Anticipatory Behavior of Software Agents in Self-organizing Negotiations

Jan Ole Berndt and Otthein Herzog

Abstract Software agents are a well-established approach for modeling autonomous entities in distributed artificial intelligence. Iterated negotiations allow for coordinating the activities of multiple autonomous agents by means of repeated interactions. However, if several agents interact concurrently, the participants' activities can mutually influence each other. This leads to poor coordination results. In this paper, we discuss these interrelations and propose a self-organization approach to cope with that problem. To that end, we apply distributed reinforcement learning as a feedback mechanism to the agents' decision-making process. This enables the agents to use their experiences from previous activities to anticipate the results of potential future actions. They mutually adapt their behaviors to each other which results in the emergence of social order within the multiagent system. We empirically evaluate the dynamics of that process in a multiagent resource allocation scenario. The results show that the agents successfully anticipate the reactions to their activities in that dynamic and partially observable negotiation environment. This enables them to maximize their payoffs and to drastically outperform non-anticipating agents.

1 Introduction

In distributed artificial intelligence, software agents model autonomous entities which plan and perform their activities in multiagent systems. These autonomous agents are able to proactively select their actions, to react to changes in their environment and to interact with each other [31]. In the latter context, iterated negotiations are a well-established means for coordinating distributed systems containing multiple agents. The participating agents can negotiate on allocations of

J.O. Berndt (✉) · O. Herzog
Center for Computing and Communication Technologies (TZI),
Universität Bremen, Bremen, Germany
e-mail: joberndt@tzi.de

M. Nadin (ed.), *Anticipation Across Disciplines*,
Cognitive Systems Monographs 29, DOI 10.1007/978-3-319-22599-9_15

resources, delegations of tasks, as well as commissions of services. This enables them to identify appropriate partners which complement their own capabilities in order to meet their individual objectives [10, 23].

Nevertheless, a problem occurs if several of these interactions take place concurrently. In this situation, the participants' activities can mutually influence each other. That is, the outcome of each negotiation depends on those being performed simultaneously. This is particularly the case in joint negotiations of cooperating agents which require them to compromise about their desired agreements. In order to enable efficient and robust multiagent coordination, the agents have to take these interdependencies into account when selecting and evaluating their respective actions in iterated negotiations. That is, they must adapt their behavior to the activities of others.

In a competitive setting, a game theoretical equilibrium [19] denotes a combination of each individual agent's best response to the others' behaviors. However, acting in a partially observable environment, the agents are unable to explicitly compute such an equilibrium. Therefore, we propose to approximate it by means of agents adapting their activities to each other. Inspired by Niklas Luhmann's theory of social systems [15, 17], our approach enables these agents to anticipate the reactions of others to their own actions. Thus, they can select best responses to the expected behaviors of others. To that end, we apply distributed reinforcement learning to the agent decision-making in iterated multiagent negotiations. Using this technique, each agent learns a best response behavior to the others' activities without the necessity to observe them directly. This results in a self-organizing system of mutually interdependent activity selections in which social order emerges from the agents' concurrent learning efforts.

We structure this paper as follows. Section 2 elaborates on concurrent iterated negotiations and discusses their challenges as well as existing approaches to address them. Subsequently, Sect. 3 presents the main contribution of this paper which is threefold. Firstly, we model concurrent negotiations as repeated games and propose multiagent learning for coordinating them. Secondly, we discuss Luhmann's notion of self-organization in social systems and its adaptation for multiagent coordination. Thirdly, we introduce decentral decision-making criteria for terminating multiagent negotiations. Section 4 empirically evaluates this approach in a distributed resource allocation scenario. This evaluation confirms the ability of learning agents to successfully anticipate each other's behaviors and provides insights into the dynamics of that process. Finally, Sect. 5 concludes on the achievements of this paper and outlines directions for future research.

2 Iterated Multiagent Negotiations

Iterated multiagent negotiations denote a process of distributed search for an agreement among two or more participants [13]. This process consists of the negotiation objects, an interaction protocol, the participating agents, and their decision-making mechanisms. The negotiation objects determine the search space

of potential agreements. In the process, the agents exchange proposals which their counterparts can either accept or reject. While the protocol defines the possible sequences of messages, an agent selects its actions among those possibilities by means of its decision-making mechanism. If the agents find a mutually acceptable agreement according to their individual preferences, the search returns this solution as its result. Otherwise, it terminates without success.

In the following, we further elaborate on these aspects of multiagent negotiations. In particular, Sect. 2.1 examines negotiation objects and protocols. This provides the foundations for discussing the challenges of agent decision-making as well as existing approaches to cope with these challenges in Sect. 2.2.

2.1 Negotiation Objects and Protocols

The negotiation objects define the topic on which the participating agents attempt to reach agreement [13]. They cover the target of a negotiation such as the desired service fulfillment or resource allocation. Moreover, they denote the cardinality of these items: Either single or multiple ones. In the latter case, the agents negotiate on possible combinations of the target products or services. Many-object negotiations require them to identify a mutually acceptable compromise out of the range of those combinations. In the following, we focus on many-object negotiations as they subsume the special case of single-object ones. Furthermore, they are equivalent to cooperative activities of several agents attempting to achieve common goals. In that case, several agents group together in teams [22, 25, 32]. These teams negotiate as composite entities in order to further their common objectives while competing with other teams or individual agents.

To structure the negotiation process, there are two basic protocol types for exchanging proposals [13]. In auction type negotiations, one or more agents exclusively propose potential agreements while the others only accept or reject them. An example for this is the Dutch auction in which the auctioneer repeatedly decreases the proposed price until one or more buyers accept the current offer. Contrastingly, in negotiations of the bargaining type the agents bilaterally exchange offers and counter-offers. Hence, they mutually attempt to steer the search in their individually favored direction. On the one hand, this increases the speed of reaching an agreement; on the other hand, it requires all participants to be capable of both evaluating and generating meaningful proposals [10]. In this paper we mainly focus on negotiations of the auction type. Nevertheless, in Sect. 3.3 we also suggest to adapt our approach to bargaining type interactions.

A well-known protocol for iterated auction type negotiations is the FIPA Iterated Contract Net [11] as depicted in Fig. 1. It is particularly suitable for situations in which a consumer agent attempts to find the best partner among the potential providers of a required service or product. In many-object negotiations, this can also be a set of agents if no single participant is able to fulfill the initiator's demands on its own. However, this approach requires the initiator to address all potential

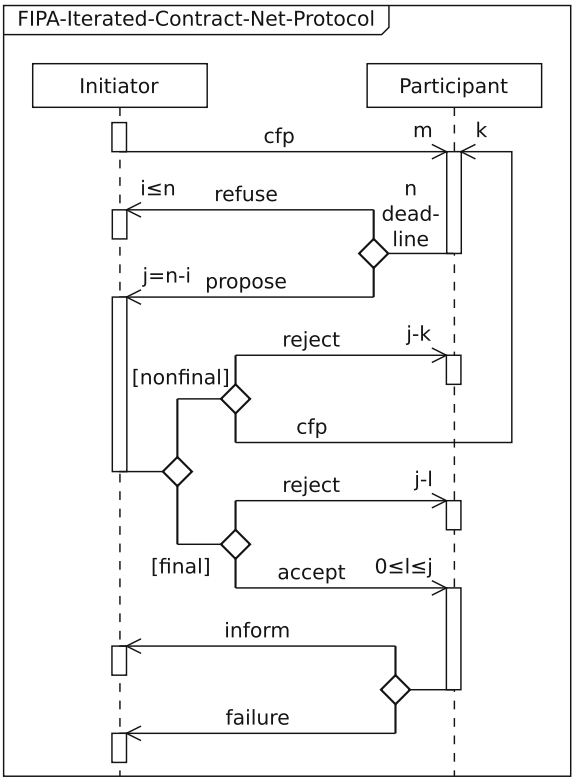

Fig. 1 The FIPA Iterated Contract Net interaction protocol (adapted from [11])

participants from the beginning on as there is no way to include additional agents during the process. If the initial selection is insufficient to fulfill the initiator's requirements, the whole negotiation will fail.

