ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 8 N. 1 (2019), 13-18
eISSN: 2255-2863
DOI: http://dx.doi.org/10.14201/ADCAIJ2019811318

Algorithm Analysis in Multi-agent Systems

Laura Pacheco, Naiara Sánchez, Antoni Tur, and C. David Téllez de Meneses

id00698146@usal.es, naiaralva@usal.es, lluquino@usal.es, id00707666@usal.es,
University of Salamanca

KEYWORD

Multi-agent systems; optimization;

ABSTRACT

This paper presents a multi-agent system that looks for the most optimum algorithm of its type. For that purpose it will use several agents which will be in charge of testing the algorithms and comparing the outcome to see which is the most efficient. Thanks to this procedure the most optimum procedure can be obtained.

1. Introduction

Nowadays many of the computing programs are designed without emphasize in the optimization of the algorithms used, which decrease the software performance with each non-optimal algorithm of the code. A multi-agent systems is presented in this paper that compares algorithms and establish its efficiency based on the others efficiency rate.

The idea is using N agents in a hierarchical multi-agent system ruled by other agent which will order them to do tests in a certain type of algorithms. The transmitter agent passes the other agents the parameters needed for doing the tests. When they have this data each agent chooses an algorithm of that type from a list, and conducts the tests on it. When the tests are over, each agent will communicate the rest of its outcome so they can compare the obtained data and this way see the most efficient algorithm. Lastly the agent that has the most efficient algorithm reports it to the transmitter agent. A multi-agent system is that system which is made of agents (generally, intelligent agents) that interacts each other. These systems are really useful to solve the problems that only with an agent is very difficult or impossible to solve. The domain of the multiagent system or the distributed artificial intelligent is a science and a technique that uses network artificial intelligent systems. The most important element in a multi-agent system are the agents.

An intelligent agent is an entity able to perceive its environment thanks to actuators. The rationality of the decisions is totally necessary, this is to say, it doesn't considered intelligent system if it doesn't able to perceive its environment and make decisions respect the information but also those decisions must be rational, although most of times the agents conduct don't use to be optimal.

In our case, it's completely necessary the use of a multi-agent system for the communication in the program. The difficulty could be if an agent does these tests on every algorithm, because it would be higher than if it uses an agent for each algorithm. It is also important to consider the important role that has the transmitter agent to the rest of agents that compose the system. Below, a JADE agents architecture is proposed where it's explained the process to follow to obtain the most optimum algorithm. It will begin with a State of art where we will comment some similar studies to this paper. Then the architecture proposal will be explained and the results will be discussed. Lastly there will be a conclusion about the paper.

2. State of Art

In base of the obtained outcome, in this paper it can claimed that the task of comparing algorithms can be make in a more efficient way by agents than by human subjects (Wu, 2001). So several agents have been used instead of only one because usually the agents efficiency increases in cooperative multi-agent systems (Tan, 1991).

Because of software systems are "becoming 'always-on' entities that cannot be stopped, restored, and maintained in the traditional way", we need the most efficient algorithms so we can reach that kind of systems as an standard (Zambonelli, *et al*., 2003).

There are tools to simulate user's behaviour using intelligent agents.(de la Prieta Pintado *et al*., 2015)

and due to reach the learning and reasoning capabilities we can use different toolkits.(Wooldridge *et al*., 1995).

Some systems works on real time or need to analyze a very huge amount of information so they need to know, implement and use the most efficient algorithms due to perform their tasks. (Alvesa *et al.,* 2015).

The multi-agent system recommends the most optimum algorithm for each problem, there are other multi-agent systems that use algorithms for doing recommendations in different fields like the multi-agent approach PUMAS-GR (Villavicencio *et al.,* 2016) that relies on negotiation techniques to improve group recommendations, the multi-agent recommender system for e-tourism (Bedi *et al.,* 2014), the multi-agent system AVATAR1 of personalized TV content (Blanco-Fernandez *et al*., 2004) or a multi-agent system for recommending training courses to engineering professionals(Marivate *et al.,* 2008).

