Collaborative Agents for Complex Problems Solving

Minjie Zhang, Quan Bai, Fenghui Ren, and John Fulcher

Abstract. Multi-Agent Systems (MAS) are particularly well suited to complex problem solving, whether the MAS comprises cooperative or competitive (self-interested) agents. In this context we discuss both dynamic team formation among the former, as well as partner selection strategies with the latter type of agent. One-shot, long-term, and (fuzzy-based) flexible formation strategies are compared and contrasted, and experiments described which compare these strategies along dimensions of Agent Search Time and Award Distribution Situation. We find that the flexible formation strategy is best suited to self-interested agents in open, dynamic environments. Agent negotiation among competitive agents is also discussed, in the context of collaborative problem solving. We present a modification to Zhang's Dual Concern Model which enables agents to make reasonable estimates of potential partner behavior during negotiation. Lastly, we introduce a Quadratic Regression approach to partner behavior analysis/estimation, which overcomes some of the limitations of Machine Learning-based approaches.

1 Introduction

Complex problem solving typically requires diverse expertise and multiple techniques. Over the last few years, Multi-Agent Systems (MASs) have come to be perceived as a crucial technology, not only for effectively exploiting the increasing availability of diverse, heterogeneous, and distributed on-line information resources, but also as a framework for building large, complex, and robust distributed information processing systems which exploit the efficiencies of organized behaviour. MAS technology is particularly applicable to complex problem solving in many application domains, such as distributed information retrieval [22], traffic monitoring systems [32], and Grid computing [35], etc.

Minjie Zhang, Quan Bai, Fenghui Ren, and John Fulcher
School of Computer Science and Software Engineering, University of Wollongong, Wollongong NSW 2522, Australia
e-mail: {minjie,quan,fr510,john}@uow.edu.au

C.L. Mumford and L.C. Jain (Eds.): Computational Intelligence, ISRL 1, pp. 361–399.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

A MAS comprises a group of agents, which can collaborate when dealing with complex problems, or alternatively perform tasks individually with high autonomy. In a MAS, agents can be characterised as either 'self-interested' or 'cooperative' [21] [34]. When different types of agents work together, management of their interactions is a very important and challenging issue for the success of MASs.

This chapter introduces two main approaches for complex problem solving via agent cooperation and/or competition, these being (i) a partner selection strategy among competitive agents, and (ii) dynamic team forming strategies among cooperative agents.

This chapter is organised as follows. Section 2 provides some background knowledge and definitions relevant to agents and MASs. In Section 3, a dynamic team-forming approach for MASs in open environments is introduced, which can be used among both cooperative and self-interested agents. In Section 4, a fuzzy logic approach for partner selection among self-interested agents via agent competition is discussed in detail. The chapter concludes and further research outlined in Section 5.

2 Self-interested and Cooperative Multi-Agent Systems

2.1 Traditional Classification

Agent activities are driven by their goal(s), and according to the properties of these goal(s), can be classified as either 'self-interested' (competitive) or 'cooperative' (benevolent) agents [20] [21]. These two types of MAS can be defined as follows:

Definition 1. A MAS that contains agents with distinct or even competitive individual goals is defined as a self-interested MAS.

Generally, an agent of a self-interested MAS collaborates with other agents to realise or maximise their local utilities.

Definition 2. A MAS that contains agents with common goals is defined as a cooperative MAS.

Normally, agents of a cooperative MAS work together toward maximising the realization of their common goal(s).

An example of a cooperative MAS application is RoboCup [4] [5] [18]. In a robot soccer team, all robot players (agents) collaborate to achieve their common goal, i.e., winning the game. A typical example of a self-interested MAS is an agent-based e-Commerce system in an electronic marketplace [15] [23] [39] [40]. In an electronic marketplace, different agents work in the same environment toward non-cooperative individual goals. However, agents still need to collaborate with others in order to maximise their individual utilities, i.e. purchase/sell items collaboratively in order to obtain the best price(s).

2.2 The Blurred Boundary

As the sophistication of MASs increases, the traditional classification of 'self-interested' versus 'cooperative' MAS becomes impractical and unreasonable in many domains [42]. In many MAS applications, a MAS can neither be a simple market system nor an agent colony. The boundary between self-interested and co-operative MASs thus becomes blurred [20] [42]. This is mainly due to the following reasons:

1. In many current MAS applications, agents can come from different organisational entities. These agents work together because the organisations they belong to have some cooperative relationships [27]. Therefore the terms and conditions of this cooperation between individual agents mainly depend on the higher-level relationship between the organisations. This kind of MAS is not purely self-interested because of the existence of common goal(s) among the agents. However such MASs can neither be classified as typically cooperative because cooperation between agent members are facile and depend not only on the system's overall utility but also on many outside factors.
2. In many MAS applications, self-interested agents are also required to take care of the global system utility via temporal cooperation in order to maintain and improve their working environments. As the social welfare of the system increases, all system members, including self-interested agents, will benefit.
3. A MAS can include agents from different organisational entities. This leads to an agent in the MAS having different attitudes toward different targets. An agent can be cooperative with agents from the same organisation as itself, yet act in a self-interested manner with agents of other organisations. Therefore, a MAS could be a system comprising both self-interested and cooperative agents. In this situation, it is difficult to identify whether the MAS is cooperative or self-interested.
4. Even within the same organisation, cooperative agents may also behave in a self-interested way due to their limited local view [16] [42].

2.3 Two Scenarios

In Sections 3 and 4, we introduce first a team-forming mechanism for cooperative problem solving via agent cooperation, followed by a partner section approach for collaboration via agent competition, in various types of agent systems.

The following two scenarios will be used in Section 3 and Section 4, respectively, to demonstrate the application of our proposed approaches supporting by experimental results.

Scenario 1

In a general service composition system, a number of services need to be combined together to execute a task in the system. For instance, if we want to transport goods overseas, we have to combine several kinds of services together, which might include packing service, road transport service, custom elated service and shipping service. An agent in a service composition system is normally used to represent a particular service, and the resource of the agent is the service that the agent can provide. In such a system, agents must work with each other like a team in order to achieve the desired goal i.e. to execute tasks cooperatively because each task must be accomplished by more than one services.

Scenario 2
A car buyer wants to purchase a car. However, there are several prospective sellers. To avoid extensive negotiation with each seller, the buyer should filter out some 'impossible' car sellers. For example, if a car seller's bid is much higher than the buyer's expectation or the seller's reputation cannot be trusted by the buyer, then the buyer will filter out such car sellers by employing the partner selection approach before the negotiation starts. During the negotiation, in order to maximise self's profit, the car buyer can predict its negotiation partner's behaviors and make corresponding responses. For example, for a car buyer in a hurry, if he estimates that a car seller cannot make further concession, then he will not spend more time on the current bargaining but looks for another possible seller. On the other hand, for a patient car buyer, if he estimates that a car seller still has scope to make future concessions, then the car buyer will make more effort on the bargaining. Therefore, by employing the behaviours prediction approach, the agent can get some advantages in bargaining.

In distributed and complex problem solving, many MAS applications face a similar situation as Scenario 1, such as Web-based grid computing, distributed information gathering, distributed monitoring systems, automated design and production lines. Scenario 2 is a typical example for self-interested MASs in the domain of e-commerce and frequently happens in wide agent-based e-trading and e-market places. Section 3 and Section 4 introduce the detail definitions and principles about two proposed approaches for agent collaboration, and also demonstrate experimental results about how to achieve agent collaboration through dynamic team formation in Scenario 1, and how to achieve agent collaboration by using a partner selection strategy in Scenario 2, respectively.

3 Collaborative Problem Solving through Agent Cooperation

As introduced in the previous section, MASs can be classified as either self-interested or cooperative, according to the features of agent goals. However, cooperation is unavoidable in most MASs regardless of whether or not they are cooperative or self-interested. Due to the distributed nature of the problem to be solved, and because of limitations in agent abilities, in many cases agents need to work together on some tasks (i.e. via cooperation).

Agent abilities are limited. To perform tasks beyond its inherent ability, an agent needs to collaborate with other agents through joining or forming a particular organisation. The organisation of a group of agents is the collection of roles, relationships and authority structures which govern agent behaviours [14]. All MASs possess some form of organisation to support agent interactions. The form of organisation guides how the agent members interact with each other. An agent team is a kind of organisational structure that supports agent cooperation. Generally speaking, each agent team is composed of a team leader and several team members. After an agent joins a team, it will cooperate with other team members towards a common goal.

In current MAS research, MAS team formation is faced with a number of challenges, especially with regard to the following two aspects:

- Many current multi-agent systems (MASs) are required to work in open and dynamic environments [1] [13] [37] [38]. Uncertainties of dynamic environments obstruct coherent teamwork and bring difficulties for agent cooperation. In dynamic environments, system constraints, resource availability, agent goals, etc. are all changeable. Changing any of these factors may directly require a MAS to deal with different situations. In a new situation, retaining outdated cooperative relationships may obstruct agents in achieving their individual goals.
- Compared with cooperative agents, cooperation among self-interested agents is more complicated and dynamic, due to their selfish features. Self-interested agents are impelled to cooperate with others by their individual goals (due to limited individual abilities). In an agent team composed of self-interested agents, temporary cooperation among agents might conflict with the selfish goals of individual agents as the environment changes. In open and dynamic environments, if factors such as agent goals, task requirements and resources change, a selfish agent may need to modify or even terminate the cooperative relationships with its colleagues, otherwise the cooperation would be in conflict or even be harmful to the individual agent goal. Considering this point, some researchers suggest using dynamic agent cooperation strategies in this kind of application. However, how long cooperation should be maintained among particular agents is always a problem.

In many MAS applications, a dynamic team-formation mechanism is needed to enable agents to automatically form and reform groups/teams to avoid profit conflicts between agents in line with changes in the environment. Toward this objective, a number of researchers try to find an optimal mechanism for dynamic team formation and member selection [30] [36] [37] [38]. Generally, in current MAS research, there are two kinds of team-formation mechanisms in widespread use, these being one-shot team formation and long-term team formation. These team-formation mechanisms are described below:

- *One-shot team-formation mechanism (for temporal cooperation)*
  In self-interested MASs, an individual agent's willingness and goals are important factors that need to be considered during team formation. Research on team formation for self-interested agents generally focuses on forming one-shot teams, also called short-term teams, for individual tasks. In this kind of mechanism, agents come together when they need to handle some tasks, and their relationships will be terminated after the tasks have been accomplished.
- *Long-term team-formation mechanism (for long-term cooperation)*
  Obviously, one-shot teams can experience frequent grouping and regrouping among agents. Each grouping/regrouping consumes some communication and

computation resources. To overcome the weakness of one-shot team formation, Rathod and desJardins proposed several stable-team formation strategies for self-interested MASs [30]. These strategies allow self-interested agents to form long-term relationships in order to reduce team formation overhead. However, for many self-interested MASs, agent goals or willingness are changeable and remain uncertain. A long-term relationship is very difficult to maintain after the goals of team member agents change.

Both one-shot team formation and long-term team-formation mechanisms have some weaknesses. One-shot team formation may bring high communication and computation overhead to a MAS. However, long-term team formation strategies are not suitable for the dynamic features of open environments and the selfish features of self-interested agents.

In this section, we introduce and compare the features of the one-shot and long-term team-formation mechanisms. In addition, to cover some shortcomings of one-shot and long-term team formations, a flexible team-formation mechanism that enables both cooperative and self-interested agents to flexibly choose team membership and duration is proposed. Factors such as historical agent performance, task requirements and resource constraints are considered in the mechanism. Especially for open environments, flexible team formation and member selection mechanisms are more suitable for agent applications. This flexible team-formation mechanism enables more dynamic and reasonable cooperation between agents and reduces unnecessary overhead and utility conflicts brought about by team formation. Due to the high uncertainty inherent in most open environments, analysis and evaluation of dynamic factors is not very straightforward. More specifically, a fixed standard for agent evaluations does not exist (e.g. how good an agent's performance is). Regarding this point, fuzzy rules are used in our flexible team-formation mechanism to evaluate the performance and importance of agents. This will enable an agent to dynamically select cooperation durations and objectives based on the results of fuzzy evaluations, and to choose cooperation mechanisms more flexibly.

3.1 Agent Cooperation in Agent Teams: The Scenario

Various MAS applications may have different system structures. In this chapter, an MAS environment is set up to demonstrate and analyse team formation and member selection mechanisms. Hence, the system structure is set up toward assisting agent communication and task allocation. Some simplifying assumptions and definitions, which can avoid adding to the scheduling and task decomposition problems, are also made, and only elementary agents and task models are included in the MAS. However, these models are sufficiently generic to be practical and applicable to a wide range of real-world applications.

