Finding interaction partners using cognition-based decision strategies

Partha Sarathi Dutta, Luc Moreau, Nicholas R. Jennings
Department of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO 17 1BJ, U.K.
Email:{psa01r,L.Moreau,nrj}@ecs.soton.ac.uk

Abstract

In this paper, we develop decision making heuristics for rational agents using artefacts of cognition such as observation, learning and memory. Specifically, we extend previous research in this area by incorporating essential aspects of multi-agent interactions such as building behavioural models via observation, selectively choosing interaction partners and forming cooperating groups by identifying mutual capabilities. In particular, we demonstrate that cognitive capabilities enable agents to successfully identify matching partners and establish cooperative groups in a community of selfish agents with varying expertise.

1 Introduction

Agent-based systems are increasingly being used in open environments such as the Semantic Web, the Grid and electronic commerce applications. Typically, agents deployed in such systems have different owners and act as self-interested entities to achieve their corresponding goals. In such systems, it is not feasible to predict in advance all possible services that agents would require in order to achieve their individual and collective goals [Jennings, 2001]. Also, it is not possible to enumerate a priori the contexts in which an agent might need to interact with another for its service requirements. Against this background, this paper develops a set of decision heuristics that assist self-interested agents in finding the necessary service providers by interacting with a limited number of known acquaintances. In particular, we demonstrate that flexible strategies can be developed using elements of cognition, such as memory and learning, that allow agents to successfully identify matching partner agents and form stable interacting groups in a population of self-interested agents.

Previous research has developed strategies of rational behaviour for agents interacting in competitive multi-agent domains [Axelrod, 1984; Sandholm and Crites, 1995]. These works have typically focused on developing utility-maximizing decision strategies for agents playing idealized "games" such as the Prisoner's Dilemma. However, issues such as whom to interact with, what information to reveal to whom and when, whether and how interactions with one agent can affect relationships with others, and whether cooperation can emerge by identifying mutual service requirements are left unaddressed. To rectify this shortcoming, in this paper we have designed agent decision making heuristics, based on capabilities of individual cognition, that address these fundamental issues. Specifically, cognitive abilities harness a rich set of information about the environment that facilitate context analysis and allow for flexible behaviour. Therefore, we believe that decision mechanisms supported by cognitive abilities are likely to lead to robust and flexible agent models that are suitable for deployment in complex, dynamic domains of practical importance.

To apply and evaluate the decision heuristics, we have simulated a generic setting where agents are required to complete sets of tasks that require both expertise and resource to be finished. Not all agents have all the expertise and resource capabilities to complete any task. Hence, an agent searches for service from those with the required capabilities. This service can be sought from agents with whom interactions exist currently. Alternatively, agents can search for new acquaintances if their current acquaintances cannot provide the required service. In so doing, the realistic constraint of each agent having a limited number of acquaintances is assessed. This is important because it is impractical for an agent to interact with all the other agents in the domain. Also, we have designed agents that interact only with those that are capable of providing the required service. Without this, a self-interested agent would stop interacting with them. During interactions, the agents observe, analyse, learn and memorize models of other agents. With the above specifications and equipped with cognitive capabilities, our agents exhibit context-dependent behaviour with respect to, whom to request assistance from, whether to acknowledge a request for assistance, whether to seek a new acquaintance, when to reveal information about new acquaintances to other agents and whether it is "useful" to continue interactions with a particular acquaintance. In this work all the decisions are solely targeted towards improving the agent's own utility: we do not consider inherently social agents that act towards the goal of the community [Panzarasa *et al.*, 2001]. In this context, we show that the agents successfully identify the capabilities of other agents and groups of mutually cooperating partners emerge despite the constraint of interacting with a limited number of acquaintances and all being self-interested entities.

This work advances the state of the art in the following ways: (a) strategies of rational decision making in a multi-agent context are designed using artefacts of cognition and, (b) agent decision strategies are shown to be effective in identifying partners and in forming cooperative groups in a community of selfish agents .

The remainder of the paper is structured in the following way. Section 2 discusses related research to establish the context for our work. Section 3 describes the domain in which we evaluate the agent decision strategies. Section 4 details formulations of the different decision heuristics. Subsequently, Section 5 outlines a brief summary of the simulation before Section 6 reports the experimental results. Finally, Section 7 concludes and presents proposals for future work.

2 Related Work

A key research issue in the field of multi-agent systems (MAS) is that of designing adaptive decision mechanisms that support agent interactions in open environments. In addition to MAS researchers proposing theoretical and computational models of agents to explain such behaviour, useful results have also been obtained from social and cognitive sciences. Both strands of work are discussed in this section.

Azoulay-Schwartz and Kraus [2001] have developed a game theoretic approach for strategy selection where agent-pairs repetitively interact by asking and responding to queries. However, in their work, agents do not evaluate the consequences of decisions on their relationships with other agents, neither do they take decisions based on their current relationships with others. Multiple self-interested agents interacting to form coalitions have been studied in the work of Lerman et. al [2000]. However, in their work, agents do not employ cognitive abilities to recognise behaviours of other agents, nor do they adapt their action selection during interactions. Learning another agent's strategy and analysing past experience to take interaction decisions are employed in [Sen, 2002]. However, assumptions such as any agent can interact with any other constrain the scalability of his strategies. Also, agents do not model environment characteristics and do not employ future lookahead to predict the behaviour of another agent. We remove all these restrictions in our work. Castelfranchi and Falcone [1998] advocate trust-based interaction mechanisms for multi-agent collaboration. Although they use a different mechanism, their work is related to our approach in that it is inspired by the notion of using cognition for autonomous decision making. Yu et. al [2003] simulate a social network for information access in a multi-agent scenario. They use referrals to search for matching expertise and show that learning improves the network quality which is a measure of how close in the network are agent-pairs with matching expertise. In our work, since agents are solely self-interested, they deliberate on the consequences of such decisions in terms of the cost incurred. Each such decision is facilitated by their individual cognition and directed towards improving their own utility. Agents with both individual and societal decision-making components are studied in [Hogg and Jennings, 2001]. Here, learning others' behaviour and employing a meta-level reasoning to adapt strategies of interactions are shown to yield improvements in utility earnings. However, though this work dwells on adaptive agent behaviour, it does not address emergence of group structures as an influence of the task environment and the individual expertise of agents.