3. Architecture Proposal

For a start the user chooses the type of algorithm on which he wishes to do the tests and reports it to the transmitter agent. The types of algorithms on which we have enabled the user to make a choice are sorting algorithm, searching algorithm, greedy algorithm and probabilistic algorithm.

Once the decision about the type of algorithm that the agents are going to compare is taken, its executed the selection of each algorithm. For that selection each agent will be given a number, so that the first agent will have the number one, the second agent the number two and so on.

In the database the algorithms are saved with some necessary data: name, type and the average time of execution. The last of them it is saved after the tests.

The first agent to which the number one has been assigned, access the algorithm database and chooses the first algorithm it finds of the type picked by the user. Once the algorithm is selected, it isn't communicated to any other agent, instead it is the turn of the second agent, which will repeat the process followed for the first agent, but instead of picking the first algorithm, it will pick the second one.

The agents will repeat this process until all of them have a algorithm assigned. When an agent chooses an algorithm, the test will begin right away doing (for example if the user has chosen to compare algorithm

of sorting type, an array of m numbers will be sort n times), doing taking at the end of the results obtained the average time taken by the algorithm to solve the proposed problem.

Finally, each agent will send the others those other average times that have obtained after the tests so that each agent will have the data of the other algorithms executed for the other agents. This way each agent can do a comparative of those times and discover which are the more efficient among all.

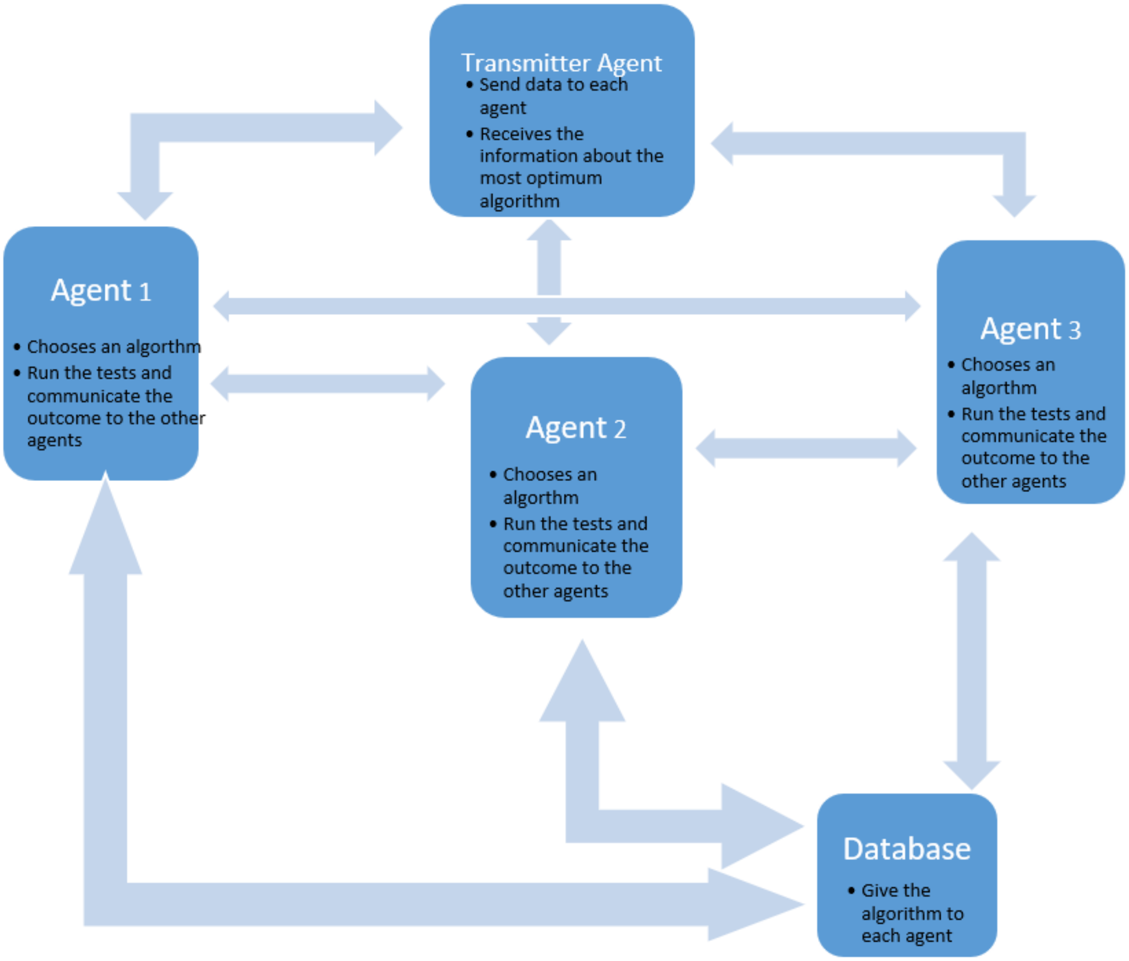
Transmitter Agent
• Send data to each agent
• Receives the information about the most optimum algorithm
Agent 1
• Chooses an algorthm
• Run the tests and communicate the outcome to the other agents
Agent 2
• Chooses an algorthm
• Run the tests and communicate the outcome to the other agents
Agent 3
• Chooses an algorthm
• Run the tests and communicate the outcome to the other agents
Database
• Give the algorithm to each agent