Figure 1 shows the general structure of team organisation. To simplify the problem, we assume that all agents are aiming to achieve rewards through accomplishing tasks sent by outside users. New tasks are published on the system *Task Board*, and will be removed from the *Task Board* after being taken by an agent or agent team.

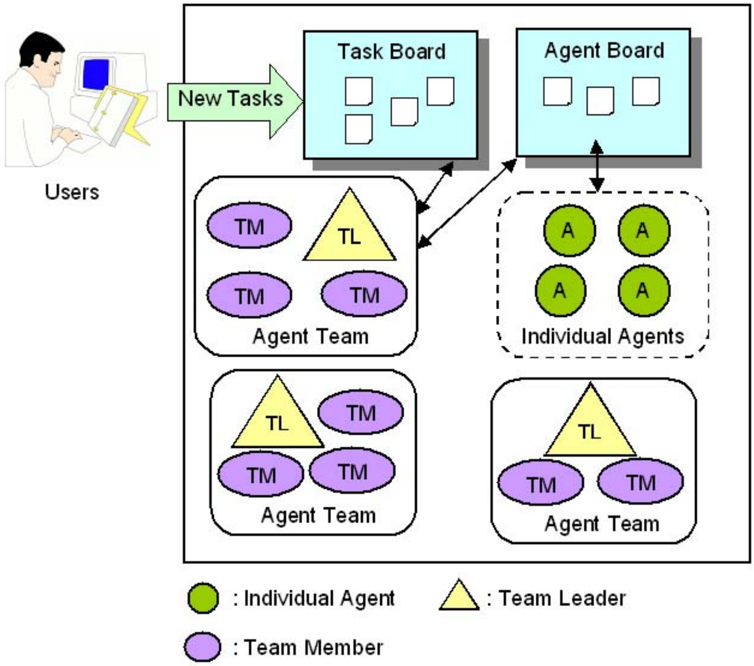


Fig. 1 The System Architecture

Published tasks are accessible to all individual agents and agent teams within the system. The number of agents in the system can be dynamic; agents can enter and leave the system at will. However, agents need to publish and remove their registration information on the system *Agent Board* before they so enter (leave). The registration information records the skills and status of an agent (see Definition 4).

Agent abilities are limited. To perform tasks beyond their individual ability, an agent needs to collaborate with other agents through joining or forming a team. Each agent team is composed of one (and only one) team leader and several team members. After an agent joins an agent team, it can receive payments from the agent team. At the same time it needs to work for the agent team for a certain period. The payment and serving term are described in the contract (see Definition 5) between the team member and the team leader.

Before presenting the team-formation mechanism, some important definitions and assumptions are given.

Definition 3. A *task* is defined formally as $t_i = \langle w_i, R_i' \rangle$, where w_i is the reward gained by an agent/agent team if task t_i is accomplished by that agent/agent team; R_i is the set of resources or skills, which are possessed by agents, required by task t_i. A task can only be assigned to one agent or agent team.

Definition 4. An *agent* is formally defined as $a_i = \langle g_i, R_i, s_i \rangle$, where g_i is a set of individual goals of agent a_i; R_i is the skills and resources possessed by agent

Table 1 Status of An Agent

s_i **value**	**Status of agent** a_i
(0, 0, 0)	Performing no task; has no agent team.
(1, 0, 0)	Performing a task; has no agent team.
(0, 1, 0)	Has a one-shot contract as a team member; performing no task currently.
(1, 1, 0)	Has a one-shot contract as a team member; performing a task currently.
(0, 1, t)	Team member of an agent team for period t; performing no task currently.
(1, 1, t)	Team member of an agent team for period t, performing a task currently.
(0, 2, 0)	The team leader of an agent team; performing no task currently. (It is assumed that the team leader cannot quit from its agent team and let t value of a team leader equal to 0.)
(1, 2, 0)	The team leader of an agent team; performing a task currently.

a_i; and s_i is the status of a_i, where $s_i = (v_a, v_p, t)$. s_i represents whether agent a_i is performing a task and participating in an agent team. The meanings of different s_i values are listed in Table 1. The names and meanings of v_a, v_p and t are as follows:

Availability v_a: represents whether an agent is performing a task. $v_a = 0$ when the agent has no task (available); $v_a = 1$ when the agent is performing a task (not available);
Position Parameter v_p: represents whether an agent is an individual agent, team leader or team member. $v_p = 0$ when the agent is an individual agent; $v_p = 1$ when the agent is a team member; $v_p = 2$ when the agent is a team leader.
Contract Completion Time t: t is the contract completion time of an agent (also see Definition 5).

Definition 5. A *contract* c_{ij} is an agreement between team leader a_i and team member a_j. It can be defined as $c_{ij} = \langle t_{ij}, p_{ij}, S_{ij} \rangle$, where t_{ij} is the contract completion time; p_{ij} is the penalty that the team leader or team member has to pay (to the other parties of the contract) if it breaks the contract and terminates the cooperation relationship before t_{ij}; S_{ij} is a set of payments that a_j can gain through serving the agent team. S_{ij} can be described as a tuple $\langle sc_{ij}, sd_{ij} \rangle$. For contracts between the team leader and team members of a one-shot team, t_{ij}, p_{ij}, and sd_{ij} are equal to 0. sc_{ij} is the payment that a_j can gain for each task completed by the agent team when a_j directly participates in the task. sd_{ij} is the dividend (or reward) that a_j can share for each task completed by the agent team, when a_j does not actually participate in that task.

Definition 6. An *agent team* is a set of agents. It can be formally defined as $AT_i = \langle MS_i, TR_i \rangle$, where MS_i is the set of agents (including the team leader) that are currently team members of AT_i; TR_i is the total resources of the entire agent team. Here it is assumed that $TR_i = \sum_{j|a_j \in MS_i}(_j) + r_i$, where R_i and R_j are resources possessed by the team leader and team members, respectively. In other words, the capability of an agent team is the sum of its team members' capabilities and the team leader's capability. We further define $\forall i \neq j : MS_i \cap MS_j = \varnothing$, which means that an agent can only participate in a single agent team.

Definition 7. A *Contributor Set* $CS_{ij}(CS_{ij} \subset MS_i)$ of agent team AT_i is the set of agents that participate in performing task t_j, where t_j is a task of AT_i. For a one-shot team, the Contributor Set is equal to MS_i of the team (also refer to Definition 6).

Definition 8. For agent team AT_i, a *Member Contribution* mc_{ijk} is the contribution of agent a_k, where $a_k \in CS_{ij}$, in performing task t_j $(t_i = \langle w, R'_i \rangle)$. mc_{ijk} equals w/N, where N is the size of Contributor Set and w is the task reward.

3.2 One-Shot and Long-Term Team-Formation Mechanisms

After presenting the system architecture and some important definitions, concepts and comparisons of the one-shot and long-term team-formation mechanisms are presented in this subsection.

3.2.1 One-Shot Team-Formation Mechanism

One-shot team-formation mechanisms are widely applied in many MAS applications. In this mechanism, agents do not initially have a team. When a task t_i is published in the *Task Board*, agents start to bid on the new task. The system facilitator will choose (or randomly select) a bidder to assign the task. After the agent successfully bids for the task, it becomes a team leader and starts to look for collaborators according to the task requirement R'_i. Finally, the agent team will disband after the task (t_i) is accomplished.

Generally, the one-shot team strategy includes the following processes. (Here, it is assumed that the agents cannot achieve the task individually.)

1. The system facilitator of the MAS publishes a new task $t_i = \langle w_i, R'_i \rangle$ on the *Task Board*, where w_i and R'_i are the reward and required resources of the task;
2. Agents, whose $g < w_i$ and s=(0, 0, 0) bid on t_i;
3. The system facilitator awards t_i to agent $a_j(a_j =< g_j, R_j, s_j >)$. At the same time, a_j becomes the team leader of agent team AT_j and modifies its s_j to (0, 2, 0). At this movement, $TR_j = R_j$;
4. a_j searches the Agent Board to look for agents with status (0, 0, 0), which can provide the lacking resources R, where $R \subseteq (R'_i - R'_i \cap TR_j)$;
5. a_j finds a required agent a_p, where $R_p \subseteq (R'_i - R'_i \cap TR_j)$;
6. a_j sends a contract c_{jp} to a_p, where $sc_{jp} \leq (w_i - g_j) \cdot sizeOf(R_p)/sizeOf(R'_i - R_i)$;
7. a_p accepts c_{jp} if $sc_{jp} \geq g_p$ or rejects c_{jp} if $sc_jp \leq g_p$;
8. If c_{jp} is accepted by a_p, $TR_j = TR_j \cup R_p$, and a_p modifies its status to (0, 1, 0);
9. Goes to Process (4) until $TR_j = R'_i$;

10. AT_j starts to perform t_i; the team leader and the team members of AT_j modifies/modify its/their statuses to (1, 1, 0) and (1, 2, 0), respectively;
11. AT_j accomplishes t_i; agents of AT_j modify their statuses to (0, 0, 0) and are released from the team.

3.2.2 Long-Term Team-Formation Mechanism

In the long-term team-formation mechanism, the agent team will not be dissolved after performing tasks. On the contrary, the team leader gives the team members some payment to maintain the cooperative relationship, even if the team member does not contribute to accomplishing the task.

The long-term team strategy normally includes the following processes [30]:

1. Team leader a_i finds several free agents, whose status values are (0, 0, 0), from the *Agent Board* and sends them contracts in order to form a team with them. Agents modify their status to (0, 1, t_{ij}) if they accept the contracts. In this case, agent team AT_i is formed successfully;
2. Team leader a_i searches the *Task Board* for a suitable task and bids on task $t_k(t_k = \langle w_k, R'_k \rangle)$, where $R'_k \subseteq TR_i$ and $w_k \geq \sum_{j|a_j \in MS_i}(S_{ij} + g_i)$ (also refer to Definitions 3 through 6).
3. If t_k is successfully bid by team leader a_i, a_i assigns t_k to team member $a_p, a_q ... a_n$, where $R_p \cup R_q, ..., \cup R_n$ is the minimum set that satisfies $R'_k \subseteq R_p \cup R_q, ..., \cup R_n$. At the same time, $a_p, a_q, ..., a_n$ modify their status to $(1, 1, t_{ip})$, $(1, 1, t_{iq}), ..., (1, 1, t_{in})$. Also, for this task performance, the Contributor Set CS_{ik} (refer to Definition 7) should be $\{a_p, a_q, ..., a_n\}$;
4. $a_p, a_q, ..., a_n$ modify their status to $(0, 1, t_{ip}), (0, 1, t_{iq}), ..., (0, 1, t_{in})$ after t_k is accomplished;
5. team leader a_i awards team member a_m ($a_m \in AT_i$) with $(sc_{im} + sd_{im})$ if $a_m \in CS_{ik}$, or sd_{im} if a_m is not in CS_{ik};

In addition, if the team leader a_i or team member a_p wants to terminate the contract before the contract completion time t_{ip}, they may process the following two steps:

1. a_i/a_p terminates c_{ip} with a_p/a_i, and pays p_{ip} to a_p/a_i;
2. a_p is released from AT_i, and its status modified to (0, 0, 0).

3.2.3 Advantages and Disadvantages of Long-Term and One-Shot Team-Formation Mechanisms

One-shot teams are suitable for dynamic MAS application domains. They always maintain loosely-coupled relationships among agents by default. However, agents in dynamic applications may also need to keep stable organisations in some situations. For example, the tasks may have some similarity, and their requirements might be similar (which means they may just need similar agent teams). In this case, frequent grouping and regrouping is not necessary, since each such grouping consumes some

Table 2 Features of One-Shot Teams and Long-Term Teams

	One-Shot Teams	**Long-Term Teams**
Communication Overhead	High	Low
Suitable Domains	Highly dynamic environments	Stable environments
Suitable MASs	Self-interested MAS	Cooperative MAS
Relationships among Team Members	Loosely coupled	Tightly coupled

system resources. In contrast with one-shot teams, long-term teams can greatly reduce the system overhead caused by grouping and regrouping. However, most current long-term team formation strategies cannot figure out when agents should form long-term teams, which agents should be included, and how long the relationships should be maintained. For self-interested MAS applications, keeping unnecessary long-term cooperative relationships could be dangerous and harmful for the overall system performance. Features of one-shot and long-term teams are summarised and compared in Table 2.

