ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 7 N. 4 (2018), 45-52
eISSN: 2255-2863
DOI: http://dx.doi.org/10.14201/ADCAIJ2018744552

Exploiting Asynchrony in Multi-agent Consensus to Change the Agreement Point

Diego Pellitero, David Sanchez, Mario Hernandez, and Alvaro Serrano

diego.pellitero@usal.es, davidss@usal.es, mariohc@usal.es, alvaritorad@usal.es
University of Salamanca

KEYWORD

Multi-agent system; consensus; Neglect Benevolence; collective behavior

ABSTRACT

Reaching agreement by consensus is fundamental to the operation of distributed systems, such as sensor networks, social networks or multi-robot networks. In real systems, the resource limitations available to individual agents and communication delays typically result in asynchronous control models of discreet time for consensus. In this paper, we model the problem where a set of agents arrive at a consensus on the value of a variable of interest, being guided by one of them.

1. Introduction

In this paper we will study how multi-agent systems make a discrete-time consensus with a compromised agent. Each agent executes an update of the consensus with a possible delay. This delay allows us to model real world features, such as delaying entry into a robotic swarm to improve performance of the same. Asynchrony is present in most multi-robot systems in the real world. We study how the asynchrony can be used to change the point of agreement of the swarm, showing that by simply changing the update period or the initial delay, the agents can change their decision. In this way, distributed robot systems will show vulnerabilities that will serve us well:

a) Enable mitigation strategies to reduce the influence of an adversary.
b) Introduce delay to improve performance.

But first, let's look at some interesting and important features about multi-agent systems. A multi-agent system (SMA) is a system composed of multiple intelligent agents that interact with each other. Multi-agent systems can be used to solve problems that are difficult or impossible to solve for an individual agent.

The set of all the agents is assigned a mission that can be decomposed into different independent tasks, so that they can be executed in parallel. Each agent has limited knowledge, and this limitation can be either the knowledge of the environment, the mission of the group or the intentions of the other agents The problem arises as an objective that cannot be achieved by a single subsystem and needs the collaboration of others to obtain the solution. In the interactions between the different agents there are four key concepts:

- Communication: Ability of the agents to exchange information and knowledge in an understandable way.
- Coordination: A set of supplementary actions that can be performed in a multiagent environment to achieve an objective, and that an agent, with the same objectives, could not achieve by its own [Malone 1988].
- Cooperation: Mechanism by which agents, working together to achieve a common objective, define a strategy to achieve this objective.
- Negotiation: It allows reaching joint coordination decisions through explicit communication [Muller, 1996].

The applications of the agent paradigm can be classified into three classes [Dignum, 2004]:

- Open systems: The structure is capable of changing dynamically, that is, its components are not known first, change over time and can be heterogeneous.
- Complex systems: They are related to large domains and unpredictable. They use modularity and abstraction to deal with problems.
- Ubiquitous systems: They have the objective of improving a computer system by using computers available in a physical environment, normally distributed, but making it completely transparent to the user.

In the case of our example we could say that the most important feature in our agents is the negotiation and we could classify the system as an open system. Depending on the reasoning system they use, there are three major agent architectures that are: Deliberative, reactive or hybrid. In our case we are going to focus on deliberative architectures. The deliberative architectures use models of symbolic representation of knowledge, this means that they are based on the classical theory of planning [Maes, 1989]. The classical theory of planning is that there is a set of plans and from these, is part of an initial state to reach a final objective state that must be met. A deliberative agent contains a symbolic model of the world where decisions are made through logical reasoning mechanisms (based on pattern matching and symbolic manipulations).

To finalize this introduction, let's have a general vision about the communication between the agents. Communication between agents allows them to synchronize actions, send and receive knowledge, resolve conflicts in the resolution of a task. The communication allows the coordination between a group of agents so that only those necessary actions are executed.

The agents use a communication language (ACL – Agent Communication Language) to communicate information and knowledge. They present two types of coordination that are global and individual. Global coordination: When the multi-agent system is able to determine and plan globally the actions of the different agents; and Individual coordination: When agents of the multi-agent system have total autonomy to decide what to do and resolve conflicts with other agents. In our case, the type of communication will be of an individual type.

