_{n+1}$. □

Proof of Lemma 5.3. Given $\omega \in \Omega$, and given n, we can consider the situation of a history of $((1,\ldots,1),(0,\ldots,0))$, i.e., $n-1$ consequent successes of agent A and $n-1$ consequent failures of agent B. Given n, we can evaluate the expected loss of avoiding answering a query ($\mathit{ignr\_loss}_A(\omega, n, \mathit{history})$) where $\mathit{history} = ((1,\ldots,1),(0,\ldots,0))$, and $\omega \in \Omega_n$ whenever this loss is larger than $c_A + o_A/p_A$, as we explained above.

However, as proved in Lemma 5.2,

$$\begin{aligned} &\mathit{ignr\_loss}_A\big(\omega, n+1, ((x,1,\ldots,1),(1,0,\ldots,0))\big) \\ &\quad \leqslant (1-p_A)\mathit{ignr\_loss}_A\big(\omega, n, ((1,\ldots,1),(0,\ldots,0))\big). \end{aligned}$$

Thus, $\mathit{ignr\_loss}_A(\omega, n, \mathit{history})$ when moving from n to $n+1$ decreases by a factor of at least $1-p_A$, so for $n \to \infty$, $\mathit{ignr\_loss}_A(\omega, n, \mathit{history})$ approaches 0. Therefore, it is also clear that if $\mathit{ignr\_loss}_A(\omega, m, \mathit{history}) > c_A + o_A/p_A$ for a particular m, $m' > m$ exists such that

$$\mathit{ignr\_loss}_A(\omega, m', \mathit{history}) < c_A + \frac{o_A}{p_A},$$

so $\omega \notin \Omega_{m'}$. Denote $n' = \mathit{min}(m')$ such that $\omega \notin \Omega_{m'}$. For each $n' < m'$,

$$\mathit{ignr\_loss}_A(\omega, n', \mathit{history}) \geqslant c_A + \frac{o_A}{p_A},$$

so $\omega \in \Omega_{n'}$.

If $\omega \notin \Omega_1$, then the strategy profile will not be an equilibrium given configuration ω for any length of history. □

Proof of Lemma 5.4. Consider the situation in which agent B answers agent A's query. In this case, agent A has a benefit of v_A. Consider now the situation in which agent B did not answer agent A's query. In the extreme case, this leads agent A to punish agent B, after a delay of $\mathcal{D}_B$. If agent A punishes agent B, agent A saves the cost of $o_A + p_A c_A$ after a delay of one query event of agent B, and it also saves possible losses due to its failure to answer a query it was supposed to answer, i.e., it saves $\mathit{ignr\_loss}_A(\omega, n, \mathit{history})$ with a probability of $1 - p_A$. Thus, we have to prove that

$$v_A > \mathcal{D}_B\big(p_A c_A + o_A + (1 - p_A)\mathit{ignr\_loss}_A(\omega, n, \mathit{history})\big).$$

Since an equilibrium exists, $\omega \in \Omega_n$, thus, according to Lemma 5.1, $\omega \in \Omega_1$, so an equilibrium exists for the one-period model. Thus $\mathit{ignr\_loss}_A(\omega, n, \mathit{history}) \leqslant \mathcal{D}_A v_A$, since one failure event in the one-period model can cause one future punishment (preventing utility of v_A), and this future payment can occur after an expected delay of at least $\mathcal{D}_A$.

For the same reason, since $\omega \in \Omega_1$, it is clear that $o_A + p_A C \leqslant \mathcal{D}_A p_A v_A$ at each stage, since otherwise it is not worthwhile to attempt to answer a query even in the one-period model. Thus, it is enough to show that

$$v_A > \mathcal{D}_B\big(\mathcal{D}_A p_A v_A + (1 - p_A)\mathcal{D}_A v_A\big) \quad \text{or} \quad v_A > \mathcal{D} v_A$$

and this is clear, since $\mathcal{D} < 1$. □

Proof of Theorem 5.2. First, we show it is enough to check the history $((1, \ldots, 1), (1, 0, \ldots, 0))$ with the last $n - 1$ successes of agent A and the last $n - 1$ successes of agent B. If agent A prefers the profile $n + 1$ over profile n, this will remain true after any other history. This is because when moving from a strategy profile of n to a strategy profile of $n + 1$, agent A earns from the fact that its punishment becomes rarer, and agent A loses from the fact that it has to answer more queries of agent B.