Figure 1: System diagram

Once the most efficient algorithm has been identified, the agent who did the tests on that algorithm will send the transmitter agent the data about that algorithm.

The transmitter agent will be on charge of displaying to the user, the data about the most efficient algorithm (name, type and average time of execution). This way the This way the search process of the most efficient algorithm will be finished.

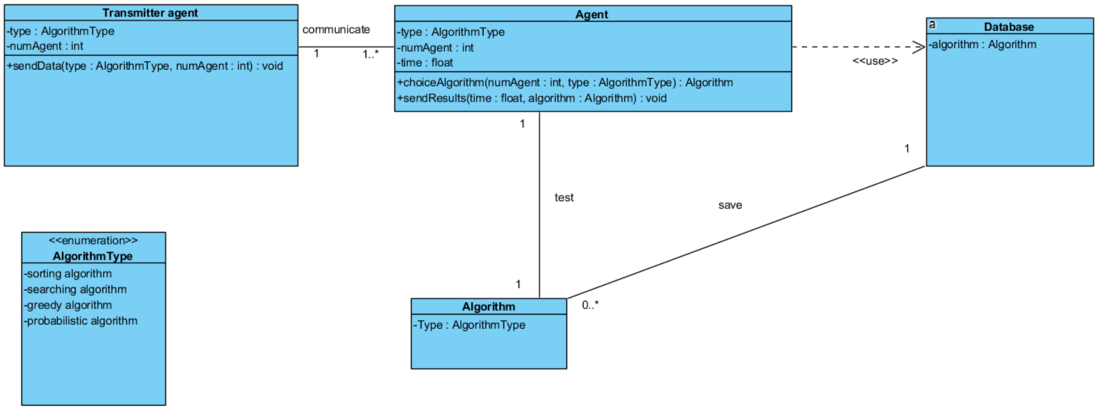

Figure 2: MAS Class diagram

4. Case Study and Results

It may be the case that there are more algorithms than agents. In this example it is possible that the most optimum algorithm won't be found. So if we want the most optimum algorithm we must instance as many agents as algorithms contains the database, so each agent will select the first free algorithm and everyone will execute, comparing then each and every time.