2.2 Agent Decision-Making: Challenges and Related Work

If there is exclusively one single initiator agent at any time, its decision-making in the aforementioned protocol is simple. It only requires to keep track of the participants' offers to identify the currently best agreement, *accept* it when no further improvements occur, and *reject* all other proposals. However, this is not the case if several of these interactions take place concurrently. In this situation, the participants receive several *cfp* messages simultaneously and their subsequent responses depend on all of these messages. Consequently, these interactions mutually influence each other's outcome as the initiator agents compete for the participants' limited capacities. In order to still achieve the best possible result of the negotiation, an agent must take the actions of all others into account. That is, it has to find a best response to its counterparts' behaviors.

To illustrate the aforementioned problem, Fig. 2 depicts a simple resource allocation example. This scenario consists of two consumer agents (A,B) acting as initiators of concurrent negotiations. They attempt to allocate resources from the provider agents (C,D) of which each has only enough capacity to fulfill the request of one consumer. If each consumer contacts both providers simultaneously, there are four different possible outcomes. Only two of those lead to a successful negotiation result for both consumers. In the other two cases, a single consumer receives two offers. Because it can only accept one of them, the other provider's resource remains unused due to its refuse message terminating the negotiation with the unsuccessful consumer agent. Hence, the agents have a 50 % chance of achieving an efficient overall coordination result.

In the general case of a set N of initiators and a set M of participants, an efficient allocation is equivalent with a surjective mapping (i.e., an onto function) from M to N. Consequently, the probability for achieving such a result is given by the possible number of these mappings [18, pp. 84–85;90] divided by the number of all possible interactions.

$$P_{eff} = \frac{1}{|N|^{|M|}} \cdot \sum_{j=0}^{|N|} \binom{|N|}{j} \cdot (-1)^j \cdot (|N| - j)^{|M|} \tag{1}$$

Figure 2 shows this probability for varying agent populations. As the number of consumers increases, a drastically higher supply of resources (i.e., number of providers) is necessary to ensure a near efficient coordination result. This holds for both the standard Contract Net protocol as well as its iterated version because in the latter, a *refuse* message terminates the interaction with its sender. Consequently, subsequent iterations can only refine the result of the first one which renders this protocol inadequate for concurrent negotiations.

To overcome the limitations of the Iterated Contract Net, we slightly modify the original FIPA protocol of Fig. 1. Instead of narrowing the set of participants to a subset of the initial receivers in each iteration, we allow for including alternative

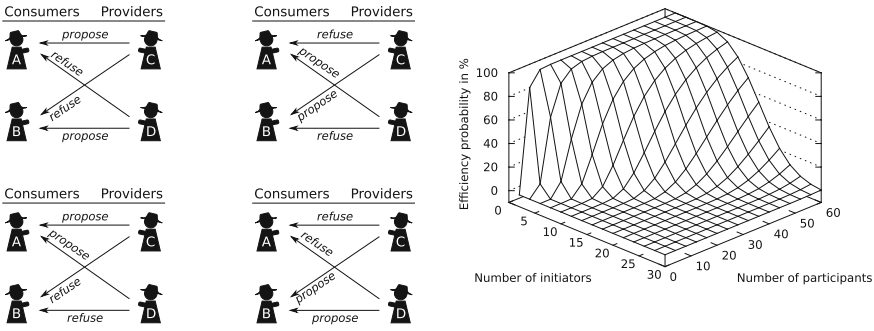

Fig. 2 Resource allocation options and probability of achieving an efficient outcome for varying agent populations in the (Iterated) Contract Net

ones while keeping their overall number constant. That is, the initiator selects a fixed number m of participants and replaces those $m-k$ which propose none or only unacceptable agreements (*refuse/reject*) with alternative candidates. Thus, it refills the set of receivers for the next iteration's *call for proposals* (*cfp*) with x new ones to a size of $k + x = m$. This enables an initiator to contact participants more than once, even if they refused their earlier allocation attempts.

Nonetheless, the agents must still find best responses to each other's activities in the modified Iterated Contract Net protocol. This is due to the fact that their activities can still collide which leads to suboptimal outcomes. While they have the ability to continue their negotiations despite failed allocation attempts, they must avoid these collisions in future iterations of the interaction. That is, they must *anticipate* their counterparts' behaviors and adequately respond to them to secure their intended negotiation results. This anticipation is crucial for achieving the desired outcomes because otherwise the agents would mutually disturb their efforts. To facilitate that end, the following concepts and methods for finding best responses are available from related work.

Determining best responses to other agents' activities is the subject of game theory [29]. If all agents pursue a best response strategy to the behaviors of the others, these strategies form a Nash equilibrium [19] in which no single agent can benefit from changing its current behavior. A Nash equilibrium denotes the agents' best possible activities in such a strictly competitive setting. Moreover, by approximating best responses to the others' behaviors, an agent maximizes its individual payoff, even if they fail to establish a corresponding best response in return. Therefore, each agent should select its actions in an iterated negotiation with respect to the others' activities.

Existing methods for computing an equilibrium of mutual best responses often evaluate the structure of the game and are computationally expensive [20]. Nevertheless, each agent only has to identify its own best strategy. Consequently, it requires a decision-making method for finding its most beneficial activities, given the actions of the others. A well-known technique for this is the *minimax* rule [28] of 2-player decision-making and its generalization for n-player settings [14]. By assuming the others to pursue their most beneficial courses of actions, this rule selects the best response to those behaviors. As a result, an equilibrium emerges from the agents' mutually dependent action selections.

However, in concurrent negotiations, the *minimax* approach requires an agent to be aware of the other participating agents, their possible actions as well as their preferences (i.e., their scoring functions for the interaction's outcome). For competitive distributed negotiations, disclosing these trade secrets is inappropriate [10]. Consequently, the agents act in a partially observable environment. In this environment, they must coordinate their negotiation behaviors while preserving the privacy of information. To achieve the latter, *combinatorial auctions* [8] provide a means for computing the best allocation of goods or services in a mediated interaction process. In these auctions, the participants express their preferences as bids on combinations of offered items. While such a bid represents the result of an agent's valuation of an offer, it hides the agent's private method for attaining that

assessment. Moreover, combinatorial auctions are particularly suitable for many objects as the participants can express bids on arbitrary combinations. Nonetheless, the winner determination is a centralized process which creates a computational bottleneck [21]. This is undesirable in distributed systems.

To overcome this problem, agents should adapt their behaviors during a negotiation according to their experiences throughout that process [27]. Hence, we propose to enable the agents to learn best responses to each other's actions from observations of their personal performance. Deriving beliefs about successful behavior from the outcome of past interactions has been shown to enable the approximation of market equilibria in repeated trading activities [12]. That is, buyers and sellers determine mutually acceptable prices for the traded items by estimating the probabilities of reaching an agreement for potential price offers. Nevertheless, this requires the presence of a common currency to express those prices.