2. State of the art

According to (Ramchurn *et al.*, 2004) negotiation is a form of interaction in which several agents, who have conflicting interests and a desire to cooperate, attempt to reach a mutual agreement which is acceptable in the division of scarce resources. This paper is based on previous studies on multi-agent consensus systems, such as (Feng *et al.,* 2017), which studies the effects that topology has on the system and its robustness, (Cheng *et al.,* 2010), which explains a new consensus model to improve its reliability in wireless networks. One of the most interesting studies would be (Xiao *et al.,* 2006) which deals with how topologies and variables linked to communication delay time affect. In this last study it is spoken of how when varying the times of communication between agents (communication time-delays) the information cannot reach other agents, reason why the system cannot reach a consensus asymptotically. Those deviations can lead to two potential scenarios, and adversary can take the advantage to move the agreement point toward a desired value or we can use it in our benefit using the Neglected Benevolence.

Neglect Benevolence is a concept that captures the idea that it may be beneficial for system performance if the human operator, after giving a command, waits for some time before giving a subsequent command to the swarm. This raises the important question of the existence of a calculation of the optimal time for the operator to give input to the swarm in order to optimize swarm behavior. Human operators are limited in their ability to estimate the best time to give input to the swarm. Therefore, automated aids that calculate the optimal input time could help the human operator achieve the best system performance.

In (Parsons *et al.,* 2000) and (Parsons *et al., 2001)* it is described as a search process within potentials agreements with the purpose of finding a solution that satisfies the requirements and stakeholder preferences. For (Huhns *et al.,* 1999) negotiation is a process through a number of agents make a decision together, each one trying to comply an individual goal. A similar vision can be found in (Rosenschein *et al., 1994*), where negotiation is seen as a form of joint decision making, where two or more parties make decisions a search in a space for possible solutions with the aim of reaching an agreement.

It is clear that studies on negotiation have not arisen exclusively in the field of the artificial intelligence but has been a subject widely studied by other disciplines: philosophy, economics, psychology, mathematics, among others. And as can be imagined, many negotiation models are based on these works. For example, mathematics has made a great contribution to game theory (Neumann *et al.,* 1947) or (Nash *et al.,* 1950) which is the basis of most of the negotiating models for e-commerce.

3. Architecture proposal

In this paper, the problem where an indeterminate number of agents reaches a consensus on the value of a variable of interest. For this purpose, the agents have access to the information of their neighbors (given the agent a_n, a_n has access to the information of a_{n-1} and a_{n+1}). Every agent sends information to the neighbors periodically(every agent has their own period). If the period and the initial delay changes, the agreement point can move close to a preferred value.

In this example (with 3 agents + one coordinator) the communication between the agents is shown by means of messages in order to carry out a negotiation and finally reach an agreement, the sniffer checks the passage of messages between agents which use request and inform to carry out the communication.

ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 7 N. 4 (2018), 45-52
eISSN: 2255-2863 - http://adcaij.usal.es
Ediciones Universidad de Salamanca -