Our work also builds upon certain design methodologies from the social network literature (e.g., [Jin *et al.*, 2001; Kautz *et al.*, 1997]). In these works, an entity is set to have a limited number of direct acquaintances at any time. Also, the relationships between acquaintances weaken if interactions fail to occur. Entities get referred to prospective collaborators via acquaintances which facilitates information dissemination. In our work, we adapt these specifications to shape our simulation dynamics. Specifically, the social network literature deals with abstract simulations of the growth of a network from a random graph, and this is not concerned with studying agent decision strategies as mechanisms for the emergence of group structures in a multi-agent system.

The final strand of related work is that of cognitive science. This is mostly focused on explaining individual cognition and rational judgments [Betchel and Graham, 1998; Forgas, 1991]. The first significant attempt to unite the threads of research in cognitive science, sociology and multi-agent systems can be accredited to the work of Sun [2001]. His work emphasises the application of the theories of cognitive science in agent-based applications. Understanding well the issues of individual cognition would enable agent researchers to model agent behaviour more accurately in open environment settings. In our work, we demonstrate that cognitive knowledge empowers an agent with the reasoning capability that helps selecting the appropriate actions to maximize its self-interest.

3 Domain and Agent Roles

We start by describing a generic setting in which agents execute tasks which require certain levels of expertise and resource, not all of which are available to all agents. This setting can be mapped to several domains of practical interest with appropriate elaboration and refinement, viz., peer-to-peer networks, supply chain networks, etc. This lack of all the required expertise and resource provides an agent with the incentive to locate other agents with the required capabilities and acquire "benefit" by receiving assistance on tasks. This, in turn, saves time and resource (required for completing the task) by handing the task over to the assisting agent. In this context, an agent can decide to initiate an interaction by asking for assistance for a task from an acquaintance with whom it has interacted in the past and has estimates of the latter's behaviour. Alternatively, it can attempt to discover new acquaintances in the hope that this will improve the quality of assistance it can get on tasks. The agent who receives requests for assistance takes the decision to accept or deny based on factors such as its estimates of the requester's capabilities, expected future "usefulness" of the requester, estimate of task cost, and the extent to which its current acquaintances cater to its requirements. Thus, for example, the probability of giving service increases if the agent evaluates that the requester can potentially provide service in the future, if the cost of the requested service is low or if it believes that serving the particular requester does not affect its relations with other acquaintances (see Section 4 for more details).

However, at first glance, it seems that agreeing to serve a request apparently contradicts selfish behaviour because the agent serving incurs cost in terms of time and resource spent for the service. However, a selfish agent can potentially serve if it evaluates a high future prospect of receiving help from the agent requesting service. Also, the fact that serving an agent earns an agent some "goodwill", helps to get reciprocated by receiving service in future. So, we identify that agents can act in two distinct roles; that of an interaction initiator or *requester* and that of a service *provider*. Some practical issues concerning such multi-agent interactions are described in the following paragraphs.

Due to communication and computational constraints, it is not possible for an agent to keep track of or interact with every other agent in a large system. We have, therefore, limited the maximum number of acquaintances that an agent can interact with at any time to a certain fixed percentage of the total number of agents. For assistance on an assigned task, an agent seeks help from these acquaintances (also described as its "neighbours" hereafter). Alternatively, when searching for a new agent, instead of depending on some ad-hoc protocol (such as, randomly seeking) of finding an agent in the entire population, the seeker can request a referral from one of its neighbours — a referral request. If acknowledged, this referral request would provide the requesting agent with the reference of another which was not present in its neighbourhood. However, a referral can be an agent that the requester already knows, in which case, the latter denies from accepting the referral. Depending on the capability of the referred agent, an agent can get new acquaintances with better service capabilities than the current ones. In this way an agent can, potentially, extend its neighbourhood size beyond the initial set and locate better matching partners.

In addition to seeking assistance or referrals from neighbours, an agent can renege interactions with an existing neighbour. As self-interested entities, agents rate their interactions with neighbours in proportion to the assistance (and hence, to the amount of benefit earned) they receive from the latter. The "relationship" (based on help requests/responses) is stronger for higher benefit earned from such interactions and vice versa. An agent tends to interact more with a neighbour from which it receives greater service assistance than from another in order to maximize the utility earned via interactions. This process leads to the cumulative strengthening of the relationships between some agents and weakening of others, and finally, reneging on some neighbours. However, an erratic task environment, with frequent changes in capability requirements for agents, can prevent long-standing stable relationships from developing. In this work, we have used a task environment where task requirements are time periodic, with a short period compared to the total time agents interact in the simulation so that they have sufficient time to learn the task arrival pattern and use the information in their decision strategies (see Section 4 for details). Lastly, we note that agents give lesser importance to past interactions (i.e., interaction ratings decay over time).

Against this background, brief descriptions of the two essential components of the agent cognitive model are given below.

Task environment: We consider an agent's model of the task environment to consist of estimates of how the expertise and resource requirements of tasks vary with time. In our simulations, each task can have one of two possible required expertise types and one of two possible required resource types; hence, a task can have one of 4 different types (designating the two expertise types as exp1 and exp2, and the two resource types as res1 and res2, the 4 different task types can be (exp1, res1), (exp1, res2), (exp2, res1) and (exp2, res2)). Also, we assume a fixed probability distribution over the task types as per Table 1. It shows the probabilities of occurrences of different task types at each time instance (for example, row 1 shows the probabilities with which task types 1, 2, 3 and 4 can occur at time instant $current\_time$ mod 1). The expertise and resource components of each task type are shown within parentheses under the corresponding task type box. This probability pattern repeats itself after every F time instances, which is the task type periodicity. We assume a task type periodicity of 4 time instances. To keep the simulation simple, we have provided the agents with perfect information about this task distribution [1].

