Trust Values for Agent Selection in Multiagent Systems

Karmelo Urzelai

Abstract

A key step of any distributed system, based on the negotiation paradigm, is the selection of an agent to whom offer a contract to solve a problem. The agents must have social knowledge that may contain a list of acquaintances. If more than one agent match the requirements needed to solve a problem, a conflict resolution criterion is necessary. The paper begins describing the agent's structure, with the knowledge that intervenes in the selection process. Afterwards, some strategies for agent selection, based on knowledge acquired during previous contracts, are presented, and a generic model is proposed to support them. The adaptation of the proposed model to societies with different features is discussed too. Finally, some extensions for the agent's selection criteria are explained. The proposed solution is simple and efficient.

1 Introduction

A multiagent system (MAS) is not only a set of knowledge based systems working at a time, there must be some kind of communication and coordination among them.

A MAS [Sichman *et al.,* 92] is concerned with the activity of autonomous agents in a common environment that interact for problem solving. Important points in MAS, with cognitive agents, are:

- Agents are autonomous
- MAS are open systems
- Agents decompose the tasks.
- They are able to solve others' problems
- The environment may change

An interesting perspective is the negotiation paradigm [Davis and Smith, 88], where agents, with problems too complex to be solved by a single agent, decompose them into subproblems and negotiate with other agents for the task allocation. The agents in charge of the subproblem solving, can decompose them again and start new negotiation processes to distribute the smaller subproblems.

A well known model of such mechanism is the Contract Net Protocol (CNET) [Smith, 88]. This model offers a framework to make heterogeneous agents cooperate with a distributed control. Problem resolution consists of four steps: problem decomposition, subproblem distribution, resolution and, finally, solution synthesis. The agents solve problems negotiating to find the best node suited to solve the tasks.

A key step in the task distribution process is the selection of an agent to whom offer a task. Given an unsolved problem, an agent can know more than one agent that can solve it. Therefore, there must be a conflict resolution process to select the most suitable agent for the problem.

The election of a control strategy for conflict resolution is a critical decision. Some hints of its importance are:

- The resulting solution can be more accurate to the problem needs or not, depending on the selection strategy.
- The system can arrive earlier to a solution if an efficient agent is contracted, and later with a lazy agent.
- A good agent selector can recognize a reliable agent, and avoid others that usually fail.

The conflict resolution depends on the available information about the acquaintances. The uncertainty decreases with the information, so, with more information better decisions can be taken.

This information is acquired with the experience, and therefore, as the agent works and offers contracts, it acquires more experience and accumulates more information about other agents, and, therefore, it will be able to make more accurate selections.

The paper proposes the use of trust values [Marsh, 94] calculated for the acquaintances, to select addresses for contract offers. A function that use previous experience is offered, that is simple and very easy to implement.

2 Environment description

Let's start defining the basic elements of the agents, and the components that take part in the task of selecting an addressee for contract offer.

The society is composed of a set of agents $S=\{A_1, A_2, ..., A_r\}$, being each agent defined as a tuple <Id, SK, PSK, AK, Pb, T> where:

- Id is the agent's identification.
- SK is the Self-Knowledge; the knowledge each agent has about itself.
- PSK is the Problem Solving Knowledge; the knowledge the agent has to solve problems.

- $AK = \{Ak_1, Ak_2, ..., Ak_m\}$ is the Acquaintance Knowledge; the knowledge that the agent has about other agents' features. It is a set of agents such that:

$$\forall\ Ak_i \in AK, \exists\ A_j \in S \ \Big|\ Ak_i \ \textit{informs\_about}\ A_j$$

- Pb is the information about the problem instances to be solved by the agent.
- T are the tasks associated to the problems being solved.

This paper focus in the Acquaintance Knowledge, that will be used to select the best agent to whom offer a contract to solve a problem. The list of acquaintances can be initially empty and be enriched with the contracts offered by the agent; for example with the answers to a broadcast query searching agents able to solve a problem.

The information about each acquaintance is stored in a tuple <Id, Lc, PC> where:

- Id identifies the acquaintance.
- Lc is the acquaintance location
- $PC = \{Pc_1, Pc_2, ..., Pc_q\}$ is a set that contains the information about the problem classes solved by the acquaintance.

And PC_i ($\forall\ i \in \{1,..., q\}$) is a tuple <Id, Ev> where:

- Id is an identification for the Problem Class. It ought to belong to an ontology of problem classes, to avoid consistency problems in the problems' recognition.
- $Ev = (Ev_1, Ev_2, ..., Ev_n)$ is an ordered vector containing evaluations of the contracts offered by the agent to the acquaintance, to solve problems of the specified class.