3.3 Flexible Team-Formation Mechanism

From the description of short-term and long-term team formation in the previous section, it can be seen that both long-term and one-shot teams have some advantages and disadvantages. One-shot teams are suitable for dynamic tasks, where the requirements of various new tasks are totally different. By contrast, long-term teams possess advantages when tasks are 'stable' or similar. For most self-interested agents, the team duration should not be fixed. Taking human society as an example, a company may sign different contracts (with different durations and conditions) with different employees. According to the performance of employees and changes in the job market, the company will typically want to make changes to these contracts in the future. For a MAS, it is also necessary to have a flexible team-formation mechanism which can enable team leaders to choose different cooperation durations with agents, according to the changing trends of task-requirements and agent performance. In this section, a flexible team-formation mechanism is introduced. In this mechanism, agent value and availability are evaluated. Team leaders will then determine the required members and choose proper cooperation durations and cost according to these evaluation results.

3.3.1 Team Member Performance Evaluation

In general, agents that always contribute to performing tasks and can bring more benefits to the team are the most valuable members of an agent team. These agents should be kept on the team for a long time. By contrast, an agent team should not include agents that bring little contribution to the team. In this mechanism, two factors, namely *Utilisation Ratio* (*ur*) and *Contribution Ratio* (*cr*), are used to evaluate the value of a team member.

Definition 9. *Utilisation Ratio* ur_{Mk} $(ur_{Mk} \in [0,1])$ is the frequency with which a team member a_k has participated in the most recent M tasks of the agent team AT_i. It can be calculated using Equation 1. The value of the parameter M is chosen by team leaders or assigned by users. Team leaders can also adjust M values according to environmental situations and team performance.

$$ur_{Mk} = \sum_{j=1}^{M} \frac{1}{M} \quad (\, k|a_k \in CS_{ij}) \tag{1}$$

Definition 10. *Contribution Ratio* cr_{Mk} $(cr_{Mk} \in [0,1])$ is the ratio that team member a_k has contributed to the agent team AT_i in the most recent M tasks. It can be calculated using Equation 2 (also refer to Definition 8).

$$cr_{Mk} = \frac{\sum_{j=1}^{M} mc_{ijk} \quad (\, k|a_k \in CS_{ij})}{\sum_{j=1}^{M} w_j} \tag{2}$$

The following example shows how to evaluate team members through Utilisation Ratio and Contribution Ratio. Suppose $t_1 =< 40, R'_1 >, t2 =< 50, R'_2 >$ and $t_3 =< 60, R'_3 >$ are the three most recent tasks accomplished by agent team AT_i. a_p, a_q, a_r and a_s are the team members of AT_i. Team members that participate in the three tasks are $\{a_p, a_q\}, \{a_p, a_r\}$ and $\{a_p, a_q\}$, respectively. According to Equations 1 and 2 , the Utilisation Ratio and Contribution Ratio values of a_p, a_q, a_r and a_s are:

$$a_p\colon\ ur_{3p} = 1, \qquad cr_{3p} = \frac{(40/2+50/2+60/3)}{(40+50+60)} = 0.5$$

$$a_q\colon\ ur_{3q} = 0.67, \ cr_{3q} = \frac{(40/2+60/3)}{(40+50+60)} = 0.33$$

$$a_r\colon\ ur_{3r} = 0.33, \ cr_{3r} = \frac{50/2}{(40+50+60)} = 0.17$$

$$a_s\colon\ ur_{3s} = 0, \qquad cr_{3p} = 0$$

Comparing Utilisation Ratio and Contribution Ratio values of the four team members of AT_i, it can be seen that a_p is the most important member of AT_i, since it frequently participated in recent tasks and contributed the most benefit to the team. On the other hand, a_s did not participate in recent tasks and contributes nothing to AT_i.

3.3.2 System Agent Resource Evaluation

With Utilisation Ratio and Contribution Ratio, a team leader can evaluate contributions of team members. However, to make reasonable contracts with team members, a team leader also needs to evaluate whether it is easy to find similar agents (which possess similar resources and skills) in the MAS. In this mechanism, Agent

Resource Availability is the parameter defined to evaluate agent resource availability in the MAS.

Definition 11. Agent Resource Availability ara_k: ara_k is the ratio of available agents (which do not have a team/task) that possess the same or more resources than team member a_k. It can be calculated using Equation 3 (Note: N_{av} here is the available agent number of the MAS).

$$ara_k = \sum_{s_i=(0,0,0)}^{R_k \subseteq R_i} \frac{1}{N_{av}} \tag{3}$$

For example, suppose that a_k is a team member of AT_i. Currently, there are ten out of twenty available agents in the MAS, which possess the same or more resources than a_k. Hence, the Agent Resource Availability value of team member a_k is: $ara_k = 0.5$.

3.3.3 Flexible Member Selection Using Fuzzy Rules

According to the values of Utilisation Ratio, Contribution Ratio and Agent Resource Availability, in this mechanism, team leaders use a fuzzy method to determine cooperation durations and cost with their team members.
Input and Output Parameters:

In the fuzzy method, Utilisation Ratio, Contribution Ratio and Agent Resource Availability are input parameters. The output parameters are Contract Term ct and Commission Amount ca. These parameters are defined in Definitions 12 and 13.

Definition 12. *Contract Term* ct_k is the parameter which denotes the duration that agent a_k should be kept in the agent team. It is an output parameter that needs to be identified through the fuzzy method. The working range of *Contract Term* is [0, *MAXTERM*]. *MAXTERM*, which is a constant that is defined in the MAS, and denotes the maximum time period that an agent can be kept in an agent team.

Definition 13. *Commission Amount* ca_k is the parameter that denotes the maximum commission that the agent team should pay to agent a_k in order to keep it in the team. It is an output parameter that needs to be identified through the fuzzy method. The working range of Commission Amount is [0, *MAXPAY*], where the parameter *MAXPAY* is decided by the team leader. *MAXPAY* denotes the maximum payment that an agent team can afford to keep a single agent as a team member.

Membership Functions for Input Parameters:

For Utilisation Ratio, the following four linguistic states [17] are selected and expressed by appropriate fuzzy sets: *Never* (N), *Seldom* (S), *Medium*, (M) and *Frequent* (F). Another input parameter Contribution Ratio also has four linguistic states, these being *None* (N), *Little* (L), *Medium* (M) and *Huge* (H). The trapezoidal [17] fuzzy membership function is adopted here to define fuzzy memberships of these

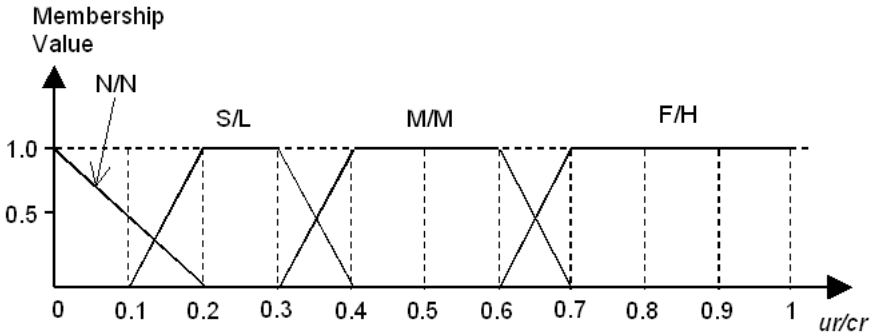


Fig. 2 Fuzzy Membership Function for *ur/cr*

four fuzzy sets. The membership functions for these four fuzzy sets are defined in Equations 4 through 7, respectively. They are also depicted in Figure 2.

$$F_{Never}(x)/F_{None}(x) = \begin{cases} 1-5x & x \in [0,0.2] \\ 0 & x \notin [0,0.2] \end{cases} \tag{4}$$

$$F_{Seldom}(x)/F_{Little}(x) = \begin{cases} 10x-1 & x \in [0.1,0.2] \\ 1 & x \in (0.2,0.3) \\ 4-10x & x \in [0.3,0.4] \\ 0 & x \notin [0.1,0.4] \end{cases} \tag{5}$$

$$F_{Medium}(x) = \begin{cases} 10x-3 & x \in [0.3,0.4] \\ 1 & x \in (0.4,0.6) \\ 7-10x & x \in [0.6,0.7] \\ 0 & x \notin [0.3,0.7] \end{cases} \tag{6}$$

$$F_{Frequent}(x)/F_{Huge}(x) = \begin{cases} 10x-6 & x \in [0.6,0.7] \\ 1 & x \in (0.7,1] \\ 0 & x \notin [0.6,1] \end{cases} \tag{7}$$

For *ara*, three linguistic states are selected, namely *Rare* (R), *Some* (S), and *Many* (M). The membership functions for *ara* are defined in Equations 8 through 10, and depicted in Figure 3.

$$F_{Rare}(x) = \begin{cases} 1-4x & x \in [0,0.4] \\ 0 & x \notin [0,0.4] \end{cases} \tag{8}$$

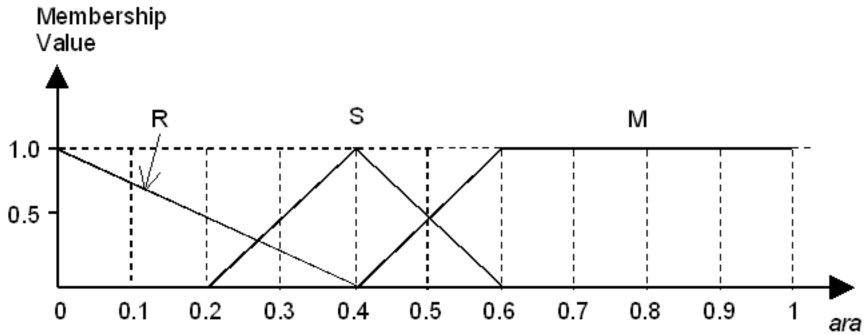


Fig. 3 Fuzzy Membership Function for *ara*

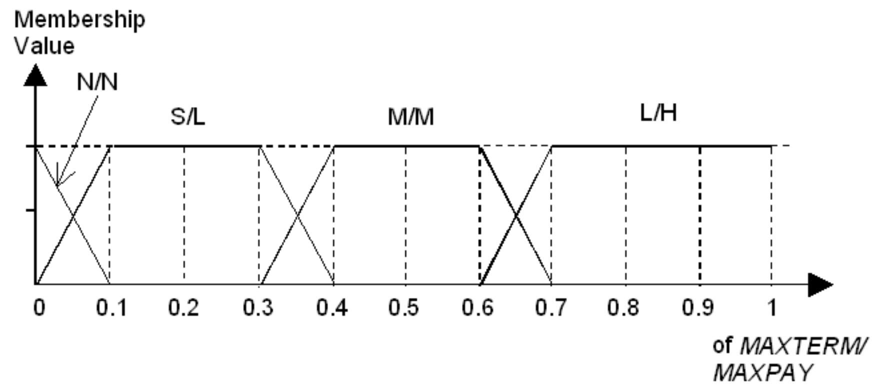


Fig. 4 Fuzzy Membership Function for *ct/cl*

$$F_{Some}(x) = \begin{cases} 5x-1 & x \in [0.2, 0.4] \\ 3-5x & x \in (0.4, 0.6] \\ 0 & x \notin [0.2, 0.6] \end{cases} \tag{9}$$

$$F_{Many}(x) = \begin{cases} 5x-2 & x \in [0.4, 0.6] \\ 1 & x \in (0.6, 1] \\ 0 & x \notin [0.4, 1] \end{cases} \tag{10}$$

Membership Functions for Output Parameters:

There are two output parameters – *Contract Term* (*ct*) and *Commission Level* (*cl*) – in the fuzzy method. For *ct*, the following four linguistic states are selected: *Long* (L), *Medium* (M), *Short* (S) and *No* (N). For *cl*, *High* (H), *Medium* (M), *Low* (L) and *No* (N) are chosen as the four linguistic states. Fuzzy membership functions of these fuzzy sets are defined in Equations 11 through 14, and described in Figure 4.

$$F_{No}(x) = \begin{cases} 1-10x & x \in [0,0.1] \\ 0 & x \notin [0,0.1] \end{cases} \tag{11}$$

$$F_{Short}(x)/F_{Low}(x) = \begin{cases} 10x & x \in [0,0.1] \\ 1 & x \in (0.1,0.3) \\ 4-10x & x \in [0.3,0.4] \\ 0 & x \notin [0,0.4] \end{cases} \tag{12}$$

$$F_{Medium}(x) = \begin{cases} 10x-3 & x \in [0.3,0.4] \\ 1 & x \in (0.4,0.6) \\ 4-10x & x \in [0.6,0.7] \\ 0 & x \notin [0.3,0.7] \end{cases} \tag{13}$$

$$F_{Long}(x)/F_{High}(x) = \begin{cases} 10x-6 & x \in [0.6,0.7] \\ 1 & x \in (0.7,1] \\ 0 & x \notin [0.6,1] \end{cases} \tag{14}$$

Fuzzy Rule Base:

A fuzzy rule base is a matrix of combinations of each of the input linguistic parameters and their corresponding output parameters. The rule base in this mechanism is described in Table 3.