Capability models: An agent records the expertise and resource requirements of another agent when the latter requests assistance for a certain task type. This information is used to build estimates of the task types in which the requester requires assistance from other agents [2]. These estimates are utilised by agents to compute the dependence of its acquaintances on its own capabilities and are important in decisions described in Section 4.5. Additionally, an agent learns the capabilities of a service provider when it receives assistance from the latter. In this work, the agents employ reinforcement learning to build such estimates (see Section 4 for details).

There are four possible capability types that an agent can have — having expertise and resource of either one, two, three or all four task types. Thus, we label an agent as $type_i$ ($i \in \{1, 2, 3, 4\}$) where i is the cardinality of its capability set.

Having described the task and agent capability models that agents develop, we show in the next section how these models are utilised by agents in their decision making processes.

4 Cognition-based Decision Strategies

In this section we formalise the agent decision strategies that are designed by incorporating the cognitive models discussed in Section 3. However, we note that our formulations are but one possible way of characterising the behaviour of agents in

[1] We believe this is reasonable and does not affect our results and conclusions in a significant way.

[2] We assume that an agent does not attempt to reveal false information.

Table 1: Probability distribution of task types over time

	Task types *(exp,res)*			
Time (t%F)	1 (e=1,r=1)	2 (e=1,r=2)	3 (e=2,r=1)	4 (e=2,r=2)
1	0.4	0.3	0.2	0.1
2	0.1	0.4	0.3	0.2
3	0.2	0.1	0.4	0.3
4	0.3	0.2	0.1	0.4

this context. The parameters that agents take into consideration for building their decision heuristics and the way they are combined to generate formal behavioural specifications are not claimed to be unique. Moreover, we do not claim they are optimal. Instead, our goal is to lay out a generic behavioural model motivated by cognitive capabilities.

In Section 3, we have identified two behavioural roles that an agent can adapt to — that of a service requester and a provider. As a requester, an agent can either:

- **request assistance** from a neighbour (see Section 4.1), or,
- **request referral** with the motivation of finding a new neighbour from whom it can gain improved quality of assistance on task types in which it lacks expertise (see Section 4.2), or,
- **accept or deny a referral** depending on whether it already knows the acquaintance or it was reneged on earlier (Section 4.3).

As a service provider an agent can either decide to,

- **grant or deny service** assistance (Section 4.4), or,
- **grant or deny referral** request (Section 4.5).

The decision to **renege on an acquaintance**, formalised in Section 4.6, can be taken by any agent and is not an action that corresponds to either the requester or the provider roles. In the following discussion, each of the above actions are elaborated and the influence of the cognitive models on action selection explained.

4.1 Requester Action: Request Assistance

Agent i requests its neighbour j to execute a task on its behalf when its own expertise and resources do not match with the task's requirements. In more detail, i computes the "likelihood" of receiving help from each of its neighbours and polls a neighbour (j) with a probability in proportion to the computed likelihood for that neighbour. To this end, we have incorporated the following factors in the decision process of i. First, i assesses how much the requirements of its task *match* with the capabilities of j. It believes that if j has the matching capability for a task, then the likelihood of getting assistance increases because j is more likely to help for a task it has expertise on and hence, incurs less cost (relation between capability match and task cost is explained later). Second, from its previous interactions with j, i computes the number of times it was helped by j as opposed to it helped j or was denied assistance by j. From these numbers, it compares the frequency with which it received benefit in the past from j with the frequency of incurring cost due to j. The higher the frequency of the first type compared to the latter, the greater is the likelihood of getting help from j. Lastly, in addition to computing how many times j had assisted in the past, i computes *how much* j had been "useful" in past interactions by helping i, allowing i to save cost (discussions on task cost later). The more j has been useful in the past, the greater is the likelihood that i would get assistance again from j. Intuitively, the last two factors help i use its past experiences in deriving the chances of receiving help from a neighbour. Now, we present formal definitions of each of these factors and a logical way of putting them together for the agents to compute the likelihood metric.

Match: For a task of type x, requiring expertise and resource types e_x and r_x respectively, an agent's capabilities "match" with the task requirements if it has both e_x and r_x in its capability set. We say the "match" value is "high" in this case and "low", otherwise. To compute the match ($match_i(x,j)$) of the requirements of task type x with the capabilities of j, i uses its own estimates (E) of j's expertise and resource capabilities, updated using reinforcement learning techniques [Sutton and Burto, 1998]: $E^x_{t+1} \leftarrow (1-\alpha)E^x_t + \alpha R$, where t refers to the interaction number, x is the task type, α is the learning rate and R is the immediate reward (the value of resource or expertise with which the helping agent executes the task). When j assists i for task type x, the latter updates its estimates (E^x) of j's capabilities for that task type. Since a task is composed of both an expertise and a resource type, separate estimates of both aspects are kept to match an acquaintance's capability with a given task. In our formulation, we say that a match is "high" (=5) if both the estimated expertise and resource values are close to the required values of the task (within a permissible error, δ (=10) %), and "low" (=1), otherwise. A learning rate of 0.5 is used.