The evaluations measure how much the results returned by the contracted agent validates the problems' expectation. So, it is a value that suggests how satisfied was the contractor with the work done by the contracted agent, in a problem instance.

We will suppose that the evaluations are in chronological order, beginning with the oldest one; therefore Ev_n is the evaluation of the results returned by the corresponding acquaintance, in the last problem contracted, of the associated problem class.

The evaluation values will range between -1.0 for a problem contracted and failed, to 1.0, for a problem correctly solved. Any value inside the interval will be valid, with positive values meaning satisfactory results, and negative values for the unsatisfactory ones.

The agents' acquaintance knowledge can be initially empty. The agent can acquire this information during its interactions with the society. The knowledge enrichment can be in three ways:

- Acquaintances: The set of known agents increases with the answer to a broadcast searching agents, or with messages of agents announcing their availability. For example, the Contract Net's bid message.
- Problem classes solved by the acquaintances: known agents can send bids for problem classes that the receiver did not know they solved. Thus, the list of problem classes solved by the acquaintance can be extended.
- Acquaintances' evaluation list: whenever a contract finishes, the returned solution is compared with its expectation and the resulting evaluation is added to the evaluations vector of the corresponding problem class of the contracted agent.

This knowledge is particular of each agent, there is neither a global knowledge, nor an agent in charge to inform the others. So that, each agent has a partial perspective of the others, that depends on its intercourses with them.

3 Agent Selection

When an agent wants to offer a contract for a problem solving, either because it cannot solve the problem or because it is too busy, it must select an addressee.

It looks into its Acquaintance Knowledge for information about agents capable to solve problems of the specified class. If it knows more than one suitable agent, it needs a conflict resolution policy to select the best agent to whom offer the problem.

The assignation of a trust value or confidence factor to each agent that can solve the problem, could guide the selection.

Let T(x) be the function that calculates the trust value for an acquaintance x, and m the number of acquaintances that deal with the problem to be solved, the selection criterion would be:

$$Ak_i \mid T(Ak_i) \geq T(Ak_j) \ (\forall\, j \in \{1,\ldots, m\})$$

Let's analyze what humans usually do to contract someone to solve a problem. For example, every time someone has toothache, he does not consult the yellow pages to go to any dentist. People usually know some dentists, and the first time they must go to them, they select the one with the best references. From then onwards, if the dentist worked well enough, patients usually repeat going to the same dentist, and do not try a new one. But, if after some visits the patient disagrees with the results, he goes to another one.

We will implement the same strategy for agent selection in a society of distributed agents, using the agent's knowledge previously defined.

A first approach is a conservative policy assigning the trust value according to the full **history** of the acquaintance. The function to calculate a trust value for an agent Ak_i, supposing that n is the number of evaluations stored for it, would be:

$$T(Ak_i) = \sum_{j=1}^{n} Ev_j$$

This function prefers acquaintances with many small positive evaluations, rather than others with few evaluations, despite they are better; and it discards acquaintances that have failed many times. The agent can try several acquaintances until it finds a reliable one.

So, the selection policy is based on the information credibility. Thus, if the agent knows someone that, up to now, has solved the problems rather well, it is preferred over someone else, that, although the new one could be better, there is less information about it, and therefore more uncertainty, about its capabilities and trustworthiness.

A second trust function, based on the history too but less conservative, consists on the average evaluation value.

$$T(Ak_i) = \frac{\sum_{j=1}^{n} Ev_j}{n}$$

In this case, the agent selection is more objective, and the new agents can be integrated faster. On the other hand, an untrustworthy agent without a regular performance can be selected based on a single lucky previous resolution.

Sometimes, the agents can grow worse, (because they get old, or simply degenerate due to non updated information), then the evaluation values acquired long time ago are not significant of the real agent's capability. So, a second approach is based on **recent history**. The trust function should be restricted to a set formed by only the last contracts solved by the agent. Let's call α to the maximum number of evaluations to consider for each acquaintance. The trust function could be:

$$T(Ak_i) = \sum_{j=n-\alpha}^{n} Ev_j$$

As explained before, this function can be also modified to use the average evaluation of the last α agents.

Finally, the third approach is a **pondered mean**, giving different weights to each contract resolution's evaluation, according to how recent they are. The weight function will have more or less slope according to the greater or smaller importance of the most recent evaluations.

For example, the simplest weight function is the identity function, $\omega(j)=j$, i.e., to multiply each evaluation by its order number.