Table 3 Fuzzy Rule Base Matrix

Agent Resource Availability		***R***	***S***	***M***
Utilisation Ratio	***Contribution Ratio***	**Output Parameters: *ct*, *cl***		
N	***N***	*ct=N, cl=N*	*ct=N, cl=N*	*ct=N, cl=N*
N	***L***	*ct=M, cl=L*	*ct=N, cl=N*	*ct=N, cl=N*
N	***M***	n/a	n/a	n/a
N	***H***	n/a	n/a	n/a
S	***N***	*ct=M, cl=L*	*ct=N, cl=N*	*ct=N, cl=N*
S	***L***	*ct=L, cl=L*	*ct=S, cl=L*	*ct=N, cl=N*
S	***M***	*ct=L, cl=L*	*ct=M, cl=M*	*ct=S, cl=M*
S	***H***	*ct=L, cl=M*	*ct=S, cl=M*	*ct=N, cl=M*
M	***N***	n/a	n/a	n/a
M	***L***	*ct=L, cl=M*	*ct=M, cl=L*	*ct=S, cl=L*
M	***M***	*ct=L, cl=M*	*ct=M, cl=M*	*ct=M, cl=L*
M	***H***	*ct=L, cl=H*	*ct=L, cl=M*	*ct=M, cl=M*
F	***N***	n/a	n/a	n/a
F	***L***	*ct=L, cl=M*	*ct=M, cl=M*	*ct=L, cl=L*
F	***M***	*ct=L, cl=H*	*ct=L, cl=M*	*ct=L, cl=L*
F	***H***	*ct=L, cl=H*	*ct=L, cl=H*	*ct=L, cl=M*

Determination of Output Membership Values and Defuzzification

Each entry of the rule base is a rule, which is defined by ANDing two linguistic input parameters to produce an output combination, in the form of: $IF(F(ur) = \alpha$ AND $F(cr) = \beta$ AND $F(ara) = \gamma)$ THEN $(F(ct) = \delta)$ AND $F(cl) = eta)$, where $\alpha \in \{$*Never, Seldom, Medium, Frequent*$\}$, $\beta \in \{$*None, Little, Medium, Large*$\}$, $\gamma \in \{$*Rare, Some, Many*$\}$, $\delta \in \{$*Long, Medium, Short, No*$\}$, and $\eta \in \{$*High, Medium, Low, No*$\}$. In this mechanism, the *AND/MIN* operator is used to combine the membership values, i.e. the weakest membership determines the degree of membership in the intersection of fuzzy sets [8] [17]. Hence, the output membership value $\mu_{\delta/\eta}(v)$ can be calculated using Equation 15.

$$\mu_{\delta/\eta}(v) = MIN(\mu_{\alpha}(ur), \mu_{\beta}(cr), \mu_{\gamma}(ara)) \tag{15}$$

With regard to output membership, the output values can be determined by tracing the membership values for each rule back through the output membership functions. Finally, the *centroid* defuzzification method [8] [17] is used to determine the output value. In *centroid* defuzzification, the output value is calculated using Equation 16, where membership of v_i is represented as $\mu(v_i)$, and k is the number of fuzzy rules which are activated.

$$DF = \frac{\sum_{i=1}^{k}(v_i \cdot \mu(v_i))}{\sum_{i=1}^{k} \mu(v_i)} \tag{16}$$

3.4 Experiments

To analyse the performance of the flexible team-formation mechanism, some experiments are conducted to compare it with the one-shot and long-term team-formation mechanisms. The experimental environment is set up to simulate the scenario introduced in Subsection 3.1. Each agent possesses one (or more) kind of resource(s), and needs to contribute its resource(s) to achieve rewards through accomplishing tasks in the system. However in most cases an agent cannot accomplish a task due to its limited resource(s). Hence, agents need to cooperate with others in order to realise their goals. This experiment simulates some real world applications. For example, in a Web service system [24], each peer can only provide a limited number of services (i.e. possesses limited resources). To execute a complex task, we need to aggregate or combine small services in different peers into larger services (i.e. form a team to perform the task).

In the experiment, a set of tasks is sent to the agents, and they perform these tasks using one-shot, long-term and flexible team-formation mechanisms, respectively. In order to avoid agent teams including too many agents for too long a time (especially for long-term teams), we set a maximum team size to limit the number of long-term team members. In this experiment, the maximum team size equals five, which means an agent team can at most keep five long-term members. Two factors are compared in the experiment, namely *Agent Searching Time* and *Reward Distribution Situation*.

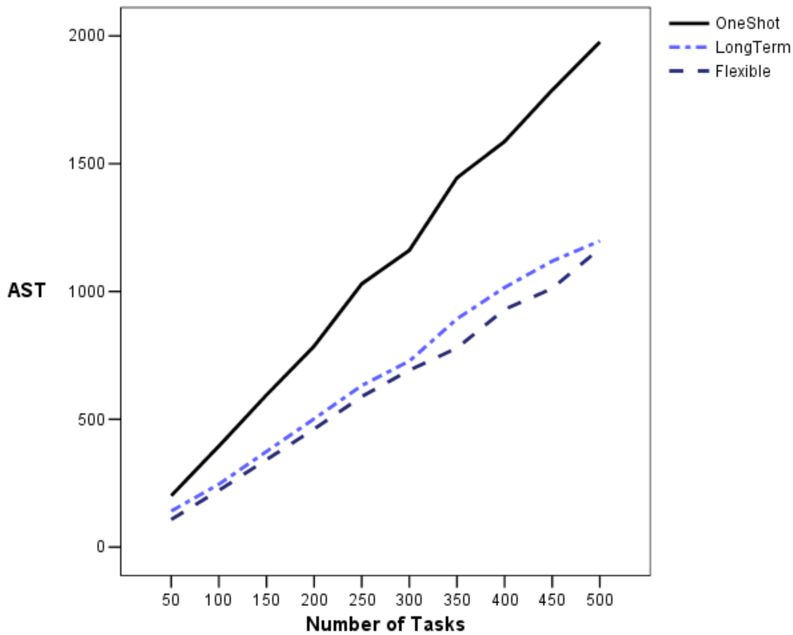


Fig. 5 Agent Search Comparison (no. of searches vs. no. of tasks)

Agent Searching Time represents the time that a team leader needs to search for required agents from the agent board to accomplish the tasks. In general, the higher the *Agent Searching Time*, the more communication cost the team leader needs to spend on searching agents.

According to the experimental result, it can be seen that the *Agent Searching Time* of one-shot team formation is much higher than both long-term and flexible team formation (See Figure 5). This is because team leaders in one-shot teams need to keep searching suitable team members for each task and disband them after a task is accomplished. With long-term and flexible team formation, the whole team (or part thereof) is retained after a task is completed. Thus these two latter strategies will have less communication overhead. The experimental result shows that long-term teams have higher *Agent Searching Time* than flexible teams. This is because, in the experiment, a long-term team can at most keep a limited number of members for a long period. Hence, after a team accomplishes several tasks, the number of long-term members will reach the limit, and the team will start to search and disband new members in subsequent tasks. The result shows that the *Agent Searching Time* of using flexible team formation is the lowest, which means it has the lowest communication overhead among the three mechanisms.

Reward Distribution Situation is the second comparison factor. It represents the rationality of agent team organisation. Without considering communication overhead, a one-shot team has an ideal organisational structure because all its team

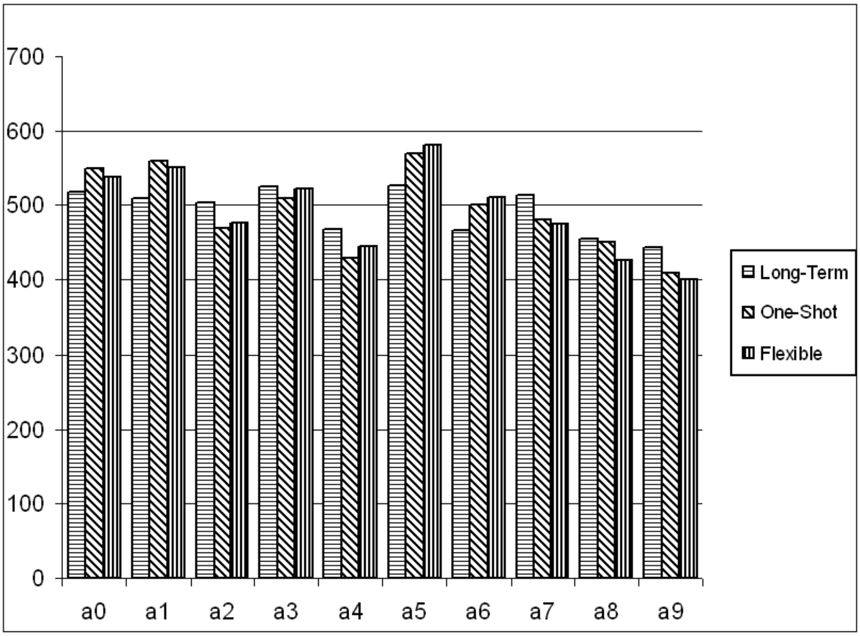


Fig. 6 Reward Distribution Situation (no. of reward units per agent)

members contribute to task executions. Hence the *Reward Distribution Situation* of one-shot teams can be considered as the benchmark for team organisation rationality. Throughout this experiment, it can be seen that the *Reward Distribution Situation* of flexible teams is closer to one-shot teams than long-term teams (See Figure 6). Therefore, flexible teams have more reasonable organisations than long-term teams.

From the experimental results, it can be seen that the flexible team-formation mechanism is more suitable for self-interested agents and open environments. In cooperative domains, agents do not care whether the reward is distributed rationally, the most important thing that cooperative agents consider is the overall benefit to the team. However, self-interested agents do consider rationality of reward distribution, and do not want to keep "less valuable" members in the team for a long period. The flexible mechanism can enable agent teams to keep valuable team members according to their performance and changing environments. Furthermore, agent teams can adjust their long-term member selection criteria through modifying the member evaluation parameters. This feature can make team formation more flexible and suitable for open environments. Therefore, compared with one-shot and long-term team-formation mechanisms, the flexible team-formation mechanism can enable self-interested agents to form more reasonable teams in an open environment with less communication overhead.

3.5 Summary

As a social entity, an intelligent agent needs to cooperate with others in most multi-agent environments. At the same time, unreasonable team-formation mechanisms could prevent agents from achieving local benefits, or lead to unnecessary system overhead. Focusing on challenges inherent in dynamic application domains, many researchers have suggested using long-term or one-shot team-formation mechanisms in MASs. However, both of these mechanisms have some advantages and disadvantages, as discussed earlier. A flexible team-formation mechanism can avoid some of the limitations of the one-shot and long-term team-formation mechanisms. It can enable agents to automatically evaluate the performance of other agents in the system, and select team members with reasonable terms and costs according to the evaluation result. In flexible team-formation, factors related to agent performance and task requirements are considered as evaluation factors. Through evaluating these factors, team compositions are more reasonable and can avoid some potential benefit conflicts among team members.

4 Collaborative Problem Solving through Agent Competition

In some application domains, agent competition can also be involved in collaborative problems. Suppose a set of autonomous agents has a global goal it wants to achieve, where this goal is too complex to be achieved by any single agent. Therefore, the global goal must be divided into several local goals and distributed to agents by considering their individual ability, requirement, restriction etc. Now each agent wants to minimize its costs, that is, prefers to do as little as possible. Therefore, even though the agents have a common goal, there is actually a conflict of interest here. Agents may argue and compete with each other in order to maximize their individual benefits and also ensure that the global goal be achieved in a timely manner. This kind of competition within a collaborative problem may pertain in applications such as resource allocation, task distribution, emergencies etc. Agent negotiation can be employed to solve competition problems.