Interaction frequency: "Beneficial" interaction frequency for i with j is the ratio of the number of times i received assistance (the benefactor) from j to the total number of interactions between the two. Analogously, i's "non-beneficial" interaction frequency with j is the ratio of the number of times i helped j and was denied assistance by j to the total number of interactions. At time t, i computes time-discounted frequencies from its interaction record with j over a finite past time period (P):

$$fr_i^{cat}(j,t) \leftarrow \frac{\sum_{T=1}^{P} \gamma^T |cond(i,j,T)}{|interactions|_1^P}.$$

The boolean variable $cond(i,j,T)$ is "true" if i and j interacted at time T. Here, cat can be one of the two frequency types, beneficial or non-beneficial. T corresponds to absolute time $(t-T)$ in the past. $|interactions|_1^P$ denotes the total number of interactions between i and j in a time of P units in the past and γ is the discount factor (a value of 0.9 is used). For beneficial frequency, i computes the numerator using interactions where it was the benefactor. For non-beneficial frequency, interactions where i either helped or was denied assistance from j are used. A value of $P = 10$ is used which is sufficient to allow i consider interactions with j for all task types (since task types repeat every 4 time instances).

Rating: Utility earned and cost incurred from j are represented in i's ratings of j. i ups its "beneficial" rating of j at that time by an amount equal to the cost [3] i saved, if either i receives help from j or from another agent that j referred to i. However, if i serves j's request, it reduces the "non-beneficial" rating of j at that time by an amount equal to the cost incurred. Alternatively, if i is denied assistance by j, i reduces its non-beneficial rating of j by a fixed positive "penalty" (=0.5). Ratings are maintained for interactions over a finite past time period P. To compute the likelihood metric, i calculates at time t the cumulative, time-discounted ratings of its neighbour j,

$$C\_rating_i^{cat}(j,t) \leftarrow \sum_{T=1}^{P} x_T \gamma^T,$$

where x_T denotes a saving at time T in the past in case of beneficial rating (cat = beneficial), or a cost incurred at time T in the past in case of non-beneficial rating (cat = non-beneficial). We note that all agents initially have zero values for both types of ratings of their neighbours. The beneficial ratings (and hence, the cumulative beneficial rating) are, therefore, positive real numbers. Similarly, non-beneficial ratings (thus, cumulative non-beneficial rating) are negative real numbers.

For i to compute at time t, the likelihood of receiving help from j on a task of type x, we develop the following method.

$$likelihood_i(j,t) \leftarrow (match_i(x,j) + C\_rating_i^{+}(j,t) * fr_i^{+}(j,t)) + (C\_rating_i^{-}(j,t) * fr_i^{-}(j,t)).$$

Here, "+" superscript is used for beneficial frequency and rating categories, "-" for non-beneficial. This measure ensures that the the higher the match of the current task with i's estimates of j's capabilities (or, lower the estimated task cost for j) and the savings earned from j (note: $C\_rating^{+} \geq 0$), and the lower the cost incurred due to j in the past ($C\_rating^{-} \leq 0$), the greater is the likelihood of receiving help from j.

An agent i uses a standard Boltzmann exploration function to compute the probability of asking neighbour j at time t :

$$Pr_i^{j,t}(ask) \leftarrow \frac{\exp(\frac{likelihood_i(j,t)}{K})}{\sum_{k \in neighbours_i} \exp(\frac{likelihood_i(k,t)}{K})}, \quad (1)$$

where, K is an exploration constant with value 0.8 and $neighbours_i$ refers to the current set of neighbours of i. Equation 1 guarantees that the probability of asking help from a neighbour increases with increasing value of the $likelihood$ metric of that neighbour and vice versa.

[3] We assume that the cost of doing a task is "high" (=5) if an agent lacks matching capability for the task. "High" cost is used as an equivalence to the agent's inability to complete that task; cost is "low" (=1), otherwise.

4.2 Requester Action: Request Referral

An agent seeks a new acquaintance if services on task types for which it lacks expertise are not available from existing acquaintances. In more detail, each agent periodically checks, using its own learned models of neighbour capabilities, whether its current neighbours have the matching expertise and resource capabilities for all of its task requirements. We say the "state" of an agent is "fit" if it has at least one matching neighbour for each of the task types it requires assistance for. This implies that the agent can potentially receive service from its neighbours on all task types in which it lacks the abilities. If there is at least one task type in which it lacks the required capabilities and does not have a neighbour with those capabilities, its "state" is "not fit". An agent with state "not fit" requests a randomly chosen neighbour to reveal one of the latter's existing neighbours. The motivation for seeking a new neighbour is to increase the chance of locating an agent with the required capabilities. We observe that precise measurement of an agent's state is possible by having accurate models of its neighbour capabilities. Such accurate information would help an agent to take decisions that are relevant, rather than random referral requests that incur unnecessary communication costs.

4.3 Requester Action: Accept/Deny Referral

An agent i accepts a referral a only if the following conditions hold for both i and a: (a) either did not have the other agent as a neighbour before (and reneged in the past), and (b) the current neighbourhood of both contains less than the maximum number of neighbours an agent can have at any time.

4.4 Provider Action: Grant/Deny Assistance

An agent i while deciding whether or not to acknowledge a service request from agent j considers the following factors. First, i evaluates the cost it will incur to execute the task for which j requests assistance. The greater the cost incurred by i, the less likely it would be to help. Second, similar to the decision process in Section 4.1, the provider i also uses its assessments of previous interactions with j (the "interaction frequency" and "rating") to evaluate the probability of granting service to the provider. Thus, the greater the frequency of beneficial interactions in the past and the more utility earned from j, the higher is the chance of helping j. In addition to the above, i also looks ahead into its own future task requirements and computes how likely is j to help on those. Specifically, i's probability of helping j increases if it evaluates j as a prospective future helper by comparing j's capabilities with its own future task requirements. To this end, we note that agents require precise models of both the task environment and of capabilities of other agents. In the following, our formulations of the above factors are described together with the method of combining them into a decision strategy.

Task cost (C_i^x) that agent i would incur to execute a task of type x. If it has the desired expertise and resource capabilities as demanded by the task, it incurs a "low" cost, otherwise, a "high" cost (similar to the previous discussion on capability match).

Interaction frequency with j. The "beneficial" and "non-beneficial" interaction frequency computations are the same as discussed in Section 4.1.

Rating of i for j. "Beneficial" and "non-beneficial" rating calculations are similar to what have been defined in Section 4.1.