In order to allow for best responses according to generic utility assessments, we rather apply *reinforcement learning* [26] to multiagent negotiations. This technique enables the agents to anticipate the expected results of their actions by observing and learning from the outcome of their previous activities. By adapting their behavior accordingly, they can establish of social order within the negotiation through a process of self-organization. They implicitly generate interaction practices which reflect the identified best responses to the unobservable activities of their competitors. To accomplish this, an agent receives a reward when performing an action from which it learns an estimation of the expected reward for potential future actions. Subsequently, it can select the next action based on this estimation. Multiagent reinforcement learning [6] has been applied successfully to approximate best response behaviors in distributed coordination tasks [5, 7]. Therefore, it is a promising approach for determining an agent's most beneficial strategy in concurrent iterated negotiations.

3 Multiagent Self-organization in Iterated Negotiations

In the following, we apply multiagent reinforcement learning to concurrent iterated many-object negotiations. Section 3.1 interprets them as repeated games and provides a formal notation for the agents' decision-making environments and behaviors. Subsequently, Sect. 3.2 motivates our approach to social self-organization, introduces its sociological foundations, and applies a stateless version of the *Q-learning* approach to agent decision-making. Finally, Sect. 3.3 discusses criteria for determining acceptable offers to terminate such a negotiation.

3.1 Iterated Multiagent Negotiations as Repeated Games

In order to facilitate a better understanding of the interdependencies of concurrent agent activities in iterated negotiations, we formalize them using the terminology of

game theory and reinforcement learning. From this point of view, a single iteration of a multiagent negotiation is a *static (stateless) game*. In such a game, each of the agents performs one action and receives a reward depending on all simultaneously executed actions. Its formal definition is as follows [6].

Definition 1 (*Static Game*) A static game is a triple $\left(N, \vec{A}, \vec{R}\right)$. N is a set of agents being indexed *1,..., n.* Each agent $i \in N$ has a finite set of atomic actions A_i. Thus, $\vec{A} = (A_1, \ldots, A_n)$. $\vec{R} = (R_1, \ldots, R_n)$ is the collection of individual reward functions for each agent *i.* Each $R_i : A_1 \times \cdots \times A_n \rightarrow \mathbb{R}$ returns *i's* immediate reward for the simultaneous execution of agent actions $a_1, \ldots, a_n$ with $\forall j \in N : a_j \in A_j$.

In a concurrently executed Contract Net, the set of agents N consists of the initiators of the simultaneous negotiations. Each of them selects a participant to send its *call for proposals* specifying the negotiation object. Thus, agent i's individual actions A_i contain all of these possible messages in conjunction with their respective receivers. Instead of distributing the rewards directly, the participants subsequently respond with a *proposal* or a *refuse* message. A participant's response depends on the entirety of messages it received in the current iteration. Each initiator can rate its individually received response by calculating its respective payoff (i.e., the negotiation's expected outcome if it accepts the received offer). Thus, an agent obtains the conditional reward for its action, even though it is unable to observe the actions of the others. Iterating this one-shot negotiation several times results in a *stage game* [6]. This repeated game describes the agents' decision-making environment during concurrent iterated negotiations. Only in its final iteration, an agent bindingly *accepts* or *rejects* its received offer. Until then, it can use the stage game to learn the most beneficial actions for that last static game.

In order to accomplish this learning, the agents repeatedly observe the payoff of their respective activities which enables them to reason about their expected reward in further iterations. A rational agent has the objective to maximize its personal payoff. Hence, it attempts to adopt a behavior which is a best response to the other agents' actions. In game theoretical terms, a deterministic best response strategy returns an action which maximizes an agent's payoff, given the actions of all others [6].

Definition 2 (*Best Response*) A best response of agent $i \in N$ to the other agents' actions $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$ is an action a_i^* which leads to the highest reward given those activities: $\forall a_i \in A_i : R_i\left(a_1, \ldots, a_i^*, \ldots, a_n\right) \geq R_i(a_1, \ldots, a_i, \ldots, a_n)$.

In a competitive environment, each agent strives to maximize its individual payoff on its own. Therefore, all agents mutually attempt to find a best response to each other's activities. Such a situation, in which no single agent can beneficially deviate from its current behavior, forms a Nash equilibrium [19]. For deterministic agent strategies, this is defined as follows.

Definition 3 (*Nash Equilibrium*) A Nash equilibrium is a vector $\left(a_1^*, \ldots, a_n^*\right)$, such that $\forall i \in N$, each action a_i^* is a best response to the others.

A Nash equilibrium does not ensure that the agents maximize their common payoff in the form of a social welfare optimum.[1] Nonetheless, it denotes each agent's best possible action relative to the others' activities if all agents attempt to maximize their individual payoff. The objective of each agent in concurrent iterated negotiations is to identify such a best response action in order to select it in its decision-making.

Additionally, there are negotiations in which not all agents compete with each other. If two or more agents pursue a common goal, they have to negotiate together in order to acquire the necessary resources or commission required services. In this case, these agents can group together in teams [23]. The set of those multiagent teams $MT \subset 2^N$ is a partition of the set of individual agents. The members of each team $mt \in MT$ cooperate in their interactions. To that end, they combine their individual rewards in a common *social welfare function*.

Definition 4 (*Social Welfare Function*) Asocial welfare function of team $mt \in MT$ maps all team members' rewards to a single value: welfare: $\mathbb{R}^{|mt|} \rightarrow \mathbb{R}$.

A team's welfare indicates the joint performance of its member agents by aggregating their individual rewards. Several different aggregation methods are available for implementing that function [9]. The most common of those is the *utilitarian welfare function* which returns the sum of the team members' rewards: $\sum_{i \in mt} R_i(a_1, \ldots, a_n)$.

In a negotiation, a team acts as a single initiator agent. That is, a particular member agent $mgr \in mt$ becomes the *team manager*. That agent sends cfp messages on behalf of all members and collects the respective rewards for the responses. Then, it aggregates them in the team's welfare function. This is equivalent to a single agent negotiating several objects. As a result, multiagent teams attempt to find joint best responses to other teams' as well as to individual agents' activities. This replaces the member agents' rewards in Definition 2 with the team's welfare. Consequently, a Nash equilibrium consists of the best combination of actions for the team given the non-members' best possible responses to those activities.

However, both individual agents and multiagent teams are unable to directly determine whether their concurrent activities form a Nash equilibrium. This is because there is no entity which can observe all of these behaviors. Instead, they must derive the best responses solely from their payoffs for the performed actions. If all agents and teams succeed in this endeavor, a Nash equilibrium emerges from their distributed efforts. To that end, the next section specifies our approach to *self-organizing negotiations* which relies on the anticipation and an adaptation of agent behaviors.

[1] A famous example for this is the prisoner's dilemma in which the equilibrium point is the only strategy combination not belonging to the Pareto frontier.

3.2 *Anticipation and Behavior Adaptation for Iterated Negotiations*

Niklas Luhmann's sociological theory of communication systems [15, 17] provides a fundamental inspiration for our approach to self-organizing negotiations. According to this theory, social order derives from actors mutually expecting each other's activities. These expectations emerge from the actors' interactions rather than reflecting static behavioral norms or fixed channels for communication. An actor observes his counterpart's behavior and selects his activities according to the other's expected reaction. Thus, an actor's action selection depends on observed activities of others and vice versa. This feedback loop of observation and expectation enables social structures to emerge from an initial state of ignorance [2] by means of interaction processes. These structures guide subsequent executions of those very processes. Luhmann refers to the generation of social structures by the term *self-organization* [16].