As the number of successes of agent A increases, its punishment becomes more rare. Thus, the motivation of agent A to move to profile $n + 1$ decreases. (For the most part it will not be punished in both profiles.) As there are more failures of agent B, its punishment becomes more abundant. Thus, agent A's motivation to move to profile $n + 1$ decreases. (Agent B will be punished in both profiles, so the additional benefits due to punishing agent B more, decreases).

History $((1, \ldots, 1), (1, 0, \ldots, 0))$ includes the highest possible number of agent A's successes and agent B's failures. (An additional failure of agent B will cause a punishment.) Thus, if it is beneficial for agent A to move to profile $n + 1$ given a history of $((x, 1, \ldots, 1), (1, 0, \ldots, 0))$, this will be true after any other history.

Consider a flow of n alternating queries. (n queries from A to B, n queries from B to A), starting with the history of $((1, \ldots, 1), (1, 0, \ldots, 0))$. Denote by $\mathit{exp\_delay}_B(n)$ the expected discount of the future due to punishment of agent B, given that agent B will be punished if its next query fails. Denote by $\mathit{exp\_loss}_A(n)$ the expected discounted loss of agent A when it is supposed to answer agent B's query, when an additional failure of

agent B will cause a punishment of B, and n failures of A will cause a punishment of A. This loss contains a future punishment of agent A after n failures of agent A, i.e., with a probability of $(1-p_A)^n$, discounted by $D^n exp\_failures_B(n+1)$, agent A will lose $p_B v_A$. Thus, $exp\_loss_A(n) = (1-p_A)^n D^n exp\_delay_B(n) p_B v_A$.

In the following, we analyze the difference in agent A's utility, due to changing the strategy profile from profile n to profile $n+1$. We consider the following possible flows of successes/failures of both agents.

- With a probability of p_B, agent B will succeed to answer the current query of agent A. In this case, in both profiles, agent B will not be punished during the next n alternating queries (since there will not be n consequence failures of agent B). In this case, the only possible change from profile n to profile $n+1$ is a possible punishment of agent A. Since we start with $history_A = (1,\ldots,1)$ this may take place only after n failures to answer B's query. During this time, agent B will not be punished, as explained above. Thus, if agent A will be punished in the future, it will lose a value of $p_B v_A$, after a delay of exactly $\mathcal{D}^n$, and with a probability of $(1-p_A)^n$. The expected value of the loss due to this punishment is $(1-p_A)^n \mathcal{D}^n p_B v_A$. (If another failure will take place at time $n+1$, then agent A will be punished in both profiles.)
- With a probability of $1-p_B$, agent B will fail to answer the current query of agent A. Then, after this failure, if profile $n+1$ is in use, agent A is supposed to answer agent B's query, but if profile n is in use, agent A is not supposed to answer this query. Any additional failure of agent B will cause it punishment in both profiles. Consider profile $n+1$. When agent A attempts to answer agent B's query agent A's cost for attempting to answer is $o_A + p_A c_A$. Again, there are 2 situations.
  - With a probability of p_A, agent A will succeed to answer agent B's query. In this case, the last event of A is a success. Thus, in the $n+1$ profile, $n+1$ failures of agent A are required in order to punish it, while in the n profile, the punishment will also be performed after n failures. Therefore, the difference is after n failures of agent A. The expected discounted loss of this failure is $exp\_loss_A(n)$. Thus, when moving to the $n+1$-model, agent A earns $exp\_loss_A(n)$.
  - With a probability of $1-p_A$, agent A will fail to answer agent B's query. In this case, if there are additional n failures of agent A, then agent A will be punished when using the $(n+1)$th model. The probability and delay of this failure is the same as the probability and delay in the nth model, where agent A was not suppose to answer the nth query of agent B. So, in this case, there is no difference between the two profiles.