Future expected usefulness: Since i has the knowledge of how the task types vary with time (see Table 1) and it has the estimates of j's expertise and resource capabilities for different task types (developed via learning during interactions with j), it can compute the match of j's capabilities with its future service requirements. Thus, i computes the future prospect of j at time t as:

$$pro_i(j,t) \leftarrow \frac{\sum_{T=1}^{F} \sum_{x=1}^{M} match_i(x,j) * Pr(x,T^{'})}{|F|}$$

Here, T counts F time instants (F is the periodicity of the task environment). $Pr(x,T^{'})$, where $T^{'} = (t + T)\%F$, is the probability of task type x occurring at time $(t+T)$ in the future (corresponds to (time = $T^{'}$,task type = x) in Table 1). M is the total number of task types. ¿From Table 1, we see that the values of F and M are both four.

Agent i uses a sigmoidal probability function,

$$Pr \leftarrow \frac{1}{1 + \exp(\frac{V - Th}{\tau})}, \tag{2}$$

to determine the probability (Pr) of accepting j's service request. This form is inspired from the Fermi function and has been used in settings [Sen, 2002; Jin *et al.*, 2001] where the probability of choosing an action is thresholded around some value Th of a control variable V. The probability takes a value of 0.5 when the control variable is equal to the threshold. The transition of the probability function around the point $V = Th$ can be made more or less skewed by adjusting the shaping parameter τ. We propose a novel method that agent i uses to compute V.

$$\begin{aligned} V \quad \leftarrow \quad & (C_i^x - C\_rating_i^-(j,t) * fr_i^-(j,t)) \\ & -(pro_i(j,t) + C\_rating_i^+(j,t) * fr_i^+(j,t)). \end{aligned}$$

Such a formulation ensures that the probability of providing service to j increases with:

- lower task costs for i,
- lower cost incurred from interactions with j in the past (lower non-beneficial rating and frequency values),
- higher prospect of j as future service provider, and,
- higher savings earned from interactions with j in the past (higher beneficial rating and frequency values).

We have used a τ value of 1 and set $Th = 0$. We note that the terms in the first pair of parentheses are comparable to those in the second pair. Also, the provider agent has a small positive helping probability when it is completely neutral (i.e., when it has no cognitive models of either the environment or other agents' capabilities and the rating and frequency terms in the control variable all have values of zero).

As more interactions occur agent i can improve its cognitive models of j's capabilities and, hence, can compute the above probability function in a more precise manner.

4.5 Provider Action: Grant/Deny Referral

First, we summarize agent i's deliberations as they relate to deciding to provide a reference of an agent a to requesting agent j. Being self-interested, i reveals a referral only if that action earns it some utility. Therefore, the first condition that i verifies is whether it has any dependence on the requester j [4]. In case it does not, its rational choice is to ignore j's request: since it does not foresee any service assistance from j, it is not inclined to acknowledge the latter's requests. In case i depends on j, it then considers whether j has a dependence on the prospective reference a. It evaluates this by comparing the requirement estimates of j with the capability estimates of a. The reason being, if j has a dependence on a, then it is likely to receive assistance from a and thus, assign the referring agent (i, in this case) a positive beneficial rating (see discussion on **rating** in Section 4.1). This, in turn, would assist i in obtaining future assistance from j. If i detects no dependence of j on a, it does not refer a because in so doing it would not earn any future expected utility (positive rating) from j. On the contrary, if i confirms that the above two dependences exist, it subsequently computes if there is a chance of losing utility by getting reneged on by either a or j or both. Agents renege on interactions with those acquaintances that fail to serve their requests or are less capable of service compared to other acquaintances. The latter situation can arise if an agent gets referred to a new acquaintance having high match and thus, reneges on some of its previous acquaintances. So, i checks whether the condition arises where either a or j, being new acquaintances, might stop interacting with it. The above factors are evaluated sequentially by the deciding agent i, and are enumerated in the following.

- i depends on j: As described before, i considers referring only if it estimates a dependence of its task requirements with the capabilities of j.
- j depends on a: i considers referring if it estimates that j depends on a's expertise. The motivation is (stated earlier), i earns positive rating from j every time the reference a helps j, which allows i to receive help from j in the future. Thus, i aims to maximize the utility earned (rating from j) from its action of referring a to j.
- i does not lose j as neighbour by referring a to j: An agent m reneges on a neighbour n if m does not receive service from it. This can happen under two circumstances: (1) n does not have the capabilities to serve m on tasks in which m requires assistance, or, (2) m has too many other neighbours on whom its dependences are much stronger than on n and hence, m does not interact with n often enough to be able to detect the latter as a beneficiary.

  Since i considers referring only if it depends on j, it ensures that the referral would not engender the latter of

[4] The "dependence" of an agent x on another agent y consists of the set of task types in which x's own requirements match with y's capabilities as estimated by x.

the above two situations where j gets a more beneficial neighbour a and stops interactions with i. It achieves this by referring only if, (a) j's dependence on a is a proper subset of j's dependence on i or, (b) j's dependence on a is completely uncorrelated with j's dependence on i. In either situations, j would not potentially renege on i after being referred agent a.

- If i detects that j depends on a and j does not renege on i as a consequence of referral, then it refers a to j if it estimates that it does not depend on a. Otherwise, it considers the following.
- Since i evaluates dependence on a, it can ensure it is not getting reneged on by a in the future following similar reasons as applied to the case of not losing j. Therefore, i refers a to j if, (a) a's dependence on j is a proper subset of a's dependence on i or, (b) a's dependence on j is completely uncorrelated with a's dependence on i. The reason being similar to that described in the third criterion above.

4.6 Reneging on Interactions with a Neighbour

Interacting with a neighbour involves computation costs to evaluate and update their behavioural models and communication overhead. Hence, a selfish agent decides to revoke interactions with those acquaintances that fail to provide its required services. The following discussion formally establishes an agent's decision to renege on a neighbour.