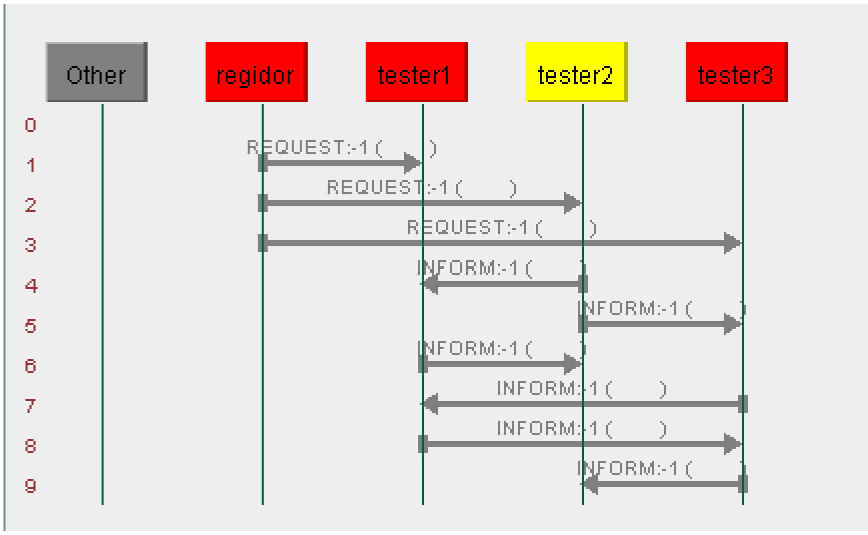

Figure 3: Screen shot of the JADE Sniffer

It can also happen the contrary, that there are more agents than algorithms, in which case there will be agents that cannot select any algorithm because there won't be any free. In this case, the agent that cannot select any algorithm will communicate to the rest of agents. Sending them a message with a failure performative indicating its local type and then it will kill itself. Because an agent that has no algorithm selected is useless for our paper. The process will not be affected by the suicide of any agent so the agents that have been assigned an algorithm will proceed with normality.

Lastly is the base case, in which the number of agents is the same as the number of algorithms.