4.1 Traditional Agent Negotiation

Motivations and aims determine agent behavior in negotiation. Therefore, it is necessary to discuss the kinds of agent behavior which can take place during negotiations. In general, agents may compete or cooperate with each other in order to reach their own goals or a common goal within a MAS. Final agreements about how to compete or cooperate are achieved through negotiation. Therefore, negotiations can be classified into *competitive* and *cooperative* according to the behaviors of its participants. In a *competitive* negotiation, participants perform the role of challengers, while in a *cooperative* negotiation, participants act as cooperators. However, both kinds of negotiation contain the following four components in general [33] :

1. The negotiation protocol,
2. The negotiation strategies,

3. The information state of agents,
4. The negotiation equilibrium.

The negotiation protocol specifies the rules of engagement in agent negotiation. It defines what kinds of (i) interaction between agents can be taken in different circumstances; (ii) sequences are allowed and (iii) deals can be made in the negotiation. For example, Rubinstein's alternating offers protocol is a very commonly used negotiation protocol. In this protocol, one of the negotiation participants makes an offer, then the other responds by either accepting the offer, rejecting it, or opting out of the negotiation. The negotiation will be finished only when all negotiation participants accept an offer, or one or more negotiation participants opt out. In general, agents should make an agreement on the negotiation protocol before the negotiation proper starts. The negotiation protocol will be designed differently by considering the following factors: (a) numbers of negotiation participants (e.g. sellers and buyers), (b) numbers of negotiation issues (e.g. a car's price, color, model and etc.), and (c) negotiation environment (buyers' market or sellers' market).

The negotiation strategy specifies the sequence of actions that the negotiation participants plan to make during the negotiation. In competition problem negotiation, agents try to maximize their own local interests during the negotiation, and also have to ensure the global goal of the negotiation. Therefore, agents may employ different negotiation strategies by considering self and/or other information. For example, an agent could bargain very hard throughout the negotiation in order to maximize its benefit or give some kind of concession under time restrictions. Also, it should be clear that a strategy which performs well with certain protocols may not necessarily do so with others. Therefore, both the negotiation scenario and protocol in use should be considered when the negotiation participant chooses a negotiation strategy.

The agents' information state describes information about the negotiation, which can be classified as 'private' and 'public' [9]. Private information describes an agent's self situation, such as the negotiation strategy, which is only possessed by that particular agent. Unless the negotiation participant agrees to share its private information with others, it is not reachable by other negotiation participants. Public information describes the negotiation environment, such as the number of negotiation participants, number of negotiation issues, negotiation protocols etc. This public information is available to all negotiation participants. In the negotiation, if all negotiation participants would like to share all their private information, then the negotiation is referred to as 'negotiation with complete information'. Otherwise, if the negotiation participant does not want to share their private information, then the negotiation is termed 'negotiation with incomplete information'. An agent's information state will impact the agent's choice of negotiation strategy.

When agents choose negotiation protocols and negotiation strategies, agents create negotiation mechanisms. During the negotiation, the negotiation mechanism must be stable, i.e. a strategy profile must constitute an equilibrium. The Nash equilibrium [26] is a commonly used concept. Two strategies are in Nash equilibrium if each negotiation participant's strategy is the best response to its opponent's strategy. The equilibrium is a very important and necessary condition for negotiation system stability. For different negotiation protocols, the equilibrium strategy may differ. However, it is required that each negotiation participant should select an equilibrium strategy in the negotiation.

In this subsection, we provide an example of negotiation between two agents. In our example, the negotiation is performed between two agents, i.e. the 'buyer' agent and the 'seller' agent. Both agents are bargaining over the price, therefore it is a single-issue negotiation. In the following, we will show the four components in our example negotiation and introduce how the negotiation is processed.

The negotiation protocol. We simply adopt the basic alternating offers protocol [28]. Let b denote the buyer agent, and s the seller agent. The negotiation starts when the first offer is made by an agent (b or s). The agent who makes the initial offer is selected randomly at the beginning of the negotiation. When an agent receives an offer from its opponent, it will evaluate it. According to this evaluation, the agent will take one of the following actions: (i) *Accept*: when the value of the offer received from the opponent is equal to or greater than the value of the counter-offer it is going to send in the next negotiation cycle. Once the agent accepts this offer, the negotiation ends successfully in an agreement; (ii) *Reject*: when the value of the offer received from the opponent is less than the value of the counter-offer it is going to send in the next negotiation cycle. Once the agent rejects this offer, providing the negotiation deadline has not been reached, the agent sends out a counter-offer to its opponent and the negotiation proceeds to the next cycle; (iii) *Quit*: when the negotiation deadline falls due and no agreement has been reached, then the agent has to quit and the negotiation fails.

The negotiation strategies. In our example, two agents are bargaining over price, therefore each agent should have some idea about its acceptability. Let $[IP^a, RP^a]$ denote the range of price values which are acceptable to agent a, where $a \in \{b, s\}$. IP^a denotes the initial price and RP^a the reserve price of agent a. In general, when $a = b$, $IP^b \leq RP^b$, and when $a = s$ $IP^b \geq RP^b$. Let $\hat{a}$ denote agent a's opponent, where $\hat{a} \in \{b, s\}$. Then the offer made by agent a to agent $\hat{a}$ at time t ($0 \leq t \leq \tau^a$), where τ^a is the deadline for agent a, is modeled as a function Φ^a depending on time as follows:

$$p^t_{a \to \hat{a}} = \begin{cases} IP^a + \Phi^a(t)(RP^a - IP^a) & a = b \\ RP^a + (1 - \Phi^a(t))(IP^a - RP^a) & a = s \end{cases} \tag{17}$$

where function $\Phi^a(t)$ ($0 \leq \Phi^a(t) \leq 1$) is called the negotiation decision function (NDF) [12]. The common way to define $\Phi^a(t)$ is:

$$\Phi^a(t) = k^a + (1 - k^a)(\frac{t}{\tau^a})^{1/\lambda} \tag{18}$$

where k^a ($0 \leq k^a \leq 1$) is the parameter which controls the initial offer. For example, when $k^a = 0$, the initial offer is IP^a, and when $k^a = 1$, the initial offer is RP^a; λ is the parameter which controls the agent behavior. Depending on the value of λ, three extreme cases show different patterns of behavior for the agent

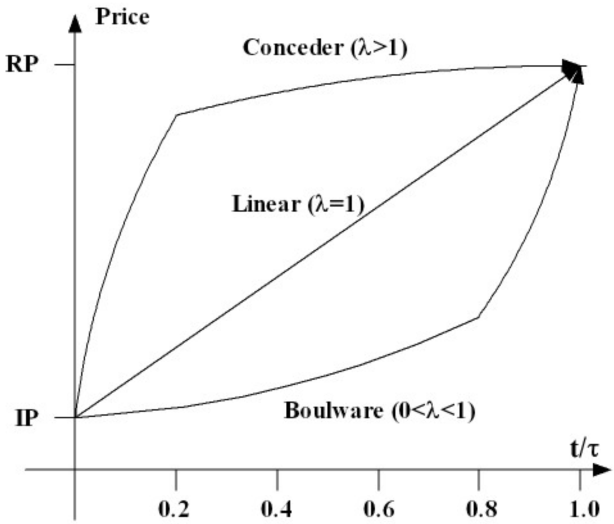


Fig. 7 Negotiation decision function for the buyer

b in Figure 7 [9]: (i) *Conceder*: when $\lambda > 1$, agent b gives more concession in the beginning of the negotiation, and less concession closer to the deadline; (ii) *Linear*: when $\lambda = 1$, agent b gives constant concession throughout the negotiation; and (iii) *Boulware*: when $0 \leq \lambda \leq 1$, agent b gives less concession initially, and more concession when the deadline is looming.

Finally, agent utility functions at time t are defined as per Equation 19.

$$U^a(p^t_{a\to\hat{a}}) = \begin{cases} RP^a - p^t_{a\to\hat{a}} & a = b \\ p^t_{a\to\hat{a}} - RP^a & a = s \end{cases} \tag{19}$$

$U^a(t)$ is the agent a's evaluation result of its opponent's offer at negotiation cycle t; based on this evaluation result, agent a can make a decision about its action.

The information state of agents. The sample negotiation is a negotiation with incomplete information, i.e. both agents s and b do not share their private information with each other.

The negotiation equilibrium. The Nash equilibrium is employed in our sample negotiation. The action, A^a, of agent a at time t is defined as follows:

$$A^a(p^t_{a\to\hat{a}}) = \begin{cases} Quit & if t > \tau^a, \\ Accept & if U^a(p^t_{\hat{a}\to a}) \geq U^a(p^{t'}_{a\to\hat{a}}), \\ Reject & if U^a(p^t_{\hat{a}\to a}) < U^a(p^{t'}_{a\to\hat{a}}). \end{cases} \tag{20}$$

where t' is the time of the next negotiation cycle. Therefore, the equilibrium strategy employed in this negotiation indicates that the agent will only accept the offer which can maximize self's benefit given the time constraint.

4.2 Partner Selection in Agent Negotiation

In the previous subsection, we briefly introduced agent negotiation and also indicated that it can be employed by agents to solve competition problems. However, due to the rapid development of autonomous agents and Internet techniques, most MAS work environments have become uncertain and dynamic. In such open and dynamic environments, when the number of potential partners is huge, performing complicated traditional negotiations with all potential partners may be expensive in terms of computational time and resources – indeed even impractical. Thus, we introduce an approach which can be employed by agents to choose partners from a large pool of potential partners with a high chance of reaching a good agreement in subsequent negotiations.

Agents may have different criteria on partner selection based on the purpose of their negotiation. Generally, in *cooperative* negotiation, agents will select a partner which will increase global benefits; while in *competitive* negotiation, agents prefer some partners which can supply the highest benefit to themselves. However, researchers have found that it is not always beneficial for agents to only cooperate with others about global tasks in *cooperative* negotiation [16] [42]. Also, in a *competitive* negotiation, agents should consider the global tasks. Furthermore, when agent behaviors are in between these two extreme cases, the existing partners selection approach is no longer suitable.

Zhang et al. proposed a dual concern model which provides an outline about the degrees of concern of an agent for its own and other's outcomes [43]. However, this model only briefly mentions the main trend of these degrees, without offering any calculation or comparison method. To address these problems, we further extended this dual concern model to allow agents to make reasonable decisions on their behaviors during partner selection based on these degrees. The extended dual concern model is shown in Figure 8.

In Figure 8, the x-axis indicates the percentage of self-concern of an agent while the y-axis is the percentage of other-concern from the agent. θ represents a *ReliantDegree* (i.e. reflection of the collaboration degree), where $\theta \in [0^\circ, 90^\circ]$. We use *selfishness* to represent the percentage of self-concern of an agent, which can be calculated by $\cos(\theta)$, and *selflessness* to represent the percentage of other-concern, which can be evaluated by $\sin(\theta)$. A *ReliantDegree* can illustrate the level of collaboration between the agent and its potential partner. From the extended model, we find that there are two extreme cases: (i) when the agent only emphasizes its own outcome, its negotiation attitude is completely *selfish* ($\theta = 0^\circ$); and (ii) when the agent only cares about its partner's outcome, its attitude is completely *selfless* ($\theta = 90^\circ$). From this model, it is clear that there are many other cases between completely *selfish* and completely *selfless* behaviors.

Suppose that there are n potential partners for an agent ID_x in a MAS. If we use a four-tuple p_i^x to represent the ith potential partner of agent ID_x, p_i^x can be formally defined by Equation 21:

$$p_i^x = < ID_i, GainRatio_i^x, ContributionRatio_i^x, ReliantDegree_i^x > \tag{21}$$

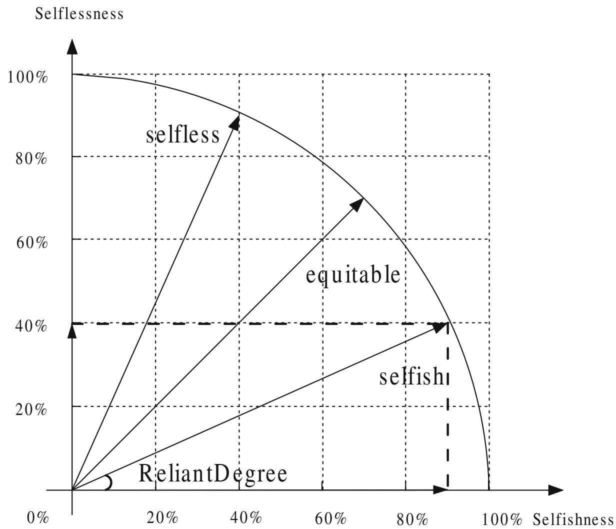


Fig. 8 The extended dual concern model

where ID_i is the unique identification of the ith potential partner, and $GainRatio_i^x$, $ContributionRatio_i^x$ and $ReliantDegree_i^x$ are factors used to evaluate the potential partner ID_i to be selected in the negotiation. These three factors are defined in Definitions 14 through 16, respectively.

