AGENT COALITION FORMATION VIA INDUCING TRUST RATIO

Osama Ismail
Central Laboratory for Agricultural Expert Systems, Giza, Egypt

Samhaa R. El-Beltagy, Reem Bahgat
Faculty of Computers and Information, Cairo University, Giza, Egypt

Ahmed Rafea
Department of Computer Science, The American University in Cairo, Cairo, Egypt

Keywords: Agents, Genetic algorithms, Trust models.

Abstract: This work presents a model for assigning trust values to agents operating within a collaborative multi-agent system. The model enables agents to assess the trustworthiness of their peers, and thus, to be able to select reliable ones for cooperation and coalition formation. In this work, the performance of a group of agents – a team – that collaborate to achieve a shared goal where the individual contribution of each agent is unknown, is evaluated. The work thus aims to present a reliable method for calculating a trust value for agents involved in teamwork. More specifically, this research presents a model – called Inducing the Trust Ratio Model - for evaluating the individual trustworthiness of a group of agents. Toward this end, the model makes use of genetic algorithms to induce the trust ratio of each coalition member. Empirical analysis is undertaken to evaluate the effectiveness of this model.

1 INTRODUCTION

The work presented herein aims to augment a multi-agent system with a trust model which enables the collaborating agents to select peers that have the best performance for future collaboration purposes. The testbed used for experimenting with the developed model is the Collaborative Expert Agents System (CEAS) Architecture (Ismael et al., 07) in which cooperation between heterogeneous knowledge based systems can be achieved. Focus is placed on developing a trust model capable of quantifying a trust measure for individual agents working in teams within multi-agent systems. Obviously, it is in the best interest of the truster to delegate a task in a way that maximizes the probability of the task being completed with the highest possible quality of service. Thus, agents must attempt to minimize the risk of failure by choosing trustworthy resources. To do so, agents must be able to accurately assess and compare the trustworthiness (i.e. the expected performance) of potential *Provider Service Agents*.

Previous work has introduced a variety of trust models based on different criteria to derive the trustworthiness of a single agent (e.g. Falcone et al., 01; 03; Jensen et al., 04; Ramchurn et al., 04; Dong, 06). However, in these models the trust value is calculated based on the individual performance of each agent within a multi-agent system. There are cases when the individual performance of an agent can not be determined in a straight forward manner. This is the case for example when an agent is collaborating with other agents for achieving a certain task, and where the result of the collaboration can be evaluated but independent evaluation of the output of each agent is not possible, For example, suppose there is a particular problem (task) that requires the collaboration of different agents to be accomplished. A team of agents formulates solutions by each tackling (one or more) sub-problems and synthesizing these sub-problem solutions into an

Ismail O., El-Beltagy S., Bahgat R. and Rafea A. (2009).
AGENT COALITION FORMATION VIA INDUCING TRUST RATIO.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 469-474
DOI: 10.5220/0001657504690474

overall solution. Therefore, the final result is the outcome of the collaboration. This result is evaluated according to its quality and represents an assessment of the performance of the team as a whole. So, evaluation of this result is an evaluation of the teamwork rather than of independent agents that contributed to the formulation of this result. Assuming that the output can be assigned a trust value, the problem addressed by this work is how to distribute this value between participating agents according to the contribution of each. Since each agent member in the team has particular capabilities, each member may participate by a different ratio in executing the allocated task. Therefore, we need to provide a mechanism by which individual agents within a team can be assigned a trust ratio according to each individual agent's capabilities; this ratio will represent the real trustworthiness of each agent as closely as possible.

Towards this end we investigate the use of genetic algorithms for inducing a trust ratio for each agent coalition member.

This paper is organized as follow: section 2 presents the proposed model – Inducing the Trust Ratio Model (ITRM). Section 3 presents the empirical evaluation of the proposed model while section 4 concludes this paper and outlines some future research directions.

2 INDUCING THE TRUST RATIO MODEL

In this work we concentrate on efficient task allocation through coalition formation as a means for cooperating agents. So, the issue of which agents to trust when forming a coalition becomes quite important. We propose a solution to this problem through the use of what we've called Inducing the Trust Ratio Model (ITRM). To build and utilize the model, three phases are undergone: the *exploration phase*, the *inducing phase* and the *refinement phase*. In the *exploration phase*, a set of test cases is randomly generated and presented to possible agent teams. The performance of each team is then evaluated. In the *inducing phase*, exploration phase results are analyzed and processed through the use of a genetic algorithm. The output of this phase is a trust value for each agent. The *refinement phase* is a phase that is always active after completion of the first two phases. In this phase agents classified as low trust agents and agents that join the multi-agent system are periodically re-evaluated so as to allow such agents the chance to improve their performance and for their trust values to be adjusted accordingly.