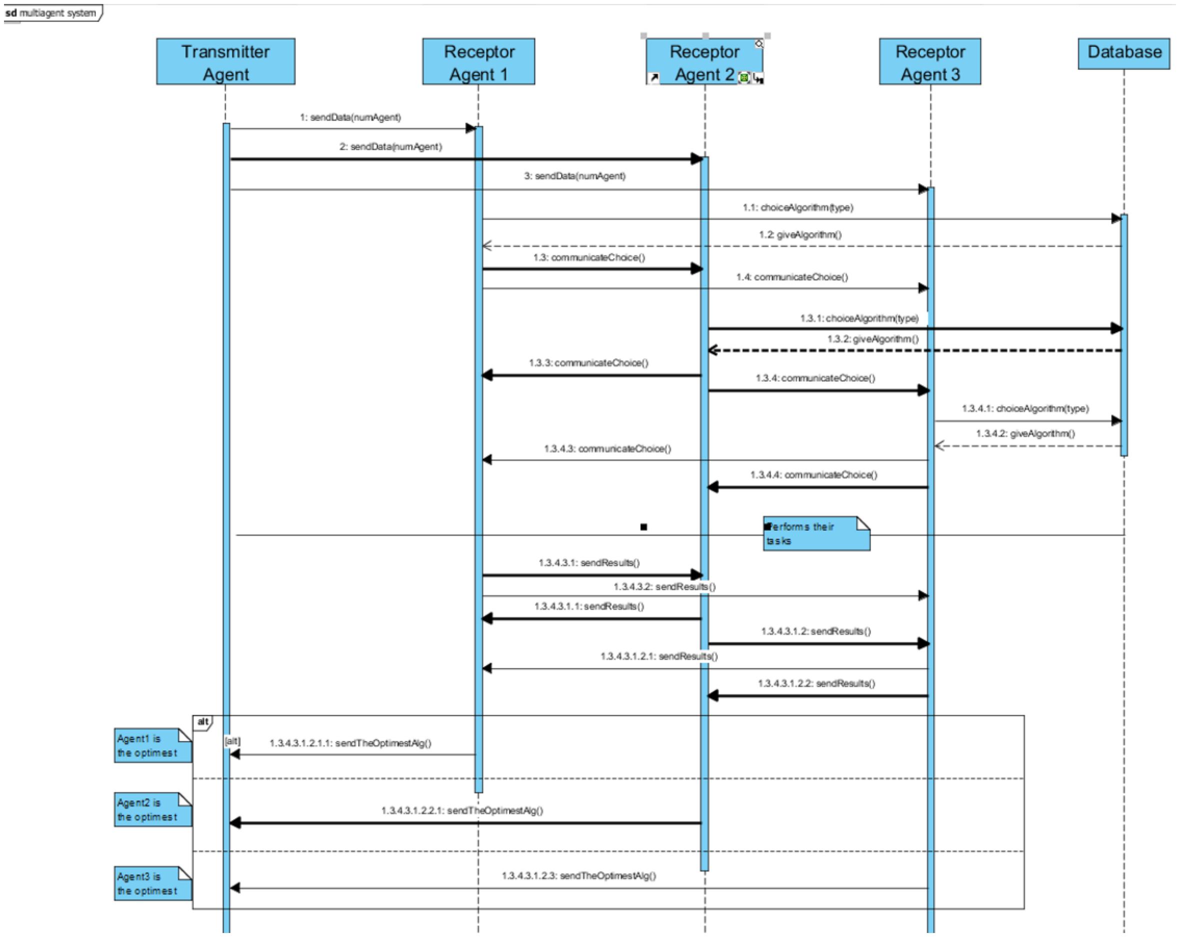

Figure 4: MAS sequence diagram

5. Conclusion

Concluded our study we can assure that the method developed is fully functional. This way, our proposal is completely valid to figure out which of the algorithms to study is optimum.

In this paper, we perform the comparison of some algorithms that we have chosen in an eventful way, but we demonstrate that our program is able to work with any kind of algorithm. Once the algorithm more optimum has been found after several comparatives, it can be useful to perform multiple projects or different ideas linked with this topic. Thanks to this, we can save up time when choosing an algorithm.

This is a very interesting topic, because we can discover the best and worst algorithms of some kind and we can see how the communication between agents is done.

6. References

Blanco-Fernandez, Y. et al., 2004. AVATAR: An Advanced Multi-agent Recommender System of Personalized TV Contents by Semantic Reasoning. s.l., s.n., pp. 415-421.

Bedi, P., Agarwal, S. K., Jindal, V. & Richa, 2014. MARST: Multi-Agent Recommender System for e-Tourism Using Reputation Based Collaborative Filtering. New York, NY, USA, Springer-Verlag New York, Inc., pp. 189-201

Blanco-Fernández Yolanda, Pazos-Arias José J., Gil-Solla Alberto, Ramos-Cabrer Manuel, Barragáns-Martínez Belén, López-Nores Martín, García-Duque Jorge, Fernández-Vilas Ana, and P. Díaz-Redondo Rebeca, 2004. AVATAR: An Advanced Multi-agent Recommender System of Personalized TV Contents by Semantic Reasoning. In: Zhou X., Su S., Papazoglou M.P., Orlowska M.E., Jeffery K. (eds) Web Information Systems – WISE 2004. WISE 2004. Lecture Notes in Computer Science, vol 3306. Springer, Berlin, Heidelberg

Christian Paulo Villavicencio, Silvia Schiaffino, J. Andrés Díaz-Pace, and Ariel Monteserin, 2016. A Group Recommendation System for Movies based on MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, Salamanca, v. 5, n. 3, p. 1-12, nov. 2016. ISSN 2255-2863.

Consensus-based Approach for Keyword Extraction from Urban Events Collections Ana Alvesa, b, Bernardete Ribeiroa a CISUC, Department of Informatics Engineering, University of Coimbra, Portugal, 2015.

De la Prieta Pintado Fernando, Rodríguez González Sara, Corchado Rodríguez Juan M., and Bajo Pérez Javier, 2015. Infrastructure to Simulate Intelligent Agents in Cloud Environments, Journal of Intelligent and Fuzzy Systems, IOS Press.

Marivate, V. N., Ssali, G., & Marwala, T., 2008. An Intelligent Multi-Agent Recommender System for Human Capacity Building. s.l., IEEE, pp. 909-915.

Tan, M., 1991. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, GTE Laboratories Incorporated,

Wooldridge, M., and Jennings, N.R., 1995. Agent Theories, Architectures, and Languages: A Survey, In: Intelligent Agents, Springer.

Wu, D. J., 2001. Software agents for knowledge management: coordination in multi-agent supply chains and auctions, University of Pennsylvania 2001

Zambonelli F., Jennings N.R., and Wooldridge M., 2003. Developing multiagent systems: The gaia methodology, ACM Transactions on Software Engineering and Methodology.