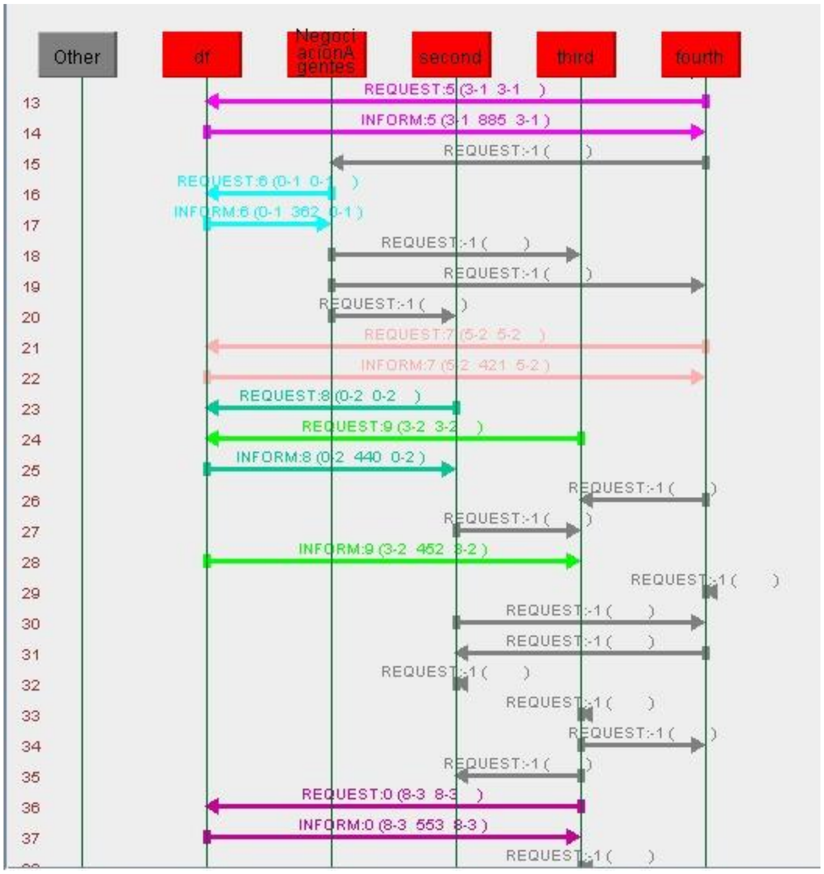

Figure 1: Interaction Diagram (example with 3 agents)

In the lines 18, 19 and 20 the coordinator sends a signal, so the other agents can start with the negotiation. After that lines the negotiator agents start to communicate with each other, trying to reach a consensus on the value of a variable. As can be seen in *Figure 2* there are 2 classes of agents which are used to resolve the problem:

1-Class: Agente_Generador_1_6

The class Agente_Generador_1_6, generates a random value and stores it on a variable where we want to reach a consensus. This type of agent get the value from other agents in order to find a value that satisfy the group of agents. This type of agent uses the behaviors OneShotAgenteComienzo and OneShotNegociaciones.

2-Class: Agente_Coordinador_7

The class Agente_Coordinador_7 receive the number of agents we want to use and waits until every agent is ready to start the consensus. This class use the behavior OneShotAgenteCoordinador.

1-Class: OneShotAgenteCoordinador

This class is the one that coordinates the agents at the beginning of the execution, it waits until the agents are ready.

2-Class: OneShotAgenteComienzo

It waits until the Coordinatior send the signal to indicate that the other agents are ready.

3-Class: OneShotNegociaciones

After the negotiator agent receive the signal to start this behavior starts running to accomplish the consensus.