2.1 The Exploration Phase

In this phase, the primary purpose is to explore the performance of a community of agents through presenting the multi-agent system with a real task and allowing it to form different teams to run on randomly generated real cases for the purpose of completing that task. The results are then assessed through system users. So, for each case the system will select the collaborating agents (team) from a number of potential agents that provide the same required service but with different qualities - in a random way - allowing the system to learn about the performance of an unknown provider (i.e. exploring the provider population). For each of these cases a system user will provide a rating as to the quality of the teamwork. The system user in this sense acts as an arbitrator judging the outcome against what s/he knows should be the result. This for example can take the form of different predefined criteria. This rating is then recorded together with the name of the team of participating agents in a transactions database, for future trust evaluation in the next phase. For simplicity we denote the name of the agents with the name of the service they provide and a suffix number (so agent x1 is a provider of service x while agent z3 is a provider for service z). Assuming we have a task that requires the invocation of three services (hence the collaboration of three agents), various agent combinations can be formed for achieving this task. A system user evaluates the output of these teams by giving it a value which represents its quality or the trust value for the teamwork. So, the rating for the teamwork's performance denotes the trust value for the entire team and represents the sum of the participating agents' performances. Therefore, the collective trust of a team can be represented by equation 1.

$$ct = \frac{\Sigma_{x \in t}\, w_x \,.\, tr_x}{n} \quad (1)$$

where, ct represents the collective trust value, w_x represent the weight given to agent x (determined according to each agent's role), tr_x represents the trust ratio that belongs to agent x, t is the set of participating agents in the team and n is the number of elements in the set t. For simplicity, we assume that all agents in the domain have the same weight. This equation is used in the next phase.

2.2 The Inducing Phase

The goal of this phase is to assign a trust ratio to each agent based on a examples obtained from the exploration phase. Since, coalition members may differ and the space of the case's possibilities is vast, exploring each possible case as outlined in the previous phase with each possible team, would be impossible. Here, we need a suitable mathematical model that considers the set of tuples in the transactions database as a set of simulations equations as represented by equation 1, and resolves the different trust values for each agent by solving these equations simultaneously. To model this problem, a genetic algorithm was employed. An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. In this case, a chromosome is a collection of trust values; and each gene represents an agent's trustworthiness. Each chromosome in the population is evaluated and ranked according to its relative strength within the population by applying its values on all equations. As stated before, each tuple can be represented as an equation stored in the transactions database. The goal from the evaluation of the chromosome is to calculate its fitness by counting the number of equations satisfied under this chromosome's values. Thus, a random population of potential solutions is created, then each one is tested for success, selecting the best chromosome to pass on their 'genes' to the next generation, including slight mutations to introduce variation. The process is repeated until the program evolves a workable solution. After reaching a predefined threshold that represents the minimum accepted fitness, a chromosome is selected to represent the optimal solution to the problem being solved. The values of the best chromosome represent the trust ratio of each agent in the domain. This trust ratio is stored in the *Service Agent Profiling* database, which can be used to enable the system to select agents that have the best performance for future collaboration.

In the selected testbed, the *Server Agent* (as illustrated in the CEAS architecture in will be the responsible unit that manages the *Service Agent Profiling* database through maintaining a record of trustworthiness for each agent in the platform. The *Service Agent Profiling* database forms the primary source for selecting partners, and is itself updated periodically as will be described in the next phase.

2.3 The Refinement Phase

We cannot assume that an agent's behavior will remain constant over time since its performance may alter (for better or worse) over time. If a truster knows that certain interacting partners provide an acceptable level of service, they might never choose to interact with any other agent that they know less about. This attitude may mean that new agents never get a foothold in the environment, even if they offer a better service than other established agents. The goal of this phase is thus to inform low performing agents of their trust rating within the system so as to allow them to improve their performance, and to periodically re-evaluate the performance of agents within the system so as to update any changes and assign trust values to new agents. Towards the fulfillment of the first part of this goal, the *Server Agent* navigates the *Service Agent Profiling* database, selects agents who have the lowest performance (i.e. those whose trust value falls below an acceptable level of performance) in each domain of service, and sends them an inform message telling them that they have the lowest trustworthiness. Towards the fulfillment of the later, the system periodically monitors the environment, taking into account environment changes, such as new coming agents, withdrawal or disappearance of previously existing working agents, agents with improved performance or agents with better performance than the current set of peers previously tested. In other words, in this phase the cycle of inducing trust ratios is restarted from the beginning with the *Service Agent Profiling* database being updated based on the new outcomes.