In previous work, we have applied expectations to the decision-making of software agents [1–5]. These agents memorize the observable effects of their own activities. Each time it has to select an action, such an agent evaluates its options according to its memory entries. That is, it searches for an action which it expects to predictably lead to an advantageous response. After executing the selected action, it observes the actual response by the addressed agent and updates its memory with that observation. That process either increases or decreases the agent's expectation for the selected activity depending on whether it under- or overestimated its outcome. This renewed expectation then becomes available for the anticipation of activity results in further interactions.

The aforementioned process enables a software agent to anticipate the outcomes of its activities without having to know their exact causes, the identities of its competitors, and their respective capabilities. To that end, it assumes its past observations to be representative for future events. It learns which of its potential interaction partners best to select in order to reach its goals. In a negotiation, this allows for an initiator agent to identify those participants which can offer the most advantageous deals. To achieve this effect, we model the process of generating expectations and selecting activities according to them by means of *reinforcement learning* [26]. In a stage game, this technique allows for the agent to learn from its experiences to increase its future performance.

A well-understood algorithm for the case of one single learning agent is *Q-learning* [30]. In its stateless form, this algorithm estimates expected rewards (action payoffs) as *Q-values Q(a)* for each possible action *a* [7]. A learning agent

[2] In this state of *double contingency*, both participants are unable to act because each of their activities depends on the other's previous actions and they lack any existing expectations for selecting them. However, Luhmann notes that this is a highly unstable fixpoint of the interaction's dynamics which never actually occurs in real encounters [15, 17]. Instead, every slight action allows for generating initial expectations which facilitate the emergence of social order.

uses the following update rule to refine its estimation when observing a reward *R(a)* for action *a*.

$$Q(a) \leftarrow Q(a) + \lambda \cdot (R(a) - Q(a)) \tag{2}$$

If each action is sampled infinitely often, the agent's Q-values converge to the unobservable true values Q^* for every learning rate λ with $0 < \lambda \leq 1$ [7, 30]. This enables the learning agent to select its activities according to their expected payoff values. Hence, as the values converge, it can identify its individually optimal action.

However, in concurrent iterated negotiation processes, several initiator agents act simultaneously. This results in interdependent effects of their actions as formalized in the preceding section. In fact, the convergence property of single agent Q-learning does not hold for a distributed setting in which several agents simultaneously adapt their behaviors. This is because their interdependent activities result in non-stationary rewards. These rewards depend on the combination of all concurrently executed actions. Consequently, an agent can observe changing effects of its actions without being able to influence them or to identify the cause of these changes. For instance, if two agents attempt to allocate resources from two resource providers, the first initiator may receive offers or rejects of its attempts depending on the simultaneous actions of the other initiator. Even if the first agent always selects the same action, it will be unable to accurately anticipate the reaction because the second agent may change its behavior. Thus, an agent's interaction environment changes during its learning process which can render its existing expectations invalid.

The same situation arises in social systems. According to Luhmann, communications are guided by expectation structures in these systems [15, 17]. Through repeated changes and mutual adaptations, these structures stabilize themselves and social order emerges. The reason for this effect lies in the reciprocal nature of expectations. All actors simultaneously generate and refine their expectations. In this process, they narrow the range of actually occurring communications within the system. This increases the likeliness of communications being successful. Hence, the participating actors can mutually anticipate each other's reactions to their activities and act accordingly instead of arbitrarily changing their behaviors. While they retain the ability to react in an unexpected manner, this makes communications sufficiently predictable to facilitate goal-directed social coordination.

In the following, we transfer the preceding considerations to concurrent multi-agent negotiations. If all agents in that setting develop expectations about the outcomes of their activities and their actions depend on those expectations, those very outcomes become increasingly predictable. This is because they narrow the range of selected actions. If they also maximize the payoffs they receive from the corresponding responses, the agents establish a Nash Equilibrium of mutual best response activities. Nevertheless, conventional reinforcement learning is unable to bring about that effect. It suffers from several agents mutually disturbing their adaptation efforts by changing their behaviors. When an agent perceives an action to yield inferior outcomes, it has to change its selection and search for an adequate

alternative option. This change can interfere with the activities of another agent. That agent is then also obliged to modify its behavior. Therefore, a chain reaction of adaptations can occur in which disturbances build up and the agents are unable to obtain social order. To avoid this and instead enable the interactions to converge to social order, the agents' action selection method must fulfill two additional conditions [5, 7].

1. At any time, every possible action of an agent must have a non-zero probability of being selected.
2. An agent's action selection strategy must be asymptotically exploitive.

Condition 1 ensures the infinite sampling of all agent actions for $t \rightarrow \infty$. An agent must always have the opportunity to explore alternative courses of action to be able to react to changes of other agents' behaviors which affect its own performance. Furthermore, that condition prevents the agents from executing strictly correlated explorations. That is, no combination of agent actions becomes impossible to occur. This is an extension of the infinite sampling requirement for single agent Q-learning: In a multiagent setting, each combination of actions must be executed infinitely often due to the payoff's dependence on all concurrently triggered other actions. Condition 2 requires the agents to pursue a *decaying* exploration strategy. This decreases the probability of concurrent exploration activities over time. Hence, the agents become less likely to disturb each other as their behaviors become increasingly predictable. As a result, their expectations can settle to stable social structures. Empirical evidence shows that these agents successfully establish mutual best responses in a variety of settings [5, 7].

In order to apply this technique to iterated negotiations, we construct the initiator agent's behavior as depicted in Fig. 3. This behavior extends the message passing activities as specified in the FIPA Iterated Contract Net protocol definition [11] with an initialization step as well as the following repeatedly executed activities.

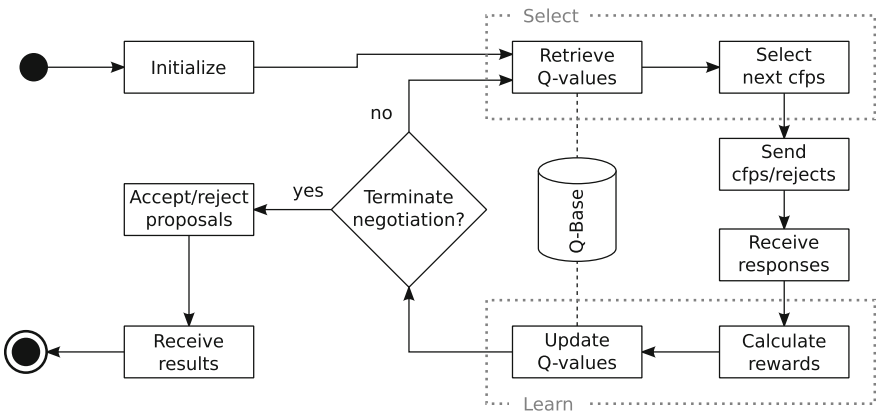

Fig. 3 Behavior of a learning initiator agent in the Iterated Contract Net

1. *Selecting* the receivers and contents for the next *calls for proposals*.
2. *Learning* from the observed responses.
3. *Deciding* on whether to terminate or continue the negotiation.