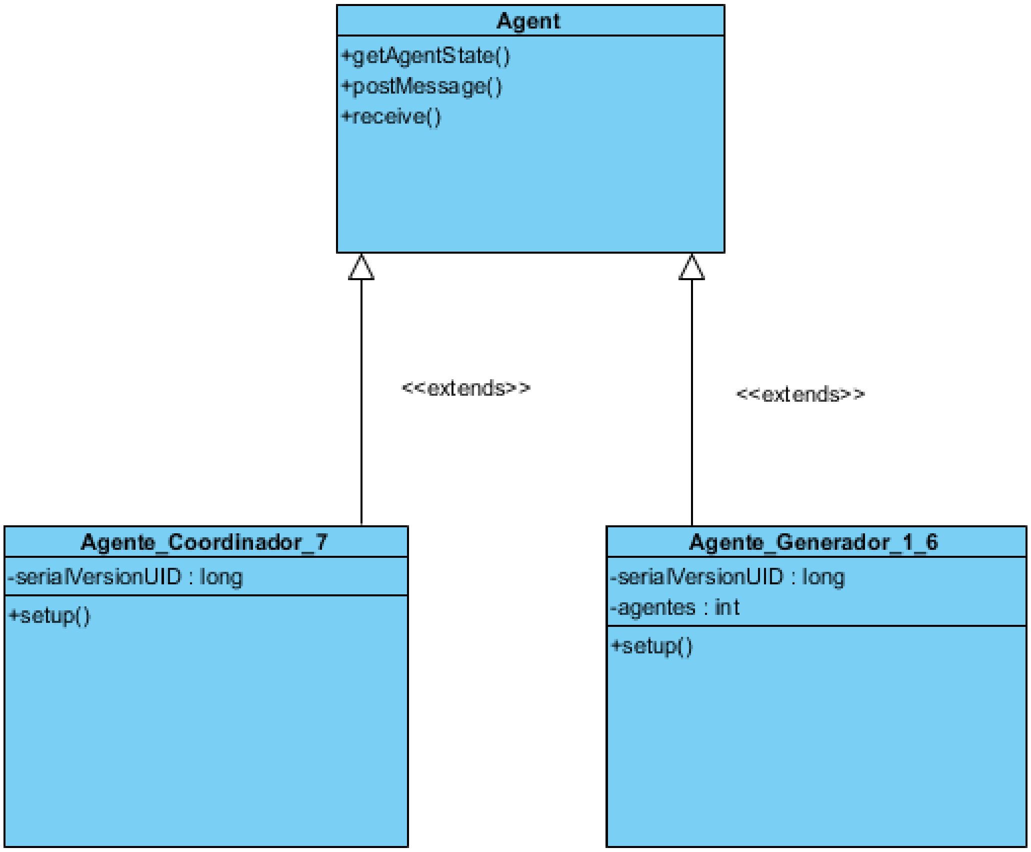

Figure 2 Class diagram of agent types

Figure 3 presents a class diagram of the behaviors used in the agents explained above.