To summarize, the total expected benefits of agent A from moving from profile n to profile $n+1$, is $p_B D_B((1-p_A)^n D^n p_B v_A) + (1-p_B) p_A D_B D_A exp\_loss_A(n)$ and this should be larger than the cost of answering the $(n+1)$th query of agent B, $D_B D_A(1-p_B)(o_A + p_A c_A)$.

$$p_B D_B\big((1-p_A)^n D^n p_B v_A\big) + (1-p_B) p_A D_B D_A exp\_loss_A(n) > D_B D_A(1-p_B)(o_A + p_A c_A).$$

Manipulating this, we obtain,

$$\frac{p_B}{D_A}\left((1-p_A)^n D^n p_B v_A\right) + (1-p_B)p_A exp\_loss_A(n) > (1-p_B)(o_A + p_A c_A).$$

And since $exp\_delay_B(n+1) < 1$, $exp\_loss_A(n) < ((1-p_A)^n D^n p_B v_A)$, so the above formula is true whenever

$$\left(\frac{p_B}{D_A} + (1-p_B)p_A\right) exp\_loss_A(n) > (1-p_B)(o_A + p_A c_A)$$

or

$$\left(\frac{p_B}{D_A} + (1-p_B)p_A\right) exp\_loss_A(n) > (1-p_B)(o_A + p_A c_A).$$

Since $\omega \in \Omega_{n+1}$, an equilibrium holds in the $n+1$-model. This means that after a history of n failures of B and n successes of A, A prefers to answer than to ignore B's query. A's expected loss due to ignoring is, again, $exp\_loss_A(n)$, since n additional failures of A will cause punishment, while an expected delay due to punishment of agent B is expected, if agent B fails in its next query.

In order for the equilibrium to hold,

$$o_A + p_A c_A + (1-p_A) exp\_loss_A(n) < exp\_loss_A(n),$$

i.e.,

$$exp\_loss_A(n) > \frac{o_A + p_A c_A}{p_A}.$$

We substitute $exp\_loss_A(n)$ with this smaller value, and then it remains to prove that

$$\left(\frac{p_B}{D_A} + (1-p_B)p_A\right)\left(\frac{o_A + p_A c_A}{p_A}\right) \geqslant (1-p_B)(o_A + p_A c_A),$$
$$\left(\frac{p_B}{D_A} + (1-p_B)p_A\right)\frac{o_A + p_A c_A}{p_A} \geqslant (1-p_B)(o_A + p_A c_A).$$

Manipulating this formula, we obtain

$$\frac{p_B}{D_A} \geqslant 0$$

and this is true, since in our model $p_B, D_A \geqslant 0$. □

Proof of Lemma 5.5. Similar to the proof of Lemma 5.1. Consider a configuration in which attempting to answer a query is beneficial given $history = best\_case_A(k, n)$. Suppose that agent A receives a query, and also suppose that both agents follow their equilibrium strategies. In this case, if the current query will not be answered by agent A, it will receive punishment with a probability of $(1-p_A)^{n-1}$, since $n-1$ future failures are required in order for a punishment event to occur. If such a punishment will be implemented, it will occur only after at least $n-1$ queries events to agent A. (A longer delay of n periods may occur if during this time agent B was punished.)

As in Lemma 5.1, it is beneficial for agent A to attempt to answer a query, if and only if its expected loss due to answering the query is not more than its losses due to ignoring the query. i.e.,

$$o_A + p_A c_A + (1 - p_A)\mathit{ignr\_loss}_A(\omega, k, n, \mathit{history}) \leqslant \mathit{ignr\_loss}_A(\omega, k, n, \mathit{history}).$$

Thus, an equilibrium exists if,

$$\mathit{ignr\_loss}_A(\omega, k, n, \mathit{history}) \geqslant c_A + \frac{o_A}{p_A}.$$

Now, consider a history different from $\mathit{best\_case}_A(k, n)$. There may be three possible situations of histories:

(1) There is at least one unanswered query in the $n - 1$ last queries of agent A, but not in the $n - k$ last queries. The history of agent B is the same as in $\mathit{best\_case}_A(k, n)$.
(2) There is at least one unanswered query in the $n - k$ last queries of agent A. The history of agent B is the same as in $\mathit{best\_case}_A(k, n)$.
(3) The history of agent B contains at least one success in the last $k - 1$ events.

First, consider case (1). Since there is no unanswered query by A in the $n - k$ last queries, if agent B will not answer the current query, then a possible punishment may happen only after additional $k - 1$ unanswered queries. Thus, case (1) is equivalent to $\mathit{best\_case}_A(k, n)$.