When entering a negotiation, each learning agent $i \in N$ initializes its Q-Base (i.e., its memory) Q_i in which it stores the expected payoffs $Q_i(a_i)$ for all its possible atomic actions $a_i \in A_i$. Its individual actions A_i consist of all *cfp* messages, given by their possible contents and receivers. The message contents depend on the agent's preferences toward the negotiation object and the receivers correspond to the possible providers of that object. In the case of a multiagent team, the team manager maintains such a memory for each of the member agents. The following considerations cover the decision-making of such a team manager because it subsumes the special case of an individual agent (being equivalent to a team with a single member).

Subsequently, the agent enters the iterated part of the negotiation. To select the next action, it considers all messages $a_i \subseteq A_i$ and looks up their stored Q-values $Q_i(a_i)$. A team manager does this for every member agent individually. In that case, maintaining a Q-base for the atomic actions instead of their combinations keeps the required storage space small when using a lookup table [5]. Nevertheless, this requires the corresponding rewards $R_i(a_i)$ to be mutually independent. This is because the team manager must aggregate those Q-values in the team's welfare function to identify the expectedly most beneficial message combinations $maxA_{mt} \in A_1 \times \cdots \times A_{|mt|}$.

$$maxA = \arg \max_{A_{mt} \in A_1 \times \cdots \times A_{|mt|}} \left(\sum_{i \in mt} Q_i(a_i) \right) \tag{3}$$

with $a_i \in A_{mt}$ being the selected action for team member i

Choosing an action set from $maxA_{mt}$ corresponds to a greedy strategy which maximizes the team's expected payoff based on its experiences so far. For that purpose, Eq. 3 computes the utilitarian welfare of the expected action outcomes. This method maximizes the average payoff of the team's members without favoring particular ones over others as long as the expectations accurately anticipate the actual outcomes. However, a team manager is unable to guarantee this because it is initially unaware of the available deals in the negotiation and other agents can change their behaviors which may provide potentials for improving its performance. Hence, in order to find out whether there is an even better option, the agent also has to explore alternative actions.

To this end, we propose to use an ε-greedy strategy. That is, in iteration t of the negotiation, the manager selects the next actions $A_{mt,t}$ from $maxA_{mt}$. with a probability of $1 - \varepsilon$ (with $0 < \varepsilon \leq 1$). If there is more than one best option, it chooses randomly among them. Alternatively, with a probability of ε, the agent selects $A_{mt,t}$ at random out of all action combinations in $A_1 \times \cdots \times A_{|mt|}$. Moreover, to ensure

the aforementioned asymptotically exploitive selection with non-zero probabilities, it employs a decaying ε-greedy strategy. This requires a sequence ε_n with $\lim_{t\to\infty}\varepsilon_t = 0$ and $\forall t \in \mathbb{N} : \varepsilon_t > 0$. An example meeting these requirements is the following quence: $\forall t > 0 : \varepsilon_t = \frac{1}{\ln(t+2)}$. This sequence leads to high exploration rates in the beginning of the negotiation which decrease over time. Once an agent has identified a highly rated combination of actions, it increasingly tends to stick to those actions as time proceeds.

After selecting the next actions, sending the chosen messages, and collecting the respective responses, the team manager proceeds with the learning part of its behavior. To assess the usefulness of the selected actions $A_{mt,t}$, it evaluates the response messages $result(a_i, t)$, $\forall a_{i,t} \in A_{mt,t}$ for each member agent $i \in mt$ by means of an individual utility measure $U_i : \{result(a_i) | \forall a_i \in A_i\} \to [0, 1]$. It uses the result of this calculation as the action's immediate reward $R_i(a_{i,t})$.

$$R_i(a_{i,t}) = U_i(result(a_{i,t})) \tag{4}$$

As the response messages depend on the concurrent actions of all agents participating in the negotiation, their utility implicitly reflects these actions as well. Thus, it is sufficient for the team manager to evaluate only the observable responses instead of receiving a conditional reward for all simultaneous activities. In order to learn from this observation, it subsequently applies the standard update rule as in Eq. 2 to modify its stored Q-value $Q_i(a_{i,t})$ for all performed actions $a_{i,t} \in A_{mt,t}$. In the succeeding iteration, the refined entries in the Q-Base serve as the new Q-values for these actions.

According to the aforementioned convergence property of the Q-learning rule, an infinite number of these iterations will lead to each agent and multiagent team learning to anticipate the best response to the others' activities. Hence, a Nash equilibrium will emerge from these distributed learning processes in concurrent negotiations. Nonetheless, an infinite negotiation never comes to a final result. To avoid this, each negotiation initiator must decide after an iteration either to accept its received response messages as the result and terminate the negotiation or to continue it in the attempt to reach a better outcome. That is, while learning the best behavior for the repeated interaction process, it must eventually apply its findings to one single iteration to bring about a result of the negotiation. To facilitate this decision-making, the next section discusses individual tactics for terminating iterated negotiations.

3.3 Termination of Iterated Negotiations

A learning agent as specified in the preceding section is unable to determine whether it has already developed a best response behavior or not. Furthermore, it cannot guarantee that stable social structures have emerged among all negotiating

agents. This is because it would have to know all other agents' possible actions as well as their actual selections, the participants' respective responses, and the agents' utility measures for evaluating these responses.

However, disclosing this information is inappropriate for competitive negotiations (cf. Section 2.2). As an alternative, negotiation *tactics* enable reaching individually acceptable agreements without requiring additional information. These tactics model an agent's bidding behavior in bargaining type negotiations consisting of offers and counter-offers. They can depend on the amount of *time* or other *resources* being available as well as on the observable bidding *behavior* of the negotiation opponents [10]. Such a tactic provides a function which approaches the agent's private *reservation value* in the course of the negotiation. This value denotes the minimal offer it is willing to accept. Thus, unless the agents come to a better agreement at some time during the negotiation, the reservation value denotes its last offer on which it insists until the end of the negotiation. If at some point in time neither agent concedes any further, the negotiation terminates without success.

In contrast to bargaining negotiations, in auction type mechanisms like the Iterated Contract Net it is unnecessary to generate counter-offers. Instead, the initiator agents only require a decision function which indicates whether or not one or more received proposals are acceptable. To this end, an agent must consider the payoff of the current offers. These values are already available from the reinforcement learning algorithm (Eq. 4). Thus, we define agent i's decision function in dependence of its utility measure U_i for evaluating the perceived results of its actions (with the manager of a multiagent team using the utility measures of all member agents). In analogy to the bargaining tactics, the agent has a *reservation level* of utilities U_{res}. This is the minimum utility it will accept for the *last offers* of the negotiation. If the reservation level turns out to be unreachable, it will terminate the negotiation without coming to an agreement.

However, in order to maximize its payoff, the agent must explore alternative actions in the course of the negotiation. Therefore, it should abstain from choosing the first option exceeding its reservation level as the final one. Only if it fails to achieve a better result, the agent should accept the current offer. To this end, we introduce an agent's *acceptance level* of utilities U_{acc} which denotes the minimum utility for the *current offer* to be acceptable. In the case of a multiagent team, the common welfare of the member agents denotes that utility. As the team manager attempts to maximize the members' joint payoff, it has to compare the team's welfare to the acceptance level in order to evaluate whether the outcome for the team is acceptable or not. Varying over time during a negotiation, the acceptance level resembles an agent's tactic in bargaining: It consists of a function describing the agent's behavior of conceding to its reservation level. To enable the agent to benefit from its learning ability, this function starts from a sufficiently high value and decreases monotonically over time. As a result, the agent rejects all but the best offers in the early iterations. Nevertheless, it becomes increasingly inclined to compromise about that utility during the negotiation process.