In our formulation, an agent i maintains a record of the number of times it has requested assistance from a neighbour j for each of the task types in which it lacks expertise. The higher the number of times such requests failed, the higher is the likelihood that i would renege on interacting with j. We let our agents use the decay function $\exp(-x_t * \beta)$ to update the current "strength" of the relationship between i and j (this form of the equation is inspired from the work on evolution of social networks [Jin *et al.*, 2001]). Here, x_t is the number of failed requests made by i to j for tasks of type t and β is the decay factor. Whenever i receives help from j on type t, it sets the decay function to 1. An agent revokes interactions with its neighbour if the decay function falls below a threshold (of 0.3) for all task types that it requires assistance for. Thus, by adjusting the value of β, the agents can have different punishing tendencies for neighbours who do not assist — in the current formulation, we select $\beta = 0.1204$ which implies that an agent i would renege on j if j fails to help i on 10 successive requests (since $\exp(-10 * 0.1204) \approx 0.3$) on tasks of each type in which i lacks expertise.

In the following section we summarily describe the simulation of agent interactions where the above decision strategies are used by agents.

5 Simulation and Interaction Dynamics

Our domain simulations can be summarized in the following steps.

- Agents are assigned their capability sets and the initial random neighbours.
- While simulation continues for N time steps, at each time step:

```
- K tasks assigned to each of a random set of agents.
- While all tasks are not completed (each task is
  assumed to take one time unit for completion):
   - All agents who do not have required expertise
     and resource for assigned tasks, ask for help.
     - The provider updates estimates of the
       requester's expertise and resource requirements.
     - Provider decides to help/not help.
     - Requester updates capability model of provider.
     - Both requester and provider assigns ratings to
       the other.
- All agents evaluate ''state'' and decide to request
  a referral
  - Provider takes referral decision.
  - Requester accepts / rejects referral, if any.
-  All agents check for possible reneging on neighbours.
```

6 Experimental Results

Our goal is to study whether and to what extent can agents locate partners with matching capabilities in the specified domain (Section 3) using the decision heuristics described in Section 4. These observations are compared with those where agents adopt random decision strategies as opposed to using cognition-based strategies. The following metrics are chosen for comparison.

Agent state: In the agent groups that emerge out of the simulations, we measure the proportion of an agent's expertise and resource requirements that match with its neighbours' (group members) capabilities. We compute the "state" of an agent i, $\frac{\sum_{x \in required_i} 1|cond_i(x)}{required_i}$, where $required_i$ is the set of task types in which i lacks expertise and resource, and the boolean variable $cond_i(x)$ is $true$ if there is at least one agent in i's neighbours that i estimates to have the capability for task type x. Thus, the "state" value is 1 for an agent having a neighbour with the matching capabilities for each task type in which it lacks required capabilities. Hence, the closer the value of an agent's "state" is to 1, the more successful (better) it has been in locating complementary partners.

Group size and composition: We measure the agent "connectedness" in the emerged groups. The metrics of interest are (a) average size of the group (or, number of neighbours) of an agent of each capability type, and (b) average number of agents of different capability types (group composition) in groups of each agent type. Group size measurements help analyse how the connectedness of agents depend on the constraint of having a limited number of acquaintances. Also, it correlates group size with "state" values of agents. Group composition analysis shows any relation that exists between the size of partnerships of different agent types and their capabilities.

In the following subsections, we report experimental results and explain our observations. A total of 50 agents are used in all experiments. Only agent types 1 and 3 are used (Section 3 explains agent types). These agent types differ in the size of their capability sets (type 1 has capability in one task type while type 3 has capabilities in three task types). This allows us to study the effects on the results, if any, due to heterogeneous capability sets of agents. The exact task types in which an agent has expertise and resource are assigned randomly in the initialization routine. At the start of a simulation, agents are connected randomly to one another

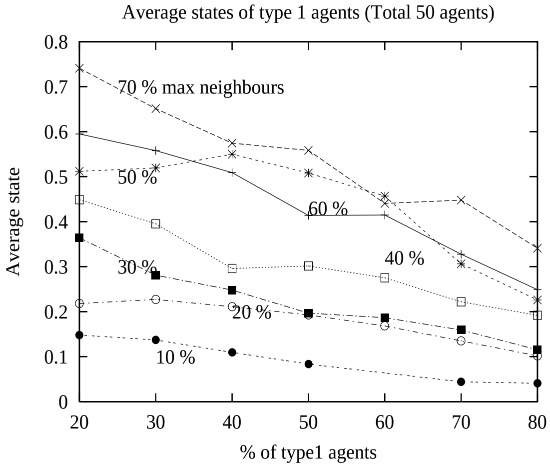

Figure 1: Agent type 1 "states"

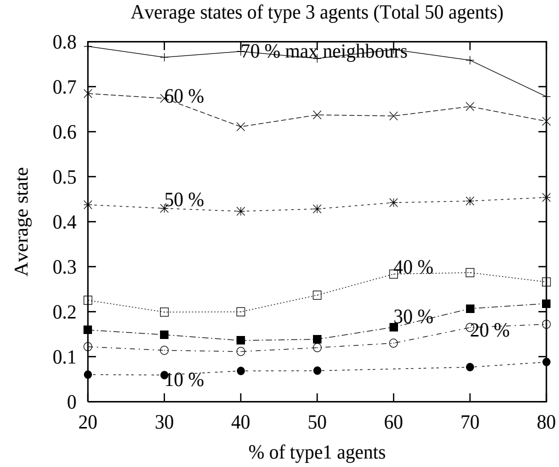

Figure 2: Agent type 3 "states"

with the number of connections for each agent bounded by the maximum allowable neighbour size. For a given random initial configuration, each simulation is repeated for 10 runs. A simulation run spans 1500 time steps (details of one time step is described in Section 5). All results reported are averaged over experiments conducted for 10 such random initial configurations, hence, over a total of 100 runs.