3 EVALUATION

This section presents the results of an experiment conducted in a collaborative knowledge-agents environment with the aim of evaluating the proposed approach. Empirical evaluation is used as the method of measurement because it allows us to assess the performance of a trust model in terms of how much benefit it can bring to its users. This requires us to compare ITRM's performance with that of another model. But, inducing the trust ratio for each coalition member –to our knowledge - has not been explicitly addressed within the field of multi-agent systems yet. Therefore, we could not apply this type of comparison. Instead, we compare the performance of the CEAS equipped with the

ITRM with the performance of the CEAS with no trust model.

3.1 Description of the Testbed

In the selected testbed, there are a number of competing *Provider Service Agents* that can fulfill a particular task, with each providing a different quality of service. Without loss of generality, and in order to reduce the complexity of the testbed's environment, it is assumed that there are four distinct set of services in the testbed, called: irrigation services, soil services, water services, and climate services, which are provided by the following *Expert Agents* respectively: *Irrigation Expert Agents*, *Soil Expert Agents*, *Water Expert Agent*s, *and Climate Expert Agent*s. Hence, there is more than one *Expert Agent* which offers the same service, but which vary in performance.

The objective here is to conduct an experiment that demonstrates the capabilities of the system in building an irrigation scheduling table. We take a practical and experimental approach in our investigation. To this end we adopt a concrete example in the agriculture domain within which to test and evaluate the ITRM. However the ITRM is not restricted to this or any particular application area.

3.2 Evaluation Tools

Two different tools have been developed to facilitate and expedite the evaluation process, the first is the TCGM and the second is the ESS. A Test Cases Generator Module (TCGM) was implemented for automatically generating a set of random test cases for different knowledge base components according to the contents of the knowledge base components under consideration (Rule Cluster File). These test cases are then stored in a composite *Test Cases Library* to be used in the execution stage. Figure 1, illustrates the overall structure of the proposed model. The main goal for implementing this module is to speed up the recommendation process (Exploration Phase – as illustrated previously in Section 2.1) through the use of a preprocessing module which is based on three steps: random test case generation, random test case execution and automatic test case evaluation. It is worth noting that the behavior obtained during this phase is representative of the behavior to be expected during actual interaction.

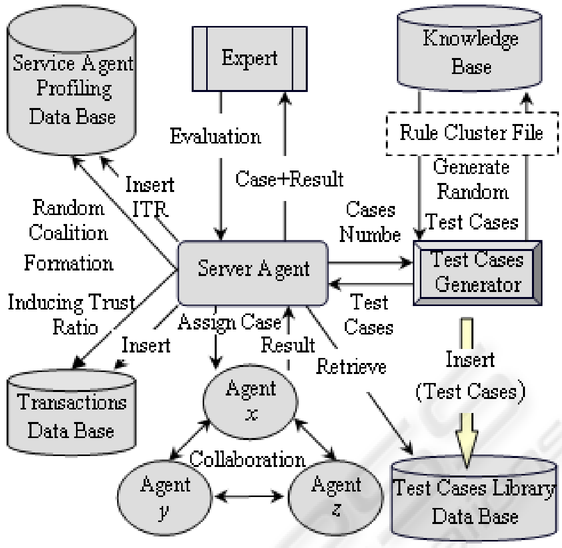

Figure 1: Overall structure of the TCGM.

To evaluate the results of the system, we've developed an Expert Simulator System (ESS) whose role is to evaluate system results. The mechanism through which the ESS works is: The ESS is equipped with a Knowledge Based System (KBS), which produces what could be considered perfect solutions to cases under consideration; these represent the base-line for comparison. Therefore, to evaluate the result of a specific case, the system sends the case parameter(s) and the result to the ESS. The ESS processes the case based on its received parameter(s). The result of this evaluation is a number denoting how closely the obtained result is to the baseline.

3.3 The Experiment

The agricultural domain is one in which numerous successful expert systems have been developed (e.g Rafea and Mahmoud, 2001). We were able to obtain twelve pre-existing expert systems for irrigation, soil, water, climate systems in agriculture and transform them into a community of cooperating agents. These agents are divided as follows: 3 *Irrigation Expert Agents*, 3 *Soil Expert Agents*, three *Water Expert Agent*s, and 3 *Climate Expert Agents*.