Following from these considerations, a team manager successfully terminates a negotiation in iteration t if the received offers' aggregated utility exceeds the current

acceptance level: $U_{acc,t} < \sum_{i \in mt} U_i\big(result\big(a_{i,t}\big)\big)$ with $a_{i,t} \in A_{mt,t}$ being the selected action for team member i. That is, the team manager computes the welfare of the whole team and decides whether the result is acceptable as the negotiation's outcome. Furthermore, it terminates the process without success if the acceptance level falls below the reservation level: $U_{acc,t} < U_{res}$. In the latter case, the team failed to reach an agreement with its interaction partners under the least acceptable conditions. Figure 4 depicts these termination criteria for a range of acceptance level functions. Analogously to the concession behaviors in bargaining negotiations, these functions tend toward either the well-known *Boulware* or the *Conceder* tactics [10]. While the former attempts to reach a highly valued agreement as long as possible, the latter quickly approaches the reservation level.

To implement these tactics, we modify the polynomial time dependent function presented in [10] according to the aforementioned considerations. In the resulting function, the acceptance level $U_{acc,t}$ in iteration t ranges between the initial value $U_{acc,0}$ and the reservation level U_{res} as long as t adheres to a given deadline t_{max}. Moreover, the acceptance level is strictly monotonically decreasing if $U_{acc,0} > U_{res}$ and t_{max} is constant.

$$U_{acc,t} = U_{acc,0} - \big(U_{acc,0} - U_{res}\big) \cdot \left(\frac{t}{t_{max}}\right)^{\beta} \tag{5}$$

According to this equation, the negotiation is guaranteed to terminate for all $t_{max} < \infty$. The parameter β controls the agent's concession behavior: While it pursues a Boulware tactic if $\beta > 1$, each $\beta < 1$ leads to a Conceder behavior. The intensity of these tactics increases the more β deviates from 1 (with $\beta = 1$ denoting the neutral linear tactic).

By means of Eq. 5, an agent controls its negotiation behavior. Setting t_{max} to a fixed point in time allows for modeling situations in which the agents must finish their negotiation before some deadline exceeds. In conjunction with the

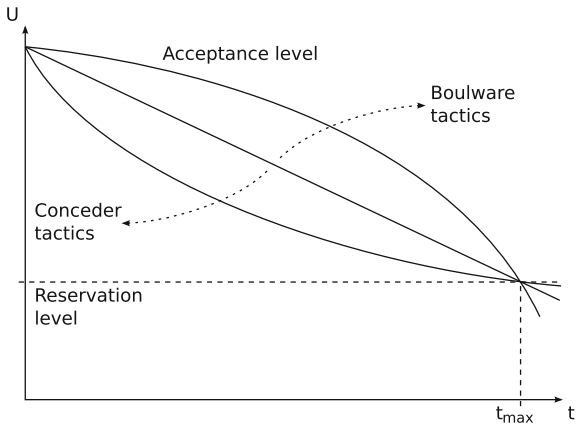

Fig. 4 Termination criteria based on acceptance and reservation utility levels

reinforcement learning technique, this termination method enables agents in concurrent multiagent negotiations to adjust their behaviors according to each other's distributed activities. While the learning approach facilitates an agent's anticipation of best responses to the unobservable behaviors of others, the termination criteria control the negotiation's duration. Moreover, deriving from negotiation tactics in bargaining, the latter even offer the possibility to transfer this approach to bilateral negotiations. As the acceptance level denotes the minimum utility for an agreement, an agent can invert its utility measure to generate counter-proposals to the perceived offers. If a common currency is used, this is easy to accomplish by mapping the learned values to price offers [12]. However, we leave this adaptation as well as the analysis of its requirements and implications to future research.

4 Evaluation

In this section, we evaluate our approach to self-organizing multiagent negotiations in a multiagent simulation. This evaluation covers the dynamics of the agents' learning efforts as they establish expectations to anticipate the behaviors of their interaction partners. In the following, Sect. 4.1 describes the design of the simulation experiments while Sect. 4.2 presents and discusses the results.

4.1 Experiment Design and Setup

In order to evaluate the proposed learning approach in iterated multiagent negotiations, we apply it to a distributed resource allocation problem using the simulation system PlaSMA [24]. Our scenario is an abstraction from a kind of problems occurring frequently in real-world applications like production scheduling and logistics [5]. This scenario contains a set N of resource consumer agents which concurrently negotiate with the resource providers in set M as depicted in Fig. 5. The member agents of these sets are indexed 1,…,n for the consumers and 1,…,m for the providers. In addition, the set of consumers is partitioned into n teams of size k. Consequently, only every kth consumer takes an active part in the negotiation as a team manager. Each consumer team requires k resource units while every provider

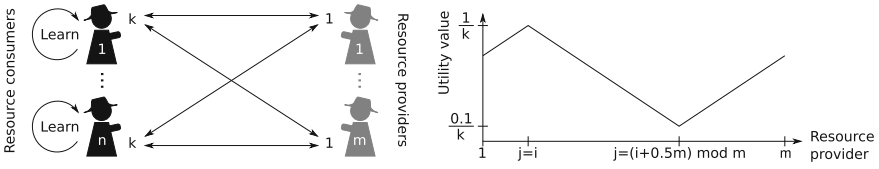

Fig. 5 Many-object resource allocation scenario with n consumers and $m = n \cdot k$ providers

has exactly one unit available. Because $|M| = |N|$, there is sufficient supply for fulfilling that demand. Thus, the agents have to find an appropriate bijection between the set of consumers and the set of providers. In this case, each consumer allocates its required amount of resources without interfering with the others.

To approximate a mutual best response allocation in that setting, the team managers act as initiators of a concurrent iterated negotiation. In each iteration, a manager selects k providers for a *call for proposals*, one for each team member. If a provider still has its resource unit, it sends an offer for the allocation; otherwise it sends a *refusal*. In the case of a provider receiving two or more allocation attempts, it randomly selects one consumer for its offer and *refuses* all other *cfps*. The initiators evaluate these responses by means of the following utility function for each team member.

$$U_i\left(ra_{i,t}\right) = \frac{1}{k} \cdot \begin{cases} \frac{|0.5m - |i-j||}{0.5m} \cdot 0.9 + 0.1 & \text{if } ra_{i,t} \text{ is a } propose \text{ message} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

with

$$\forall a_{i,t} \in A_{mt,t} : ra_{i,t} = result\left(a_{i,t}\right)$$

According to Eq. 6, each agent $i \in N$ has an individual utility function. If the response to the selected action is an offer, its utility ranges between $\frac{0.1}{k}$ and $\frac{1.0}{k}$ depending on the respective sender. Otherwise, it is zero. Hence, the usefulness of the different provider's resources varies for each consumer. Figure 5 depicts the resulting function. There is only one provider offering an optimal payoff. Because these providers differ for all consumers, there is exactly one optimal resource allocation (namely that allocation which maps all consumers to the providers with the same index). Being unaware of the described scenario and the actions of other agents, this optimum is difficult to achieve for the team managers. In its attempts to maximize its performance, a team manager has to find the best activities for each of its members while competing with the managers of other teams for those results. This requires it to search for resource providers which reliably offer high payoffs. The agents must anticipate these outcomes in order to maximize their performance because an arbitrary selection of actions and mutual disturbances will lead to poor coordination results.