6.1 Agent States

We measured agent states by increasing the maximum number of neighbours an agent can have (allowable neighbour size), from 10% to 70% in steps of 10. A maximum neighbour size of x% means that an agent can interact with (or, have as neighbours) at most x% of the total agent population. For each value of allowable neighbours, the proportions of type 1 and type 3 agents were varied from $(20\%, 80\%)$ to $(80\%, 20\%)$ in steps of 10%. Figures 1 and 2 show the variation of agent states with changing composition of type 1 and type 3 agents, respectively, for different values of maximum neighbours.

Observation 1: *Agent states improve with increase in the number of allowable neighbours.*
In both Figures 1 and 2, we note that, with increase in the maximum neighbour size, the agents attain better state values. With maximum neighbour size = 70%, a type 3 agent gets a state value of almost 0.8, which indicates a very close match of its requirements and the capabilities of its neighbours. This indicates that with more available neighbours to interact with, agents using cognition-based decision strategies are able to locate better matching partners. The state values of agents using random decisions are found to be always zero. These agents do not interact with others using strategies similar to the cognitive agents. Hence, they fail to recognise helpful partners. This in turn leads to their having poor (zero) state values.

However, we observed that increasing the neighbour size, the communication cost increases. We recorded the total number of requests (graph not provided) an agent makes for all service requirements over a simulation run. From this, we found that the average number of requests made in one simulation run increased almost linearly from about 2500 to close to 22500 as the neighbour size was increased from 10% to 70% with a total 50 agents and with 50% of each agent type (1 and 3).

Observation 2: *The state of type 1 agents decreases with increase in their numbers, while that of type 3 agents remains unchanged.*
For the same maximum neighbour size, a decrease in the state value is observed for type 1 agents, as their proportion in the population increases (see Figure 1). The state values of type 3 agents, on the contrary, remain almost unaffected by the changing composition (Figure 2). Type 1 agents require assistance for 3 different task types in which they lack expertise and resource. To attain a high state value, they require assistance in most of the required task types. But, agents receive cooperation only if they can reciprocate (see decision strategy in Section 4.4). Type 1 agents have scarce resource and expertise to be potential service providers. However, when they are a minority in the population compared to type 3 agents, they can find a few type 3 agents that contribute significantly to improve their state values (note graph when type 1's are minority). On the contrary, as type 1 increases in the population, they are able to form more partnerships with agents of their own type than with type 3 (result in Section 6.2 shows this). A coalition with its own type is not as beneficial for a type 1 agent as is that with type 3. In a type1 - type1 partnership, only 1 of its 3 required capabilities are satisfied compared to all 3 in a type1 - type3 partnership. Hence, type 1 agent states decrease with an increase in their numbers. Type 3 agents, however, require assistance for only one task type and can be potential cooperators in several task types. These factors enable them to find cooperating neighbours (mainly of its own type) and thus, maintain a stable state value independent of their number in the population.

6.2 Group Characteristics

In addition to studying how well are the agents, using cognitive decision strategies, able to locate matching partners, we are interested to know how many agents and of what type form groups.

Observation 3: *Average group size of an agent type increases with increase in the number of agents of that type.* Table 2 shows the average group size of type 1 and type 3 agents. Simulations were run using a moderate value (30%) of the maximum neighbour size and changing the proportion of the agent types in the population. It is observed that with an increase in their numbers, both agent types form larger groups. This implies that the connectedness of an agent is directly related to the number of agents of its type in the population. However, we observe a negative correlation between the group size and the state values of type 1 agents. As a majority in the population, they form larger groups but have lower state values (see Figure 1) than when they are less in number. With an increase in number, type 1 agents form groups with more agents of their own type than type 3 (see discussion on group composition), which does not help to increase the states of these agents (explained previously in state results). Table 2 also reveals that type 1 agents have slightly larger groups than an equal proportion of type 3 agents. This is due to the difference in the sizes of capability sets of the two agent types. A type 1 agent i has capability in only one task type. It can, hence, find several other type 1 agents that have one matching capability type out of the 3 that it requires and can therefore maintain partnerships through mutual cooperation. Thus, although a type 1 - type 1 partnership does not result in a significant state improvement of the agents forming the partnership, they are more in number than type 3 - type 3 partnerships. We believe that the differences in group sizes would be more pronounced if the capability types in which agents differ increase in number, i.e, with more heterogeneity in agent types.

Observation 4: *In a population of agents with heterogeneous capability sets, larger partnerships form between agents of the same type.* Table 3 summarizes how many type 1 and type 3 agents are present on an average in groups of an average type 1 and an average type 3 agent. It shows that an agent forms more coalitions with agents of its own type than with others. The size of partnerships between similar agents increases with an increase in their numbers. This skewness about which agent type to make partnerships with is because partnerships are based on utility-generating interactions: a type 3 agent (that has a greater set of capabilities than a type 1) is more likely to revoke a type 1 partner (which has exactly 1 capability type) from its neighbourhood when it recognises the poorer cooperation capability of the latter compared to the agents of its own type [5]. However, we see type 3 agents have more type 1 partners when the number of the latter type becomes dominant in the population (Table 3: row corresponding to $(70, 30)$ and column of group composition of agent type 3). A type 1 agent develops partnerships with other type 1 agents that have its missing capability type. Such partnerships increase with increase in the number of type 1 in the population.

We note that this behaviour of agents to preferentially make partners requires accurate agent models, developed using cognition-supported decision mechanisms. Agents designed with random decision strategies quickly renege on all neighbours. Since they do not employ observation-based modelling of agent capabilities, they are unable to detect the benefit of interacting with any agent. Thus, measurement of group characteristics is not realisable for such agents.

[5] It is unlikely that a type 1 agent has the exact matching capability that a type 3 requires.