Definition 14. $GainRatio_i^x$ is the percentage benefit that agent ID_x obtains out of the global benefit upon completion of the task. $GainRatio_i^x$ can be calculated as $GainRatio_i^x = \frac{S}{L} \times 100\%$, $GainRatio_i^x \in [0, 100\%]$, where S denotes the benefit that agent ID_x gains by selecting partner agent ID_i as its partner, and L denotes the global benefit by completing the task.

Definition 15. $ContributionRatio_i^x$ is the percentage benefit that agent ID_i obtains out of the global benefit upon completion of the task. $ContributionRatio_i^x$ can be calculated as $ContributionRatio_i^x = \frac{I}{L} \times 100\%$, $ContributionRatio_i^x \in [0, 100\%]$, where I denotes the benefit that partner agent ID_i gains by cooperating with agent ID_x, and L denotes the global benefit by completing the task.

Definition 16. $ReliantDegree_i^x$ represents agent ID_x's attitude to the negotiation, and also indicates the dynamic behavior of the agent, such as selfishness, selflessness, or other. $ReliantDegree_i^x$ can be calculated as $ReliantDegree_i^x = \arctan(\frac{Cr_x^i}{Cr_i^x})$, $ReliantDegree \in [0°, 90°]$, where Cr_x^i indicates how much agent ID_x trusts partner agent ID_i, which can be defined as the trading success ratio from partner agent ID_x to ID_i, or can be assigned by the system based on the performance record of partner agent ID_i, and Cr_x^i indicates how much partner agent ID_i trusts agent ID_x, which can be defined in the similar way as Cr_x^i.

Then an agent ID_x's evaluation of its potential partner ID_i is represented by $CollaborateDegree_i^x$, which is defined as follows:

$$CollaborateDegree_i^x = \Psi(p_i^x) \tag{22}$$

where $CollaborateDegree_i^x \in [0,1]$. This indicates the tendency that agent ID_i will be selected as a partner in subsequent negotiations by agent ID_x. The bigger the $CollaborateDegree_i^x$, the higher the likelihood that agent ID_i will be selected. The function Ψ specifies how to evaluate a potential partner. The interested reader can refer to [31] for a (non-linear) fuzzy approach to Ψ. In this chapter, we only consider a linear approach to Ψ.

In order to cover all potential cases in partner selection, we need to consider not only both *GainRatio* and *ContributionRatio*, but also the preference of the agent on these two criteria. It is proposed that the agent's preference on *GainRatio* and *ContributionRatio* can be represented by a normalized weight. Let w_g stand for the weight on *GainRatio*, w_c stand for the weight on *ContributionRatio*, and $w_c + w_g = 1$. Then the *CollaborationDegree* between agent ID_x and its potential partner ID_i is defined as follows:

$$CollaborateDegree_i^x \;=\; GainRatio_i^x \times w_g + ContributionRatio_i^x \times w_c \tag{23}$$

The *collaborationDegree* ($\in [0,1]$) indicates the degree for which the potential partner should be selected by the agent. The bigger the *collaborationDegree*, the more chance that the potential partner will be selected by the agent. In general, there are three extreme cases on different combinations of w_c and w_g, namely:

- When $w_g = 0$ and $w_c = 1$, *CollaborateDegree* is calculated based only on *ContributionRatio*, i. e. agent ID_x's attitude to negotiation is fully *selfless*.
- When $w_g = 1$ and $w_c = 0$, *CollaborateDegree* is calculated based only on *GainRatio*, i. e. agent ID_x's attitude to negotiation is fully *selfish*.
- When $w_g = w_c = 0.5$, *CollaborateDegree* is calculated based equally on *GainRatio* and *ContributionRatio*, i. e. agent ID_x's attitude to negotiation is *equitable*.
- Besides the above three cases, the restriction of $w_g + w_c = 1$ can also reflect agent ID_x's attitude to *GainRatio* and *ContributionRatio* in other cases.

The weights w_g and w_c can be calculated by employing the value of *ReliantDegree*, which are defined by Equation 24 and Equation 25, respectively.

$$w_g = \cos^2(ReliantDegree) \tag{24}$$

$$w_c = \sin^2(ReliantDegree) \tag{25}$$

Finally, by combining Equations 23 through 25, the potential partners are evaluated by considering the factors of *GainRatio*, *ContributionRatio* and *ReliantDegree*. The *collaborationDegree* between the agent ID_x and its potential partner ID_i is:

$$CollaborateDegree_i^x = GainRatio_i^x \times \cos^2(ReliantDegree_i^x) + ContributionRatio_i^x \times \sin^2(ReliantDegree_i^x) \quad (26)$$

where $CollaborateDegree_i^x \in [0,1]$. Then the collaboration degrees set (*CollaborateDegree*^{x}) between the agent ID_x and all its potential partners are generated according to Equation 27.

$$CollaborateDegree^x = \{CollaborateDegree_i^x\}, i \in [1,n] \quad (27)$$

Finally, any sorting algorithm can be employed to select favorable partners or exclude unsuitable partners from the collaboration degree set $CollaborateDegree^x$ for the agent ID_x.

In this chapter, three examples are demonstrated. In each example, agent g is going to select the most suitable partner from three potential partners (agents g_a, g_b and g_c). These examples will illustrate how the proposed approach selects the most suitable partner for the agent.

Table 4 Example 1

Agent	*Gain Ratio*	*Contribution Ratio*	*Reliant Degree*	*Collaborate Degree*
g_a	80%	20%	0°	0.8
g_b	50%	50%	0°	0.5
g_c	20%	80%	0°	0.2

In Example 1 (Table 4), as the agent ID_x performs as a fully *selfish* agent ($w_g = \cos^2(0°) = 1$ and $w_c = \sin^2(0°) = 0$), the potential partner who can offer the biggest *GainRatio* will be selected as the most suitable partner. From Table 4, agent g_a should be selected as the most suitable partner because it can contribute the highest *GainRatio* to agent ID_x among the three potential partners. By using our proposed Equation 26, agent g_a is also chosen as the most suitable partner because the *CollaborateDegree* for agent g_a is the largest among the three potential partners.

In Example 2 (Table 5), as the agent ID_x performs as a fully *selfless* agent ($w_g = \cos^2(90°) = 0$ and $w_c = \sin^2(90°) = 1$), agent g_c should be selected as the most suitable partner because it has the largest *ContributionRatio*. According to Equation 26, agent g_c is also selected as the most suitable partner because the *CollaborateDegree* for agent g_c is the largest among the three potential partners.

Table 5 Example 2

Agent	*Gain Ratio*	*Contribution Ratio*	*Reliant Degree*	*Collaborate Degree*
g_a	80%	20%	90°	0.2
g_b	50%	50%	90°	0.5
g_c	20%	80%	90°	0.8

Table 6 Example 3

Agent	*Gain Ratio*	*Contribution Ratio*	*Reliant Degree*	*Collaborate Degree*
g_a	80%	20%	0°	0.8
g_b	80%	20%	45°	0.5
g_c	80%	20%	90°	0.2

In Example 3 (Table 6), the agent ID_x has different attitudes to its potential partners. For potential partner g_a, agent ID_x performs as a *selfish* agent so that only the *GainRatio* (80%) will be used to select the most suitable partner. For potential partner g_b, agent ID_x performs as an *equitable* agent so that both *GainRatio* (80%) and *ContributionRatio* (20%) will be used to evaluate whether g_b could be chosen as a suitable partner. Therefore, the final benefit by considering both *GainRatio* and *ContributionRatio* for g_b should be between 20% and 80%. For potential partner g_c, agent ID_x performs as a *selfless* agent so that only the benefit of *ContributionRatio* (20%) will be used for the selection of g_c as a partner. By comparing the three cases, agent g_a should be selected as the most suitable partner because agent ID_x would gain the largest benefit(80%) when collaborating with agent g_a. According to Equation 26, agent g_a is also selected as the most suitable partner because the *CollaborateDegree* for agent g_a is the largest among the three potential partners.

Therefore, from the examples, it can be seen that by considering the factors of *GainRatio*, *ContributionRatio* and *ReliantDegree* between the agent and its potential partners, a partner selection mechanism can be generated dynamically to allow agents to adapt to their individual behaviors in negotiation. The selection result is also accurate and reasonable.

4.3 Behavior Prediction in Agent Negotiation

Negotiation is a means for agents to communicate and compromise to reach mutually beneficial agreements [10] [19]. However, in most situations, agents do not possess complete information about their partners' negotiation strategies, and may have difficulty in making a decision on future negotiation, such as how to select suitable partners [3] [25], or how to generate a suitable offer in the next negotiation cycle [29]. Therefore estimation approaches which can predict uncertain situations and possible changes in the future are required to help agents to generate good and efficient negotiation strategies. Research on partners' behavior estimation has been a very active area in recent years. Several estimation strategies have been proposed [6] [7] [41]. However, as these estimation strategies are used in real applications, some limitations begin to emerge, such as inaccurately estimated results or substantial time cost.

Machine Learning is a popular mechanism adopted by researchers in agent behavior estimation. In general, this kind of approach comprises two steps in order to estimate an agents' behavior. In the first step, the proposed estimation function is required to be well trained using the available training data. Therefore, in a way, the performance of the estimation function is virtually determined by the training result. In this step, as much data as possible is employed by designers to train a system. The training data could be synthetic and/or collected from the real world. Usually, synthetic data is helpful in training a function to enhance its problem solving ability for some particular issues, while real world data can help the function to improve its ability in complex problem solving. After the system has been trained, the second step is to employ the estimation function to predict partner behavior in the future. However, no matter which and how many data are employed by designers to train the proposed function, the training data will never be sufficiently comprehensive to cover all situations in reality. Therefore, even though an estimation function is well trained, it is also quite possible that some estimation results do not make sense at all for some kind of agents whose behavior records are not included in the training data. Currently, as negotiation environments become more open and dynamic, agents with different kinds of purpose, preference and negotiation strategy can enter and leave the negotiation dynamically. This Machine Learning-based agent behavior estimation function may not work well in some more flexible application domains, for reasons of (i) lack of sufficient data to train the system, and (ii) requiring too many resources during each training process.

In order to address the aforementioned issues, in this subsection we introduce a quadratic regression approach for analysis and estimation of partner behaviors during negotiation. The proposed quadratic regression function is:

$$u = a \times t^2 + b \times t + c \tag{28}$$

where u is the expected utility gained from a partner, t ($0 \leq t \leq \tau$) is the negotiation cycle and a, b and c are parameters which are independent of t. It is noticed that the three types of agents' behaviors in Figure 7 can be represented by this function as follows:

- $a > 0$ (Boulware): the rate of change in the slope is increasing, corresponding to smaller concession in the early cycles but large concession in later cycles.
- $a = 0$ and $b \neq 0$ (Linear): the rate of change in the slope is zero, corresponding to making constant concession throughout the negotiation.
- $a < 0$ (Conceder): the rate of change in the slope is decreasing, corresponding to large concession in early cycles, but smaller concession in later cycles.