Our evaluation assesses the capability of the proposed approach to approximate an allocation with the aforementioned properties. It focuses on the agents' learning dynamics in order to evaluate the impact of their self-organization during the course of a negotiation. To this end, we test it in a scenario with a set of 1200 consumer agents which we subdivide into 20 teams of 60 members each. We vary the team managers' learning rates λ between zero and one in order to evaluate their impact on the learning dynamics. In this context, $\lambda = 0$ means that an agent maintains no expectations at all. Thus it selects every action at random. This serves as a baseline configuration to mark the lower bound of the expectable coordination performance.

For each agent $i \in N$ and every atomic action $a_i \in A_i$, we set the initial Q-values to $Q_i(a_i) = 0$. This leads to a purely explorative behavior in the beginning of the negotiation and in case of repeated refusals. This initialization and randomized action selection avoids a premature over-estimation of potential agreements. Nonetheless, as soon as an agent observes a (partially) successful combination of actions, it utilizes the ε-greedy strategy to exploit its experience. Thus, the agent increasingly tends to stick to those actions which have been beneficial in past iterations.

To terminate the negotiation, the agents employ a time dependent heuristic as specified in Eq. 5. They use an initial acceptance level of $U_{acc,0} = 1$, a reservation level of $U_{res} = 0.0$, a Boulware negotiation tactic ($\beta = 3$), and a deadline of $t_{max} = 800$ iterations. The Boulware tactic increases the impact of their learning as the agents slowly concede to their reservation levels. Each experiment consists of 120 simulation runs.

4.2 Experiment Results and Discussion

Figure 6 depicts the average number of consumer agent teams participating in the negotiation over time for varying learning rates. It shows that the agents' learning efforts significantly reduce the time required for identifying an acceptable negotiation result. While the non-learning agents require more than 700 iterations for most of them to terminate their negotiations, the learning rates of $\lambda = 0.2$ and $\lambda = 0.4$ achieve this in about 500 iterations. The higher learning rates result in durations between those values. These results indicate that the generation of social order has a large impact on the time required for finding an appropriate resource allocation. The team managers learn which resource providers to contact in order to receive advantageous offers. Thus, they tend to repeatedly select those options which

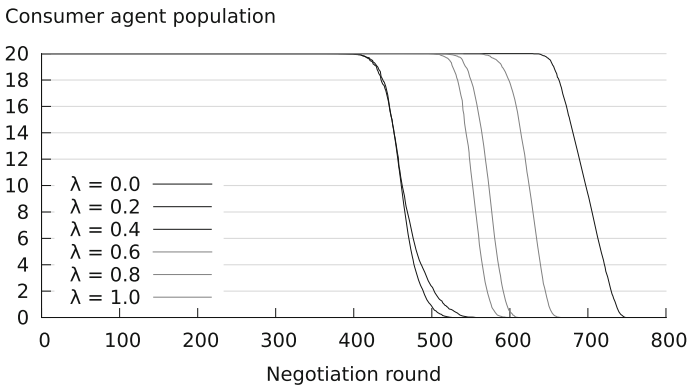

Fig. 6 Number of teams participating in the negotiation over the course of time

provide high payoffs. Although they occasionally explore alternative ones, they only adopt them if these actions provide a significant advantage over the already known activities.

Moreover, maintaining expectations for every single action of individual team members enables the team managers to systematically change their selections for those individual members. Thus, their activities become both increasingly stable and successful for small learning rates which leads to early identifications of acceptable results. By contrast, higher learning rates ($\lambda \geq 0.6$) lead to faster adoptions of alternative activities. This can lead to mutual disturbances between the multiagent teams. Hence, they require more time for their negotiations (while still being superior to a non-learning approach).

Nevertheless, the duration of a negotiation is only loosely connected to the actually achieved result quality. To complement the preceding results from that perspective, Fig. 7 depicts the development of the average team welfare during the negotiation. This confirms the aforementioned effects of the learning rate. The random action selection results in largely constant welfare values at a low level around 0.35.

Contrastingly, the learning approach leads to gradually increasing welfare values over time. This particularly holds for small learning rates. Therefore, the agents adopt successful behaviors and refine them if they manage to find superior options for specific actions. As their activities become increasingly predictable, they learn to anticipate the corresponding outcomes. This is evident in the later iterations where the welfare increases rapidly. In these iterations, the first teams terminate their negotiation processes by permanently allocating the offered resources. Other agents cannot receive any further offers from the corresponding providers. Consequently, the results of those actions become perfectly predictable.

The more teams finish their negotiation, the easier it is for the remaining ones to adapt their behaviors accordingly. While the average welfare for these agents is still suboptimal, its development shows that they are able to establish expectations to successfully anticipate and increase the outcomes of their activities. This enables

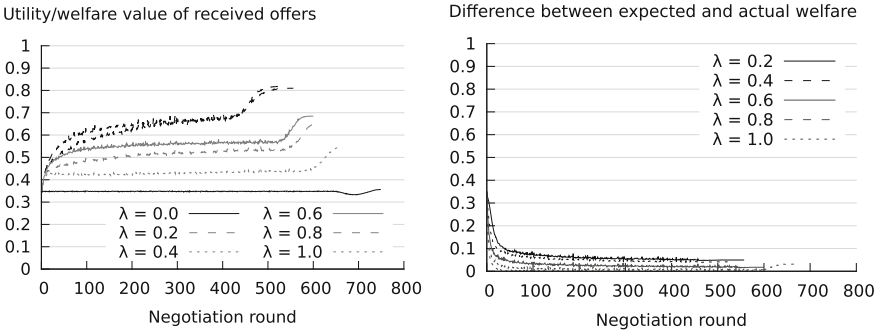

Fig. 7 Development of the received offers' welfare as well as differences between the expected and actually observed outcomes for the teams over time

them to drastically outperform non-anticipating agents. In particular, the anticipative approach improves the final result by up to more than 130 % (final result of 0.818 ± 0.001 for $\lambda = 0.4$ compared to 0.356 ± 0.001 for $\lambda = 0.0$).[3]

Finally, Fig. 7 also presents the differences of the aforementioned observed welfare values and the expected ones for the selected actions. A small difference denotes an accurate anticipation of the results while a large one indicates an agent's failure to expect the actual outcome of its activities. The figure shows that all positive learning rates lead to a convergence of these differences toward zero. As a result, the agents are able to anticipate their negotiation partners' offers. High learning rates lead to even smaller deviations from the real outcomes. This confirms that the agents rapidly adapt their expectations in that case which leads to the discussed tendency to disturb each other. Because they are equally as fast in their reactions to those disturbances, they retain their expectations' accuracy. However, this hampers their ability to generate stable social structures. By contrast, agents with lower learning rates accept slightly larger deviations from their expectations without overreacting to them. This leads to the previously observed higher performance and the successful emergence of social order.

5 Conclusions and Outlook

In this paper we have proposed the application of multiagent reinforcement learning to concurrent iterated negotiations. We have analyzed the standard negotiation mechanism for multiagent coordination. This analysis has shown that the mechanism is unable to ensure successful negotiation outcomes. To overcome its shortcomings, our approach enables negotiating agents to anticipate each other's behaviors and adapt their own activities accordingly. In that context, the agents can group together to cooperate with each other within a team while several of these teams still compete for the best negotiation results.