Table 2: Average group size (50 agents, max neighbour 30%)

(type1 %, type3 %)	*Type 1 groups*	*Type 3 groups*
(30,70)	3.38	6.04
(40,60)	5.03	5.68
(50,50)	5.73	5.65
(60,40)	6.85	4.54
(70,30)	7.08	4.31

Table 3: Average group composition (50 agents, max neighbour 30%)

	Type 1 groups		Type 3 groups	
(type1 %, type3 %)	Type 1	Type 3	Type 1	Type 3
(30,70)	1.88	1.95	0.8	5.2
(40,60)	3.34	1.69	1.12	4.6
(50,50)	4.2	1.5	1.5	4.12
(60,40)	5.9	1	1.5	3
(70,30)	6	0.9	2.1	2.2

6.3 Reneging and Referral Patterns

In addition to computing agent state values and their connectedness at the end of the simulations, we are interested in studying their behaviour in course of the simulation. Specifically, we note how their decisions to renege on neighbours and ask or provide referrals vary over time.

Observation 5: *Frequency of reneging neighbours decreases and finally stops during the simulation. Referral requests continue while granting references stop over time.* Declining to interact with a neighbour or ask and/or grant referrals are important decisions that determine the final state and group structure of agents. In our experiments, we have observed a decay in the rate at which neighbours are reneged on with time. The reason being, even though agents do not always find partners with all matching capabilities, they are able to locate those with some matching capabilities to continue assisting each other and hence, maintain partnerships. The regularity in the task environment is an important influence in this context. With an erratic task environment, long lasting interactions that would allow agents to recognise the capabilities of each other for different task types simply would not have occurred.

We make an additional observation that agents continue referral requests during the entire simulation period. Relating to the decision mechanism described in Section 4.2, we infer that an agent continues requesting referrals since all of its task requirements are not satisfied from the available expertise of its current neighbours (it has not achieved a perfect state value of 1). However, revealing reference information ceases. This is because, since agents have neighbours from whom they receive assistance in one or more of their task requirements, referral becomes too "costly" — following the decision mechanism described in Section 4.5. This, in turn, means agents refrain from granting referrals because they evaluate a finite risk of losing neighbours by so doing.

6.4 Other Observations

It is observed that the learned values of neighbour expertise and resource estimates converge to their true values at the end of the simulation. Also, we make a note that in the decision related to asking for help (Section 4.1), a constant value is used for the exploration factor K. In our simulations, agent neighbourhoods are dynamic structures; hence, adjusting the exploration factor to encourage exploitation would lead to a complete lack of exploration of new neighbours obtained through referrals. Without sufficient interactions with new neighbours agents would fail to build accurate estimates of their capabilities which are necessary for effective decision making.

7 Conclusions and Future Work

In this paper, we have presented a set of novel decision mechanisms for a self-interested cognitive agent that interacts with other cognitive agents having different expertise types in a time-periodic task environment under some practical constraints. Using simulations, we have demonstrated that elements of cognition such as memory and learning help agents to successfully detect matching partners and form groups under such constraints. We also showed that, the utility earned from services of group members depends on capabilities of agents and the maximum allowable number of agents they can interact with. In this context, group size depends on the number of agents in the population. In addition, a given agent type is found to have a higher preference to form partnerships with others of its own type. That agents are able to receive service, and hence, earn utility, from group members of matching capabilities indicates the effectiveness of the decision strategies under the specified conditions.

Some of the interesting areas of future investigation are, (a) adding a random component to the task environment and designing decision strategies that help agents to appropriately adapt to such changes; (b) studying the effects of "multi-hop" referrals, where if a neighbour does not grant a referral request, it can pass it on for further evaluation to other neighbours; and, (c) applying and evaluating the decision mechanisms in problems such as multi-agent coordination using task delegation among self-interested agents.

References

[Axelrod, 1984] R. Axelrod. *The evolution of cooperation*. Basic Books, New York, 1984.

[Azoulay-Schwartz and Kraus, 2001] R. Azoulay-Schwartz and S. Kraus. Stable strategies for sharing information among agents. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1128–1134, 2001.

[Betchel and Graham, 1998] W. Betchel and G. Graham, editors. *A companion to congnitive science*. Blackwell, Oxford, UK, 1998.

[Castelfranchi and Falcone, 1998] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive autonomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS)*, pages 72–79, 1998.

[Forgas, 1991] J. P. Forgas, editor. *Emotion and social judgement*. Permagon Press, New York, 1991.

[Hogg and Jennings, 2001] L. M. Hogg and N. R. Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 31(5):381–399, 2001.

[Jennings, 2001] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

[Jin *et al.*, 2001] M. Jin, M. Girvan, and M. E. J. Newman. The structure of growing social networks. *Phys. Rev. E*, 64, 026118:381–399, 2001.

[Kautz *et al.*, 1997] H. Kautz, B. Selman, and M. Shah. Referralweb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[Lerman and Shehory, 2000] K. Lerman and O. Shehory. Coalition formation for large-scale electronic markets. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 167–174, 2000.

[Panzarasa *et al.*, 2001] P. Panzarasa, N. R. Jennings, and T. J. Norman. Social mental shaping: modelling the impact of sociality on autonomous agents' mental states. *Computational Intelligence*, 17(4):738–782, 2001.

[Sandholm and Crites, 1995] T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995. Special issue on the Prisoner's Dilemma.

[Sen, 2002] S. Sen. Believing others: pros and cons. *Artificial Intelligence*, 142:179–203, 2002.

[Sun, 2001] R. Sun. Cognitive science meets multi-agent systems: A prolegomenon. *Philosophical Psychology*, 14(1):5–28, 2001.

[Sutton and Burto, 1998] R. S. Sutton and A. G. Burto, editors. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[Yu *et al.*, 2003] B. Yu, M. Venkatraman, and M. P. Singh. An adaptive social network for information access: architecture and experimental results. *Applied Artificial Intelligence*, 17(1):21–38, January 2003.