Now, consider case (2). In any state with one failure or more in $n - k$ last events of agent A's history, $\mathit{ignr\_loss}_A(\omega, k, n, \mathit{history})$ will be higher than after the $\mathit{best\_case}_A(k, n)$ history, since an unanswered query in the present may be concatenated to the other failures, and causes a punishment after a shorter delay than in $\mathit{best\_case}_A(k, n)$.

Finally, consider case (3). In any case with one success or more in the last $k - 1$ queries to agent B's history, the probability of punishment of agent B is lower than after the $\mathit{best\_case}_A(k.n)$ history, since more than one failure of agent B in the near $n - k$ queries is required in order to punish it. Any punishment of agent B causes a delay in the answers required by agent A. Thus, if the probability of punishing agent B decreases, the expected delay of the time when agent A will be punished increases, so $\mathit{ignr\_loss}_A(\omega, k, n, \mathit{history})$ also increases.

To summarize, $\mathit{ignr\_loss}_A(\omega, k, n, \mathit{history})$ for any given history, is higher than or equal to $\mathit{ignr\_loss}_A(\omega, k, n, \mathit{best\_case}_A(k, n))$. Thus, if $\mathit{ignr\_loss}_A(\omega, n, \mathit{best\_case}_A(k, n)) > c_A + o_A/p_A$. This will also hold for the other histories.

In other words, if it is beneficial for agent A to use the equilibrium strategies after the history of $\mathit{best\_case}_A(k, n)$, it will be beneficial for it to use this strategy given any other history. Symmetrically, we can also prove that if it is beneficial for agent B to use the equilibrium strategies given the history of $\mathit{best\_case}_B(k, n)$, agent B will be motivated to use this strategy given any other history. Combining these two results, we can conclude that if the equilibrium exists both for $\mathit{best\_case}_A(k, n)$ and $\mathit{best\_case}_B(k, n)$, it will also exist after any history.

Finally, if an equilibrium exists, then each agent will attempt to answer the query of its opponent. In particular, each agent will do this given the particular history of $\mathit{best\_case}_i(k, n)$. □

Proof of Lemma 5.6. Consider the history of $best\_case_i(k,n)$, and increasing k to $k+1$. This will decrease the probability of punishment of agent i after an event of ignorance, since in the $best\_case_i(k,n)$ history it has no considerable failures. Moreover, the possible punishment, if performed, will be done later (after at least k additional failures instead of $k-1$ additional failures), and this will decrease the present value of the loss due to punishment.

However, $best\_case_i(k,n)$ also depends on k. $best\_case_i(k-1,n)$ is different from $best\_case_i(k,n)$, and it includes $k-2$ consequent failures of agent j instead of $k-1$ (in order to avoid immediate punishment when checking stability). Thus, as k decreases, the probability of punishment of agent j remains unchanged. (Punishment will be inflicted given one more query to agent j that is unanswered.) Thus, enlarging k while changing $best\_case_i$ respectively, causes the threat of punishment of agent j to decrease, while the time and probability of punishing agent i remains unchanged. □

Proof of Theorem 5.3. Consider a strategy profile where punishment is performed after $k2$ unanswered queries from n, and consider a configuration ω for which an equilibrium exists. Since an equilibrium exists,

$$ignr\_loss_A(\omega, k2, n, best\_case_A(k2,n)) > o_A + c_A p_A.$$

Now consider $k1 = k2 - x$. According to Lemma 5.6,

$$ignr\_loss_A\big(\omega, k, n, best\_case_A(k,n)\big) > ignr\_loss_A\big(\omega, k+1, n, best\_case_A(k+1,n)\big).$$

By transitivity, also

$$ignr\_loss_A\big(\omega, k, n, best\_case_A(k,n)\big) > ignr\_loss_A\big(\omega, k+x, n, best\_case_A(k+x,n)\big).$$

So,

$$ignr\_loss_A\big(\omega, k1, n, best\_case_A(k1,n)\big) > ignr\_loss_A\big(\omega, k2, n, best\_case_A(k2,n)\big).$$

Thus, $ignr\_loss_A(\omega, k1, n, best\_case_A(k1,n)) > o_A/p_A + c_A$.