The anticipation of their activities' effects allows for the agents' distributed approximation of best responses to their counterparts' actions without requiring them to directly observe those actions. Taking inspiration from Luhmann's theory of social systems [15, 17], we enable the learning agents to derive expectations from their received offers. The resulting behaviors are generated in a self-organizing process of anticipation and adaptation. Therefore, they are an emergent effect of the agents' concurrent learning efforts. The agents approximate this result by means of individual decision criteria for the termination of a negotiation process.

For the empirical evaluation of this approach, we have applied it to a multiagent resource allocation scenario. The results show that the learning agents successfully anticipate each other's behaviors. Their performance in terms of negotiation time and achieved payoff depends on their applied learning rates. If these rates are too small,

[3]All deviations are half-widths of the 99 % confidence interval.

they are unable to develop any expectations at all. If they are too large, the agents tend to overreact to their observations. Consequently, they require a balanced parameter setup to facilitate the generation of stable social structures. In that case, their adaptation method enables them to achieve high payoffs in small amounts of time. Nevertheless, all tested parameters lead to (drastic) improvements of the agents' average performance in comparison with a non-anticipative benchmark setting.

To summarize, the contributions and results of this paper are as follows.

- Anticipation enables software agents to select adequate activities in a partially observable negotiation setting.
- Social systems theory provides valuable inspiration for implementing anticipative behaviors in artificial agents. Their mutual anticipation of those behaviors leads to the emergence of social order among multiple agents.
- Anticipative behaviors improve the performance of software agents in negotiations by up to more than 130 % (in the evaluated setting with the tested parameter values).

Nevertheless, there are still questions open for future research. While we have briefly mentioned the possibility to transfer our method to bargaining type negotiations, its actual implementation and evaluation will be subject to future work. Moreover, additional analyses of our existing approach will facilitate a better understanding of its components and their interaction. In particular, to guarantee the convergence of the reinforcement learning part to mutual best responses, an analytical assessment of self-organizing negotiations is necessary. Additionally, further empirical evaluations will focus on different scenarios with heterogeneously parameterized populations to assess the capabilities and limitations of distributed learning for the anticipation of agent behaviors in concurrent iterated negotiations.

References

1. Berndt, J.O.: Self-organizing supply networks: autonomous agent coordination based on expectations. In: Filipe, J., Fred, A. (eds.) ICAART 2011, vol. 2, pp. 104–113. SciTePress, Rome (2011)
2. Berndt, J.O.: Self-organizing logistics process control: an agent-based approach. In: Filipe, J., Fred, A. (eds.) Agents and Artificial Intelligence, pp. 397–412. Springer, Berlin (2013)
3. Berndt, J.O., Herzog, O.: Efficient multiagent coordination in dynamic environments. In: Boissier, O., Bradshaw, J., Cao, L., Fischer, K., Hacid, M.S. (eds.) WI-IAT 2011, pp. 188–195. IEEE Computer Society, Lyon (2011)
4. Berndt, J.O., Herzog, O.: Distributed learning of best response behaviors in concurrent iterated many-object negotiations. In: Timm, I.J., Guttmann, C. (eds.) MATES 2012, pp. 15–29. Springer, Berlin (2012)
5. Berndt, J.O., Herzog, O.: Distributed reinforcement learning for optimizing resource allocation in autonomous logistics processes. In: Kreowski, H.J., Scholz-Reiter, B., Thoben, K.D. (eds.) LDIC 2012, pp. 429–439. Springer, Berlin (2013)
6. Buşoniu, L., Babuška, R., De Schutter, B.: Multi-agent reinforcement learning: an overview. In: Srinivasan, D., Jain, L. (eds.) Innovations in Multi-Agent Systems and Applications—1, pp. 183–221. Springer, Heidelberg (2010)

7. Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. In: AAAI 1998. pp. 746–752. Madison, USA (1998)
8. Cramton, P., Shoham, Y., Steinberg, R. (eds.): Combinatorial Auctions. The MIT Press, Cambridge (2006)
9. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Negotiating socially optimal allocations of resources. J. Artif. Intell. Res. **25**, 315–348 (2006)
10. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robot. Auton. Syst. **24**(3–4), 159–182 (1998)
11. Foundation for Intelligent Physical Agents: FIPA Iterated Contract Net Interaction Protocol Specification. Standard (2002), document No. SC00030H
12. Gjerstad, S., Dickhaut, J.: Price formation in double auctions. Game. Econ. Behav. **22**(1), 1–29 (1998)
13. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M.J., Sierra, C.: Automated negotiation: prospects. Methods Chall. Group Decis. Negoti. **10**, 199–215 (2001)
14. Luckhart, C., Irani, K.B.: An algorithmic solution of N-person games. In: AAAI 1986. vol. 1, pp. 158–162. Morgan Kaufmann, Philadelphia, USA (1986)
15. Luhmann, N.: Soziale Systeme. Grundriß einer allgemeinen Theorie. Suhrkamp, Frankfurt (1984)
16. Luhmann, N.: Probleme mit operativer Schließung. In: Luhmann, N. (ed.) Die Soziologie und der Mensch, Soziologische Aufklärung, vol. 6, pp. 12–24. Westdeutscher Verlag, Opladen (1995)
17. Luhmann, N.: Social Systems. Stanford University Press, Stanford (1995)
18. Mazur, D.R.: Combinatorics. A guided tour. MAA Textbooks, The Mathematical Association of America, Washington (2010)
19. Nash, J.: Non-cooperative Games. Ann. Math. **54**(2), 286–295 (1950)
20. Porter, R., Nudelman, E., Shoham, Y.: Simple search methods for finding a Nash equilibrium. Game. Econ. Behav. **63**(2), 642–662 (2008)
21. Ramezani, S., Endriss, U.: Nash social welfare in multiagent resource allocation. In: David, E., Gerding, E., Sarne, D., Shehory, O. (eds.) Agent-Mediated Electronic Commerce, pp. 117–131. Springer, Heidelberg (2010)
22. Schuldt, A.: Multiagent coordination enabling autonomous logistics. Springer, Heidelberg (2011)
23. Schuldt, A., Berndt, J.O., Herzog, O.: The interaction effort in autonomous logistics processes: potential and limitations for cooperation. In: Hülsmann, M., Scholz-Reiter, B., Windt, K. (eds.) Autonomous Cooperation and Control in Logistics, pp. 77–90. Springer, Berlin (2011)
24. Schuldt, A., Gehrke, J.D., Werner, S.: Designing a simulation middleware for FIPA multiagent systems. In: Jain, L., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N., Cao, L. (eds.) WI-IAT 2008, pp. 109–113. IEEE Computer Society Press, Sydney (2008)
25. Schuldt, A., Werner, S.: Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time. In: Müller, J.P., Petta, P., Klusch, M., Georgeff, M. (eds.) MATES 2007, pp. 121–132. Springer, Berlin (2007)
26. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
27. van Bragt, D.D.B., La Poutré, J.A.: Why Agents for Automated Negotiations Should Be Adaptive. Netnomics **5**(2), 101–118 (2003)
28. von Neumann, J.: Zur Theorie der Gesellschaftsspiele. Math. Ann. **100**, 295–320 (1928)
29. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
30. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3–4), 279–292 (1992)
31. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10** (2), 115–152 (1995)
32. Wooldridge, M., Jennings, N.R.: The cooperative problem-solving process. J. Logic Comput. **9**(4), 563–592 (1999)