Now, we can define a general trust function, T(x), for each acquaintance, Ak_i, of the conflict set of the available agents for a problem contract, as:

$$T(Ak_i) = \sum_{j=1}^{n} Ev_j * \omega(n-j)$$

Where ω (x) is the weight function, that for an evaluation's ordinal number returns its weight over the final trust value. Its argument is *n-j* to facilitate the control of the weights assigned to the evaluation values, according to the chronology; beginning with the newest, and finishing with the oldest.

The function has different values for each approach previously discuses (Figure 1):

- a constant value for the function based on the history.
- a constant value for the parameters lower than a threshold α, and 0 otherwise, for the recent history policy.
- a decreasing value as the parameter increase, until 0, for a pondered mean.

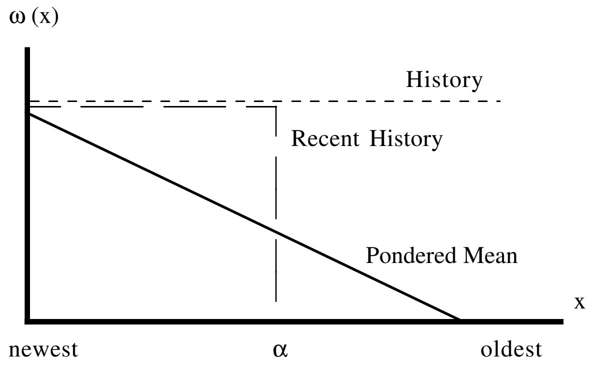

Figure 1: Weight function examples

4 Trust value settlement

The acquaintances that have never been contracted before need a default trust value to compete with the others in the selection process. Depending on the policy type, that value can be higher or lower.

The evaluation values rank between -1.0 (bad evaluation) and 1.0 (good evaluation), and therefore, an initial neutral value is 0. But the default value can be modified according to the society's objectives.

Depending on the interest to integrate new agents or not, the default trust value for agents that have never been contracted already, can be a little bigger or lower than 0.

The selection of the trust function depends on the society, the agents, and the social policy.

In a society, where the agents can become obsolete or can change very fast, a policy that analyzes more than the few most recent evaluations, can lead to out-of-date trust values, and therefore, to a bad selection, making the society not to work fitted. For those cases, the weight function must have a strong slope, giving more importance to the latest evaluations, or be as the one defined to analyze only the recent history (Figure 2).

On the other hand, if the society's agents are very stable, the trust values will be more accurate if they are calculated using more evaluations, and giving them similar weights (Figure 2).

If the agents are extremely steady, and the social policy is not conservative, the best trust function may be the calculation of the evaluation average.

For a conservative policy, where uncertainty can be dangerous, unknown agents are not desirable, therefore, they would have small negative values; e.g., -0.2

On the other hand, a policy that relies on new agents, and wants to integrate new agents should assign positive values for them, e.g., +0.2

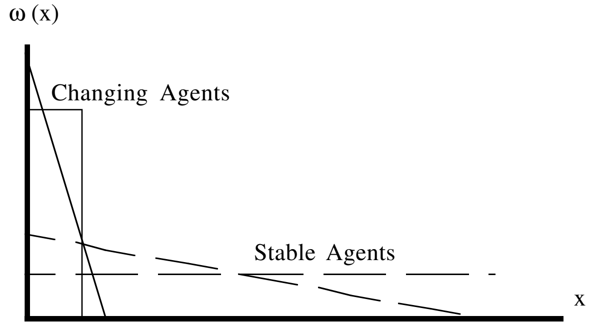

Figure 2: weight functions depending on the agent type

For the agents that have never been contracted for the pending problem class, but that there is information about its behavior in other problem classes, some uncertainty can be avoided. An approximate trust value can be calculated using the evaluation values of the other problem classes solved by the acquaintance. The resulting value can be increased or decreased, according to its credibility (as it has been explained for the default values of the unknown agents).

The proposed trust functions have the problem that they are calculated organizing the evaluations by their chronological order, but not taking into account the real time when they were done. So, the last contract evaluations of two acquaintances have the same weight, despite one has been recently done, and the other was done long time ago. But, in a changing society an agent not contracted for a long time would have similar uncertainty than an agent never contracted. That problem can be corrected associating a *time* value to each evaluation, that indicates the moment when it was done. Thus, the weight function, $\omega(x)$ could receive the time value as parameter, instead of an ordinal number.

5 Additional selection criteria

Besides the trust values, calculated from the evaluations of previous contracts, additional information can be useful for the selection of an addressee for a new contract.