For the purpose of comparison, we implemented a version of the testbed without the facilities offered by the ITRM and then another with these facilities. To differentiate between the system built with a trust model and the one without, we will refer to the former as CEAS+ and the latter as CEAS-. Our empirical evaluation consists of a series of simulations tailored to show the developed model's performance. In each experiment there are two stages: in the first stage the system generates a set of random test cases (training cases), and presents the multi-agent system (CEAS-) with the task of solving these cases. In this stage, the system selects an agent team randomly and without use of the trust model. The collective performance for each team in the platform is then automatically calculated (by using the ESS). In the second stage the same set of test cases is presented to CEAS+ where team formation is based on previously recorded trust values and the collective performance for the selected team is calculated also using the ESS. The results for each case are then compared.

The performance of each group of agents in terms of utility gain is plotted on a chart to show the trend of performance change. The ***x*** axis represents the number of interactions (number of cases) and the ***y*** axis represents the effectiveness percentage (performance). When the number of interactions is set to forty, the effectiveness average for CEAS+ is about 93.03 %, indicating that the system can achieve a stable and high performance at this point while the effectiveness average for CEAS- is about 86.19 % for the same number of cases as illustrated in Figure 2.

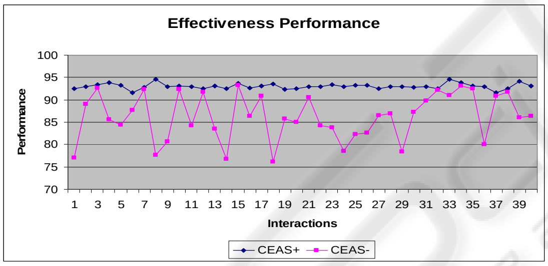

Figure 2: Effectiveness performance based on 40 previous test cases.

When the number of interactions is set to sixty, the effectiveness average for CEAS+ is about 93.02 %, while the effectiveness average for CEAS- is about 86.26 %, as illustrated in Figure 3. When the number of interactions is set to eighty, the effectiveness average for CEAS+ is about 93.09 %, while the effectiveness average for CEAS- is about 84.85 %, as illustrated in Figure 4. When the number of interactions is set to one hundred, the effectiveness average for CEAS+ is about 92.99 %, while the effectiveness average for CEAS- is about 85.78 %, as illustrated in Figure 5.

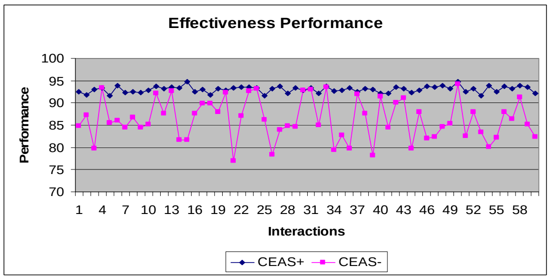

Figure 3: Effectiveness performance based on 60 previous test cases.

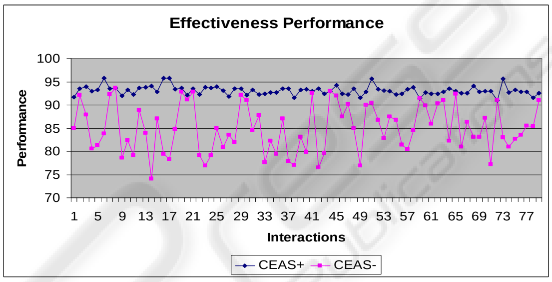

Figure 4: Effectiveness performance based on 80 previous test cases.

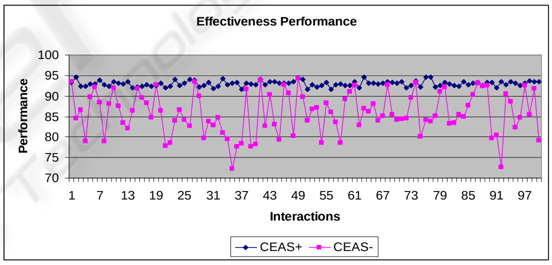

Figure 5: Effectiveness performance based on 100 previous test cases.

3.4 Hypothesis Testing

A mere comparison of the performance of the two systems does not allow us to conclude that one system performs better than the others in all cases. This is because the population of possible situations is infinitely large and the results from one experiment are only from a small sample of that population. Given this problem, statistical inference techniques should be used since they allow us to draw a conclusion about an unseen population given a relatively small sample. To the extent that a sample is representative of the population from which it is drawn, statistical inference permits generalizations of conclusions beyond the sample (Cohen, 95). The hypothesis testing method as a statistical inference technique, allows us to confirm with a predefined confidence level, whether the difference of the two means of the two sample groups' performance, actually indicates that one system has higher performance than the other, and hence, eliminate the random factor in selecting the samples.