We firstly transfer the proposed quadratic function 28 to a linear function, as follows. Let

$$\begin{cases} x = t^2 \\ y = t \end{cases} \tag{29}$$

Then Equation 29 can be rewritten as:

$$u = a \times x + b \times y + c \tag{30}$$

where both a and b are independent of variables x and y. Let pairs $(t_0, \hat{u_0})$, ..., $(t_n, \hat{u_n})$ be instances from each negotiation cycle. The distance (ε) between the real utility value ($\hat{u}_i$) and the expected value (u_i) should obey the Gaussian distribution $\varepsilon \sim N(0, \sigma^2)$, where $\varepsilon = \hat{u}_i - a \times x_i - b \times y_i - c$. Now since each $\hat{u}_i = a \times x_i + b \times y_i + c + \varepsilon_i$, $\varepsilon_i \sim N(0, \sigma^2)$, $\hat{u}_i$ is distinctive, and the joint probability density function for $\hat{u}_i$ is:

$$\begin{aligned} L &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2\sigma^2}(\hat{u}_i - ax_i - by_i - c)^2] \\ &= (\frac{1}{\sigma\sqrt{2\pi}})^n \exp[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c)^2] \end{aligned} \tag{31}$$

where L indicates the probability that a particular $\hat{u}_i$ may occur. Because each $\hat{u}_i$ comes from the historical record, we should use their probabilities as L's maximum value. Obviously, in order to make L achieve its maximum, $\sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c)^2$ should achieve its minimum value. Let

$$Q(a,b,c) = \sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c)^2 \tag{32}$$

We calculate the first-order partial derivative for $Q(a,b,c)$ on a, b and c respectively, and let their results equal zero, as follows:

$$\begin{cases} \frac{\partial Q}{\partial a} = -2\sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c)x_i = 0 \\ \frac{\partial Q}{\partial b} = -2\sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c)y_i = 0 \\ \frac{\partial Q}{\partial c} = -2\sum_{i=1}^{n}(\hat{u}_i - ax_i - by_i - c) = 0 \end{cases} \tag{33}$$

Then the Equations 33 can be expanded to:

$$\begin{cases} (\sum_{i=1}^{n} x_i^2)a + (\sum_{i=1}^{n} x_i y_i)b + (\sum_{i=1}^{n} x_i)c = \sum_{i=1}^{n} x_i \hat{u}_i \\ (\sum_{i=1}^{n} x_i y_i)a + (\sum_{i=1}^{n} y_i^2)b + (\sum_{i=1}^{n} y_i)c = \sum_{i=1}^{n} y_i \hat{u}_i \\ (\sum_{i=1}^{n} x_i)a + (\sum_{i=1}^{n} y_i)b + nc = \sum_{i=1}^{n} \hat{u}_i \end{cases} \tag{34}$$

Let PU, PA, PB and PC be the coefficient matrices as follows:

$$PU = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i & n \end{vmatrix} \tag{35}$$

$$PA = \begin{vmatrix} \sum_{i=1}^{n} x_i \hat{u}_i & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} y_i \hat{u}_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} \hat{u}_i & \sum_{i=1}^{n} y_i & n \end{vmatrix} \tag{36}$$

$$PB = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \hat{u}_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i \hat{u}_i & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} \hat{u}_i & n \end{vmatrix} \tag{37}$$

$$PC = \begin{vmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \hat{u}_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \hat{u}_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} \hat{u}_i \end{vmatrix} \tag{38}$$

Because $PU \neq 0$, the parameters a, b and c have a unique solution, which is

$$\begin{cases} a = \frac{PA}{PU} \\ b = \frac{PB}{PU} \\ c = \frac{PC}{PU} \end{cases} \tag{39}$$

Previously, we proposed a quadratic regression function to predict partners' behavior, and also specified how to determine parameters a, b and c. However, it should be mentioned that the proposed quadratic regression function can only provide an *estimation* on possible partner behaviors, which might not exactly accord with the partners' *real* behaviors. In practice, agents' estimated behaviors should be close to their real actions. The closer the estimated behaviors to the real actions, the higher the probability that the estimated behaviors will occur. Thus we can deem that the differences (ε) between the estimation behaviors and the real behaviors obey the Gaussian distribution $N(\varepsilon, \sigma^2)$. Thus, if the deviation σ^2 can be calculated, we can make a precise decision on partner behaviors. It is known that there is more than 68% probability that partners' expected behaviors are located in the interval $[u-\sigma, u+\sigma]$, more than 95% that partners' expected behaviors lie in $[u-2\sigma, u+2\sigma]$, and more than 99% in the interval $[u-3\sigma, u+3\sigma]$.

In order to calculate the deviation σ, we firstly calculate the distance between the estimation results (u_i) and the real results on partners' behaviors ($\hat{u}_i$) as follows:

$$d_i = \hat{u}_i - u_i \tag{40}$$

It is known that all d_i ($i \in [1, n]$) obey the Gaussian distribution $N(0, \sigma^2)$. Then σ can be calculated as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (d_i - \overline{d})^2}{n}} \tag{41}$$

where,

$$\overline{d} = \frac{1}{n} \sum_{i=1}^{n} d_i \tag{42}$$

Now by employing the Chebyshev inequality, we can calculate (1) the interval of partners' behaviors according to any accuracy requirements; and (2) the probability that any particular behavior may occur in potential partners in the future.

The Chebyshev's inequality is:

$$P(|\hat{u}_i - u_i| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2} \tag{43}$$

where $\hat{u}_i$ is an instance, u_i is the mathematical expectation, σ is the deviation and ε is the accuracy requirement. This function indicates that based on a particular accuracy requirement ε, the possibility that the real behavior $\hat{u}_i$ is included in the interval $[u_i - \sigma, u_i + \sigma]$ is greater than $1 - \frac{\sigma^2}{\varepsilon^2}$.

In this chapter, we demonstrate three scenarios to indicate the agent behaviors prediction approach. Also, we compare the proposed quadratic regression approach with the Tit-For-Tat [9] and random approaches. The experimental results illustrate the outstanding performance of our proposed approach. In order to simplify the implemented process, all agents in our experiment employ the NDF [11] negotiation strategy. The partners' behaviors cover all possible situations in reality, which are conceder, linear and boulware. In experiments, we use the average error (EA) to evaluate the experimental results. Let u_i be the predicted result in cycle i and $\hat{u}_i$ be the real instance in cycle i, then AE_i is defined as follows:

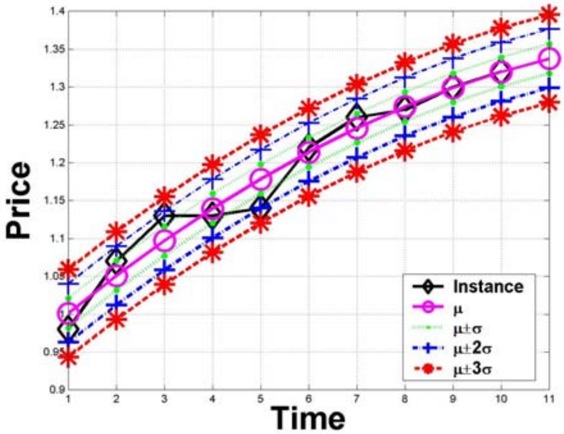
$$AE_i = \frac{\sum_{k=1}^{i} |\hat{u}_i - u_i|}{i} \tag{44}$$

The AE_i indicates the difference between the estimated results and the real value. The smaller the value of AE_i, the better the prediction result.

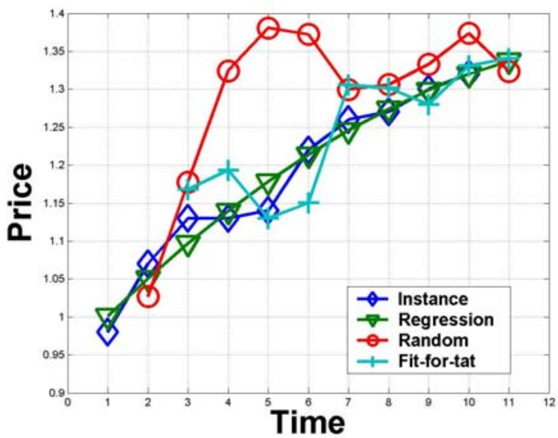
In the first scenario, a buyer wants to purchase a mouse pad from a seller. The acceptable price for the buyer is in $[\$0, \$1.4]$. The deadline for the buyer to finish this purchasing process is 11 cycles. In this experiment, the buyer adopts conceder behavior in the negotiation, and the seller employs the proposed approach to estimate the buyer's possible price in the next negotiation cycle. The estimated results are displayed in Figure 9(a) and the regression function is:

$$u = -0.002 * t^2 + 0.055 * t + 0.948$$

It can be seen that in the $8th$ negotiation cycle, the proposed approach estimates a price of \$1.26 from the buyer in the next cycle. Then according to the historical record in the $8th$ cycle, the real price given by the buyer in this cycle is \$1.26, which is exactly same as the estimation price. Furthermore, it can be seen that in cycles 4, 6, 9 and 10, the estimated prices are also the same as the real value. The estimation prices for the $2nd$, $3rd$ and $7th$ cycles are \$1.05, \$1.10 and \$1.25, respectively, and the real prices given by the buyer in these cycles are \$1.07, \$1.13, and 1.26, which differ only slightly between the estimated prices and real prices. According to Figure 9(a), all real prices are located in the interval of $[\mu - 2\sigma, \mu + 2\sigma]$, where μ is the estimated price and σ is the changing span. The $AE_{10} = 0.015$, which is only 1% of buyer's reserve price. Therefore, the prediction results by employing the proposed approach are very reliable.



(a) Prediction results for scenario 1

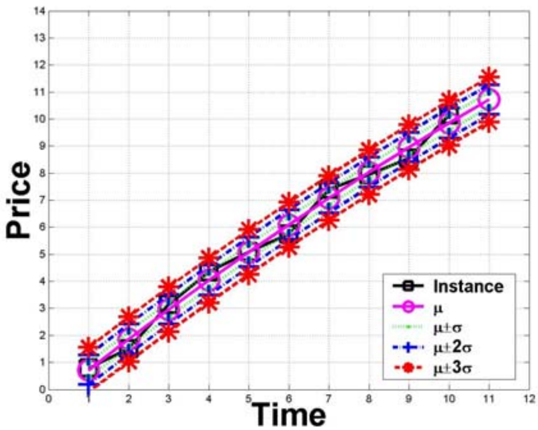


(b) Prediction results comparison for scenario 1

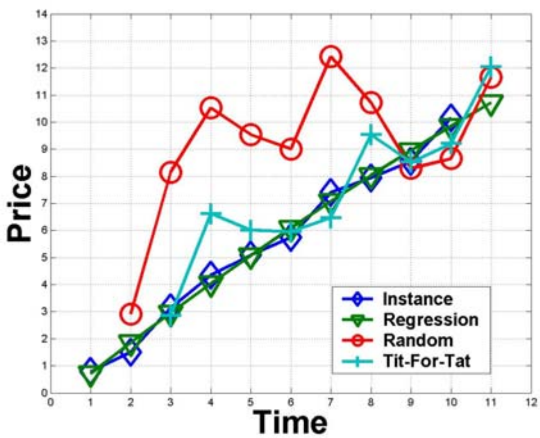
Fig. 9 Scenario 1

In Figure 9(b), we compare results between the proposed approach and two other estimation approaches (Tit-For-Tat and random). It can be seen that even though the Tit-For-Tat approach can follow the trend of changes in the buyer's price, $AE_{10} = 0.078$ which is five times our proposed approach. For the random approach, it cannot even catch the main trend. The AE_{10} for the random approach is 0.11, which is ten times our proposed approach. The experimental results convince us that the proposed approach outperforms both the Tit-For-Tat and random approaches when a buyer adopts conceder negotiation behavior.

In the second scenario, a buyer wants to buy a keyboard from a seller. The desired price for the buyer is in the interval of $[\$0, \$14]$. We let the buyer employ the linear negotiation strategy, and still set the deadline to 11 cycles. The seller will employ



(a) Prediction results for scenario 2



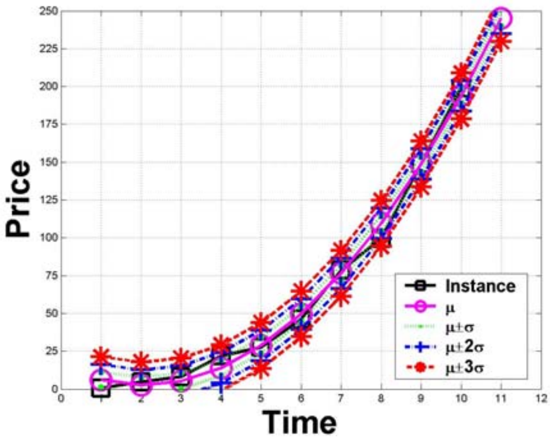
(b) Prediction results comparison for scenario 2

Fig. 10 Scenario 2

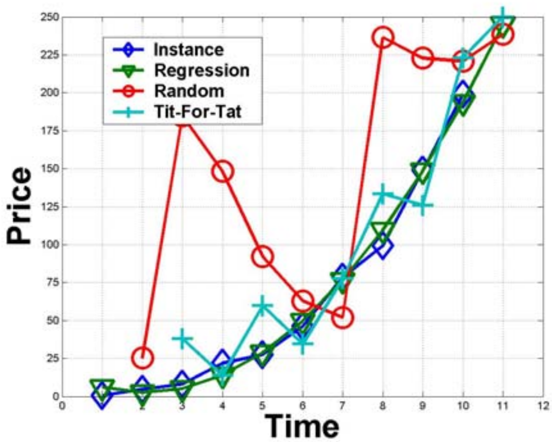
our proposed prediction function to estimate the buyer's offer. The estimated results are illustrated in Figure 10(a) and the estimated quadratic regression function is:

$$u = -0.015 * t^2 + 1.178 * t - 0.439$$

It can be seen that in the $3rd$, $5th$ and $8th$ cycles, the estimated prices are exactly the same as the real offers made by the buyer. The biggest difference between the estimated price and the real value is just 0.4, which happens in the $9th$ cycle. The average error in this experiment is only $AE_{10} = 0.24$, which is no more the 2% of the buyer's reserve price. The estimated quadratic regression function fits the real prices very well.