An important datum for a good agent selection is the *cost*. A cost estimation can be included in the acquaintance tuples. The cost can be divided into the contract's cost, and the communication's cost. Depending on the problem type and the quantity of communication needs for it, the former or the later have more influence on the resulting value. For example, if there are budget restrictions, a near agent can be preferred to a better one, in an environment with an expensive communication net.

The *location* can be important in other cases too; as for example in noisy environments to avoid communication problems, or when the society is divided into geographic coalitions.

Another parameter definable for the acquaintances, to be used during the agent selection process, is the *efficiency*, based for example in the agent's speed. Its treatment can be similar to the one explained for the evaluations.

Additional social restrictions can be taken, as, not to offer many contracts to the same agent to avoid overload it with too much work, if the contracts are paid then to contract debtors, personal preferences due to a friendship, ...

All these criteria can be combined in a single general selection function, that associates different weights to each one, according to the importance of each criterion for each problem and environment.

A more interesting option, more complex but more powerful and flexible, is to have a Meta-Control level to combine different strategies. So, if the application of a strategy does not solve completely the conflict, a second strategy is applied over the acquaintances selected with the previous strategy.

6 Conclusions

Agents' selection policy is a fundamental part of any distributed problem solving environment.

The paper focus in using the experience acquired during previous contracts to assign trust values to the acquaintances. Those values could lead the selection of the most trustworthy agent to whom offer a contract.

In the long term, the agent's tendency is to make a kind of coalitions, where the agents try to contract always the same agents for the same problem classes. The difference with classical coalitions is that the selection by means of trust values is more flexible. Coalition's agents are committed to contract other coalition's members, despite that they were included into the coalition because they are good solving other tasks, and may be there is not an adequate agent in it.

Another advantage with respect to coalitions is that if the agent grows worse the trust values change and the preferences too.

In societies based on coalitions, the new agents are hardly integrated, but with a high default trust value they can be easily integrated.

So, the agent's selection by means of trust values is more dynamic than coalitions.

Besides, coalition formation procedure usually is rather inefficient, it can be NP-complete, although some heuristics can reduce its complexity [Shehory and Kraus, 95], and can require a large number of communications to set up the coalitions.

On the other hand, the proposed solution is simple and efficient without overloading the processor with complex calculus.

Anyway, both perspectives can be combined to work together, using the trust values for the selection of the agents into a coalition.

The selection based on trust values can seem close to systems that associate plans with predetermined organizational structures to problem classes [Kinny *et al*,. 94]. It is not so. Despite that problem assignation can seem definable before the main problem's decomposition, the problem assignation is dynamic, and can change during the problem solving process, to adapt itself to the new data, and to the agent's availability.

Another advantage of our trust values is that the lying agents do not interfere heavily in the society's performance, because the values are calculated from the agent's internal evaluations of its own experience.

The selection by means of trust values, proposed in this paper, has been used in our MAS environment, with positive results. The examples show how agents select the acquaintances from wich they have best experiences, and discard the ones that usualy fail. When agents have acquaintances with good evaluations all contracts are offered to them, but, if later on they fail several times, a different acquaintance is selected. We have seen how the communication load decreases because, with a bad selection policy, unreliable agents could be selected, and when they failed, new negotiation processes had to be repeated with other agents.

References

[Davis and Smith, 88] R.Davis,R.G.Smith. Negotiation as a Metaphor for Distributed Problem Solving. *Readings in Distributed Artificial Intelligence*. A.H.Bond & L.Gasser Eds. Morgan Kaufman. pp:333-356.1988

[Kinny *et al*,. 94] D.Kiny, M.Ljungberg, A.Rao, E.Sonenberg, G.Tidhar, E.Werner. Planned Team Activity. *Lecture Notes on Artificial Intelligence,* 830: 227-256. Castelfranchi-Werner Eds. Springer Verlag 1994

[Marsh, 94] S.Marsh. Trust in Distributed Artificial Intelligence. *Lecture Notes on Artificial Intelligence,* 830: 94-112. Castelfranchi-Werner Eds. Springer Verlag 1994

[Shehory and Kraus, 95] O.Shehory, S.Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of 14th IJCAI*, pages 655-661. Montreal, Canada. 1995.

[Sichman *et al.,* 92] J.S.Sichman, Y.Demazeau, O.Boissier. When can knowledge-based systems be called agents?. *Brazilian Symposium on Artificial Intelligence*, Rio de Janeiro, Brazil, 1992. pp:172-185

[Smith, 88] R.G.Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *Readings in Distributed Artificial Intelligence*. A.H.Bond & L.Gasser Eds. Morgan Kaufman. pp:357-666.1988