Table 1: Terms used in the hypothesis testing procedure.

Term	Definition
N	The number of interactions chosen as the test period (number of test cases).
P_{CEAS+}	The mean performance of a sample of agents using ITRM after their n^{th} interaction.
P_{CEAS-}	The mean performance of a sample of agents using no trust model after their n^{th} interaction.
S_{CEAS+}	The standard deviation of the performance sample of CEAS+.
S_{CEAS-}	The standard deviation of the performance sample of CEAS-.

The result of carrying out the hypothesis testing procedure for different test periods (i.e. 10, 20, 30, 40, 60, 80, and 100 interaction) is illustrated in Table 2.

Table 2: Hypothesis testing results.

Number of	P_{CEAS+}	P_{CEAS-}	S_{CEAS+}	S_{CEAS-}	*SE*	*DF*	*t*	*P-value*
10	87.77	84.67	1.47	4.45	1.479	4.96	2.1	0.045
20	91.13	84.71	0.84	4.91	1.114	16.32	5.8	1.46E-5
30	92.92	82.74	0.65	5.83	1.071	26.07	9.5	3.02E-10
40	93.03	86.19	0.63	5.20	0.829	57.25	8.3	1.32E-11
60	93.02	86.26	0.71	4.55	0.595	297.5	11.4	2.16E-25
80	93.09	84.85	0.98	5.11	0.585	585	14.1	2.59E-39
100	92.99	85.78	0.68	5.08	0.513	732.86	14.1	3.32E-40

Since, the *P-value* for all cases is less than the *significance level* (0.05), we cannot accept the null hypothesis. Therefore, this table shows that the corresponding hypothesis tests conclude that the CEAS+ outperforms the CEAS- and that the performance difference is statistically significant (using the confidence level of 95%).

4 CONCLUSIONS

Previous work addressing trust, has investigated *active trust*, but *passive trust* has not been explicitly addressed within the field of multi-agent systems to date. In active trust, the performance of individual agents from various perspectives is evaluated using various sources of trust information, such as, direct interaction or through witness reports. But, in such cases, the agent that is meant to be evaluated is known in advance. In passive trust (addressed by this work), the performance of an agent within a group of agents that collaborate to achieve a shared goal is what is being evaluated. To do so, a trust ratio for each agent in the team is induced. The presented model for achieving this task: Inducing the Trust Ratio Model (ITRM) is thus a novel model for trust evaluation that is specifically designed for general application in multi-agent systems. In order to verify the claim that this model is both effective and useful, empirical evaluation was carried out. Through this evaluation it was demonstrated that agents using the trust model - ITRM - provided by CEAS are able to select reliable partners for interactions and, thus, obtain better utility gain compared to those using no trust measure.

REFERENCES

Cohen, P., 1995. *Empirical Methods for Artificial Intelligence,* The MIT Press.

Dong, H., Jennings, N., Shadbolt, N., 2006. An Integrated Trust and Reputation Model for Open Multi-Agent Systems. In *Journal of Autonomous Agents and Multi-Agent Systems*.

Falcone, R., Barber, S., Korba, L., Singh, M., (editors) 2003. *Trust, Reputation and Security: Theories and Practice,* Volume 2631 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg.

Falcone, R., Singh, M., Tan, Y., (editors) 2001. *Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives,* Volume 2246 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg.

Ismael, O., El-Beltagy, S., Bahgat, R., Rafea, A., 2007. Collaborative Knowledge Based System through Multi-Agent Technology. In *the 42nd Annual Conference on Statistics, Computer Science, and Operations Research.* Institute of Statistical Studies and Research, Cairo University.

Jensen, C., Poslad, S., Dimitrakos, T., (editors) 2004. *Trust Management,* Volume 2995 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg. *Proceedings of the Second International Conference, iTrust 2004 Oxford,* UK, March 29 – April 1, 2004.

Rafea, A., Mahmoud, M., 2001. The Evaluation and Impact of NEPER Wheat Expert System. *Fourth International Workshop on Artificial Intelligence in Agriculture IFAC/CIGR*, Budapest, Hungary.

Ramchurn, D., Dong, T., Jennings, R., 2004. Trust in Multi-Agent Systems. *The knowledge engineering review,* 19(1): 1-25, March 2004.