(a) Prediction results for scenario 3



(b) Prediction results comparison for scenario 3

Fig. 11 Scenario 3

Figure 10(b) compares results for the Tit-For-Tat approach, random approach and our proposed approach. It can be seen that the proposed approach is much closer to the real price than the other two approaches. The average error for the Tit-For-Tat approach is $AE_{10} = 2.52$, which is 18% of the buyer's reserve price. The average error for the random approach is very high – $AE_{10} = 4.82$ (34% of the buyer's reserve price). A second experimental result is that when partners perform with linear behaviors, the proposed approach also outperforms the other two approaches.

In the third scenario, a buyer wants to purchase a monitor from a seller. The suitable price for the buyer is in $[\$0, \$250]$. In this experiment, the buyer employs a boulware strategy in the negotiation. The deadline is still 11 cycles. The estimated quadratic function is:

$$u = 3.038 * t^2 - 12.568 * t + 15.632$$

The estimated results are shown in Figure 11(a), it can be seen that the proposed quadratic regression approach predicted buyer's prices successfully and accurately. Except for the $4th$ and $8th$ cycles, other estimated prices differ very little from the buyer's real offers. The average error in this experiment is $AE_{10} = 4.07$, which is only 1.6% of the buyer's reserve price. Therefore, we can say with confidence that from these estimation results, the seller can make accurate judgement about the buyer's negotiation strategy, and make reasonable responses in order to maximize its own benefit.

Finally, Figure 11(b) shows comparison results with two other estimation functions for the same scenario. For the Tit-For-Tat approach, the average error is $AE_{10} = 57.74$, which is 23% of the buyer's reserve price. For the random approach, the average error is $AE_{10} = 83.12$, which is 33% of the buyer's reserve price. Therefore, it can be seen that when the agent performs a boulware behavior, the proposed approach significantly outperforms the other two approaches.

From these experimental results, we can conclude that the estimated quadratic function regression approach can successfully estimate partners' potential behaviors. Moreover, the estimation results are accurate and sufficiently reasonable to be adopted by agents to modify their strategies in negotiation. The comparison results among the three types of agent behavior estimation also demonstrate the outstanding performance of our proposed approach.

In this section, we introduced agent negotiation for solving complex problems between collaborative agents. Firstly, we pointed out that agent competition can also be involved in collaborative problems. Then we introduced some basic knowledge about agent negotiation for conflict resolution. Furthermore, we introduced a partner selection approach and agents' behavior prediction approach for complex negotiation environments and illustrated some experimental results to show the improvements. In conclusion, we can say that agent negotiation is a very significant mechanism for agents to solve conflicts which may occur during complex problem solving procedures.

5 Conclusion

Complex problem solving requires diverse expertise and multiple techniques. MAS is a particularly applicable technology for complex problem solving applications. In a MAS, agents that possess different expertise and resources collaborate together to handle problems which are too complex for individual agents. Generally, agent collaborations in a MAS can be classified into two groups, namely agent cooperation and agent competition. These two kinds of collaborations are unavoidable for most MAS applications, but both present challenges. In addition, two main approaches for complex problem solving via agent cooperation and agent competition have been introduced – these being a dynamic team formation mechanism for cooperative agents, and a partner selection strategy for competitive agents. These two approaches can be applied to coordinate utility conflicts among agents, and make a MAS more suitable for open dynamic working environments.

Research into dynamic team formation can be extended in the following two directions. Currently, team formation research is based on a simple agent organisation. However, in many current MAS applications, more complex organisational structures, such as congregation [2], are adopted. Building a mechanism to support complex organisational formation is one research direction for the future. Furthermore, different organisational structures are suitable for different circumstances. In a complex dynamic working environment, agents may need to choose different organisational structures due to a changing environment. To develop mechanisms that enable agents to not only select cooperation partners but also dynamically choose organisational structures is another avenue for future research.

Further work on agent negotiation can proceed in two directions, as (i) Currently, most agent negotiation strategies and protocols can only handle the negotiation with single issue. However, with expansion of application domains, negotiating multiple issues will become a significant trend. Therefore, research on multi-issue negotiation will become a future direction. (ii) Most negotiation environments currently mainly focus on the static situations, but fail to take into account where a negotiation environment becomes open and dynamic. In an open and dynamic environment, agents can perform more flexibly to enhance their benefits. Also an open and dynamic negotiation environment is much more efficient in handling real world applications. Therefore, changing the negotiation environment from static to open and dynamic is another significant research direction on the topic of agent negotiation for the future.

Another potential direction is to extend our current research to complex domains in which agents can show semi-competitive behaviours or temporary collaborative behaviours in different situations.

Acknowledgements. The authors would like to acknowledge the support of the Intelligent Systems Research Centre at the University of Wollongong.

References

1. Artikis, A., Pitt, J.: A Formal Model of Open Agent Societies. In: Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, pp. 192–193 (2001)
2. Brooks, C., Durfee, E., Armstrong, A.: An Introduction to Congregating in Multiagent Systems. In: Proceedings of 4th International Conference on Multiagent Systems, Boston, USA, pp. 79–86 (2000)
3. Brzostowski, J., Kowalczyk, R.: On Possibilistic Case-Based Reasoning for Selecting Partners in Multi-agent Negotiation. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS, vol. 3339, pp. 694–705. Springer, Heidelberg (2004)
4. Candea, C., Hu, H., Iocchi, L., Nardi, D., Piaggio, M.: Coordination in Multi-agent RoboCup Teams. Robotic and Autonomous Systems 36(2), 67–86 (2001)
5. Castelpietra, C., Iocchi, L., Nardi, D., Piaggio, M., Scalzo, A., Sgorbissa, A.: Coordination among Heterogeneous Robotic Soccer Players. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Takamatsu, Japan, pp. 1385–1390 (2000)

6. Chajewska, U., Koller, D., Ormoneit, D.: Learning An Agent's Utility Function by Observing Behavior. In: Proceedings of 18th International Conference on Machine Learning, pp. 35–42. Morgan Kaufmann, San Francisco (2001)
7. Coehoorn, R., Jennings, N.: Learning on Opponent's Preferences to Make Effective Multi-issue Negotiation Trade-offs. In: Proceedings of the 6th International Conference on Electronic Commerce, ICEC 2004, Delft, Netherlands, October 2004, pp. 59–68. ACM Press, New York (2004)
8. Eberhart, R., Simpson, P., Dobbin, R.: Computational Intelligence PC Tools. AP Professional Press, Orlando (1996)
9. Faratin, P., Sierra, C., Jennings, N.: Negotiation Decision Functions for Autonomous Agents. Journal of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)
10. Fatima, S., Wooldridge, M., Jennings, N.: Optimal Agendas for Multi-issue Negotiation. In: Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2003), pp. 129–136. ACM Press, New York (2003)
11. Fatima, S., Wooldridge, M., Jennings, N.: An Agenda-Based Framework for Multi-Issue Negotiation. Artificial Intelligence 152(1), 1–45 (2004)
12. Fatima, S., Wooldridge, M., Jennings, N.: Optimal Negotiation of Multiple Issues in Incomplete Information Settings. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), New York, USA, pp. 1080–1087. IEEE Computer Society, Los Alamitos (2004)
13. Gerkey, B., Mataric, M.: Multi-robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. In: Proceedings of the IEEE International Conference on Robotics and Automation, Taibei, China, pp. 3862–3868 (2003)
14. Horling, B., Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. The Knowledge Engineering Review 19, 281–316 (2005)
15. Ito, T., Parkes, D.: Instantiating the Contingent Bids Model of Truthful Interdependent Value Auctions. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, pp. 1151–1159 (2006)
16. Jung, H., Tambe, M., Kulkarni, S.: Argumentation as Distributed Constraint Satisfaction: Applications and Results. In: Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, pp. 324–331. ACM Press, New York (2001)
17. Klir, G., Yuan, B. (eds.): Fuzzy Sets and Fuzzy Logic Theory and Applications. Prentic Hall, Upper Saddle River (1995)
18. Koes, M., Nourbakhsh, I., Sycara, K.: Heterogeneous Multirobot Coordination with Spatial and Temporal Constraints. In: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI), June 2005, pp. 1292–1297. AAAI Press, Menlo Park (2005)
19. Kraus, S.: Strategic Negotiation in Multiagent Environments. The MIT Press, Cambridge (2001)
20. Lesser, V.: Reflections on the Nature of Multi-agent Coordination and Its Implications for an Agent Architecture. Journal of Autonomous Agents and Multi-Agent Systems 1(1), 89–111 (1998)
21. Lesser, V.: Cooperative Multiagent Systems: A Personal View of the State of the Art. IEEE Transactions on Knowledge and Data Engineering 11(1), 133–142 (1999)
22. Lesser, V., Horling, B., Klassner, F., Raja, A., Zhang, S.: BIG: An Agent for Resource-Bounded Information Gathering and Decision Making. Artificial Intelligence Journal, Special Issue on Internet Information Agents 118(1-2), 197–244 (2000)
23. Li, C., Chawla, S., Rajan, U., Sycara, K.: Mechanisms for Coalition Formation and Cost Sharing in an Electronic Marketplace. Technical Report CMU-RI-TR-03-10, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (April 2003)
24. Milanovic, N., Malek, M.: Current Solutions for Web Service Composition. IEEE Internet Computing Magazine 8(6), 51–59 (2004)

25. Munroe, S., Luck, M., d'Inverno, M.: Motivation-Based Selection of Negotiation Partners. In: 3rd International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2004, pp. 1520–1521. IEEE Computer Society, Los Alamitos (2004)
26. Nash, J.: The Bargaining Problem. Econometrica 18, 155–162 (1950)
27. Oliveira, E., Rocha, A.: Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Process. In: Sierra, C., Dignum, F.P.M. (eds.) AgentLink 2000. LNCS, vol. 1991, pp. 78–96. Springer, Heidelberg (2001)
28. Osborne, M., Rubenstein, A.: A Course in Game Theory. The MIT Press, Cambridge (1994)
29. Parsons, S., Sierra, C., Jennings, N.: Agents that Reason and Negotiate by Arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
30. Rathod, P., desJardins, M.: Stable Team Formation among Self-interested Agents. In: Proceedings of AAAI Workshop on Forming and Maintaing Coalitions in Adaptive Multiagent Systems, San Jose, USA, pp. 29–36 (2004)
31. Ren, F., Zhang, M., Bai, Q.: A Fuzzy-Based Approach for Partner Selection in Multi-Agent Systems. In: Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), Melbourne, Australia, pp. 457–462. IEEE Computer Society, Los Alamitos (2007)
32. Rigolli, M., Brady, M.: Towards a Behavioral Traffic Monitoring System. In: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, Netherlands, pp. 449–454 (2005)
33. Rosenschein, J., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge (1994)
34. Shen, J., Zhang, X., Lesser, V.: Degree of Local Cooperation and its Implication on Global Utility. In: Proceedings of 3rd International Joint Conference on Autonomous Agents and MultiAgent Systems, New York, USA, July 2004, vol. 2, pp. 546–553 (2004)
35. Sim, K.: A Survey of Bargaining Models for Grid Resource Allocation. ACM SIGECOM: E-commerce Exchange 5(5), 22–32 (2005)
36. Tambe, M.: Agent Architectures for Flexible, Practical Teamwork. In: Proceedings of the 14th National Conference on Artificial Intelligence, Rhode Island, USA, pp. 22–28 (1997)
37. Tambe, M.: Towards Flexible Teamwork. Journal of Artificial Intelligence Research 7, 83–124 (1997)
38. Tambe, M.: Implementing Agent Teams in Dynamic Multi-agent Environments. Applied Artificial Intelligence 12(2-3), 189–210 (1998)
39. Tsvetovat, M., Sycara, K.: Customer Coalitions in the Electronic Marketplace. In: Proceedings of the 4th International Conference on Autonomous Agents, Barcelona, Spain, pp. 263–264. ACM Press, New York (2000)
40. Yamamoto, J., Sycara, K.: A Stable and Efficient Buyer Coalition Formation Scheme for E-marketplaces. In: Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, pp. 576–583. ACM Press, New York (2001)
41. Zeng, D., Sycara, K.: Bayesian Learning in Negotiation. International Journal of Human-Computer Studies 48(1), 125–141 (1998)
42. Zhang, X., Lesser, V., Wagner, T.: Integrative Negotiation In Complex Organizational Agent Systems. In: Proceedings of the 1st International Joint Conference on Autonomous Agents & MultiAgent Systems (AAMAS 2002), July 2002, pp. 503–504 (2002)
43. Zhang, X., Lesser, V., Wagner, T.: Integrative Negotiation among Agents Situated in Organizations. IEEE Transactions on Systems, Man, and Cybernetics, Part C 36(1), 19–30 (2006)