Symmetrically, for agent B, we can also prove that if

$$ignr\_loss_B\big(\omega, k2, n, best\_case_B(k2,n)\big) > o_B + c_B p_B$$

then

$$ignr\_loss_B\big(\omega, k1, n, best\_case_B(k1,n)\big) > \frac{o_B}{p_B} + c_B.$$

Combining these two properties, the conclusion is that if an equilibrium exists for $k2$, it will also exist for $k1 < k2$. □

References

[1] M. Aoyagi, Mutual observability and the convergence of actions in a multi-person two-armed bandit model, J. Econom. Theory 82 (1998) 405–424.
[2] R.M. Axelrod, The Evolution of Cooperation, Basic Books, New York, 1984.
[3] R. Azoulay-Schwartz, Protocols, strategies and learning techniques for reaching agreements more effectively, PhD Thesis, Bar Ilan University, Ramat Gan, Israel, 2001.
[4] V. Bala, S. Goyal, Learning from neighbors, Rev. Econom. Stud. 65 (1998) 595.
[5] D. Carmel, S. Markovitch, Learning models of intelligent agents, in: Proc. of AAAI-96, Portland, OR, 1996, pp. 62–67.
[6] P. Chalasani, S. Jha, O. Shehory, K.P. Sycara, Strategies for querying information agents, in: Cooperative Information Agents II, in: Lecture Notes in Artificial Intelligence, vol. 1435, Springer, Berlin, 1998, pp. 94–107.
[7] Y. Freund, M. Kearns, D. Ron Y. Mansour, R. Rubinfeld, R. Schapire, Efficient algorithms for learning to play repeated games against computationally bounded adversaries, in: Proc. of 36th IEEE Symposium on the Foundations of Computer Science, Milwaukee, WI, 1995, pp. 332–341.
[8] D. Fudenberg, J. Tirole, Game Theory, MIT Press, Cambridge, MA, 1991.
[9] L.P. Kaelbling, A.W. Moore, Reinforcement learning: A survey, J. Artif. Intell. Res. 4 (1996) 237–285.
[10] Y. Mor, Computational approaches to rational choice, Master's Thesis, Hebrew University, Jerusalem, Israel, 1996.
[11] M.J. Osborne, A. Rubinstein, Bargaining and Markets, Academic Press, San Diego, CA, 1990.
[12] D.C. Parkes, L.H. Ungar, Learning and adaption in multiagent systems, in: Proc. of AAAI-97 Workshop on Multiagent Learning, Providence, RI, 1997.
[13] R. Radner, Repeated principal agents games with discounting, Econometrica 53 (1985) 1173–1198.
[14] W. Raub, J. Weesie, Reputation and efficiency in social interactions: An example of network effects, Amer. J. Soc. 96 (3) (1990) 626–654.
[15] T.W. Sandholm, R.H. Crites, Multiagent reinforcement learning in the iterated prisoner's dilemma, Biosystems (1995) 147–166.
[16] A. Schaerf, Y. Shoham, M. Tennenholtz, Adaptive load balancing: A study in multi-agent learning, J. Artif. Intell. Res. 2 (1995) 475–500.
[17] S. Sen, N. Arora, Learning to take risks, in: Proc. of AAAI-97 Workshop on Multiagent Learning, Providence, RI, 1997, pp. 59–64. Workshop Notes available as AAAI Technical Report WS-97-03.
[18] S. Sen, M. Sekaran, Individual learning of coordination knowledge, J. Experimental Theoret. Artif. Intell. 10 (3) (1998) 333–356.
[19] S. Sen, A. Biswas, S. Debnath, Believing others: Pros and cons, in: Proc. of ICMAS-2000, Boston, MA, 2000, pp. 279–286.
[20] S. Sen, Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents, in: Proc. of ICMAS-96, Menlo Park, CA, 1996, pp. 322–329.
[21] F. Seredynski, Coevolutionary game-theoretic multi-agent systems: The application to mapping and scheduling problems, in: Z.W. Ras, M. Michalewicz (Eds.), Foundations of Intelligent Systems, in: Lecture Notes in Artificial Intelligence, vol. 1079, Springer, Berlin, 1996.
[22] G. Zacharia, Collaborative reputation mechanisms for online communities, Master's Thesis, MIT, Cambridge, MA, 1999.