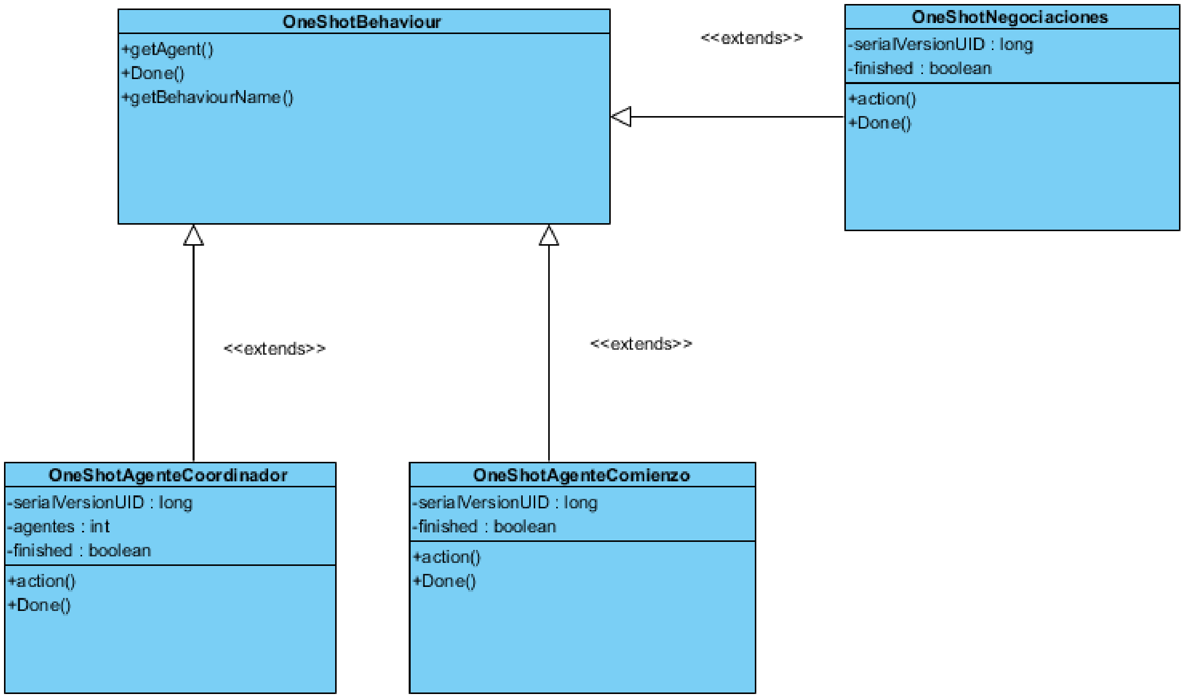

Figure 3 Class diagram of behaviors

4. System Analysis

A multi-agent system has been implemented for the simulation of a collaboratory agreement reaching algorithm. The goal of the simulation is to prove the effectiveness of the algorithm for obtaining an agreement on the value of a number, in a multi-agent environment. The algorithm involves several steps which can be summarize in the following description of a use case:

The agents start the decision process with a random number which will be communicated to the rest of agents. Depending on the mean for all the responses, each one of the agents will propose another random number, closer to the value mean value calculated before. This process will repeat until an agreement is reached.

There are two different agents (*Figure 2*), that implement behaviors, involved in the decision process. The first type of agent, of which only one instance is created, will hold the role of coordinator agent. This agent will act as arbitrator between the rest of agents. It will be the first to launch itself and will ask how many generating agents are going to participate so that later, when they join, they can organize themselves and start their executions at the same time.

The second type of agent is the generating agent. At least two instances of this type will be created, which will be in charge of reaching an agreement. They present two OneShot behaviors (*Figure 3*) that will be launched sequentially. The first behavior is used to contact the coordinating agent and when the rest of agents are ready, the coordinating agent can pass them to the execution. The second behavior to be executed once the coordinating agent gives them step, is the negotiation. The second behavior will not end until everyone has reached an agreement. Each agent will propose a random number that will be communicated to the rest, and with that information will calculate the mean value for the proposals of the rest of the agents, including the proposal made by itself; then it will give a result. This result is compared with the initial proposal in order to determine if an agreement has been reached or else, the value has to be changed in order to move towards a common value in the following iterations of the algorithm.

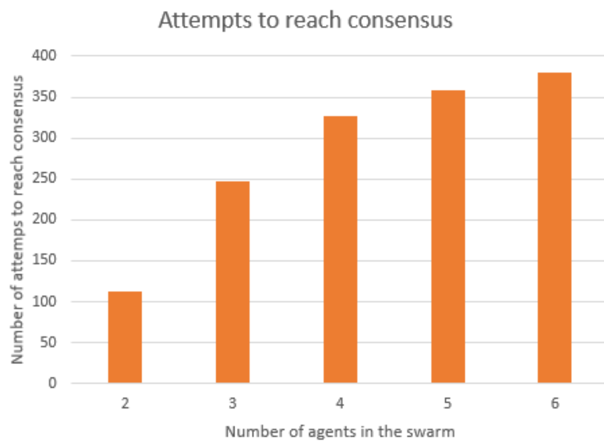

Figure 4 Attempts vs. agents involved

Several tests have been carried out with different number of agents to see the performance of the proposed algorithm in regard to number of participants in the negotiation process. All tests are performed with a range of values from 0 to 1000. A total of 9 tests have been done with each configuration. In (*Figure 4)* can be seen the number of attempts made by the agents vs. the number of agents in the swarm. It is clear that with low number of agents the performance increase.

5. Conclusion

As seen in (Nagavalli *et al.,* 2018) using the discrete-time consensus protocol, can be influenced changing the initial delay and update periods. We have seen that this influence can have two different effects, one of them is that an adversary can take advantage to lead the swarm to an agreement point closer to the adversary desired value. The other effect is the Neglected Benevolence, a condition where we can improve the performance of the swarm by changing the time between inputs or updates to a specific one.

The agents presented in this paper can communicate only to its neighbors to send the updates periodically. The two types of agent (Figure2) used is one coordinator (it waits until every agent joins the swarm) and some generators (generate a random value and starts the consensus protocol).

As we have seen in the tests the bets performance is achieved with a low number of agents, but this performance has a b-side, good independence of the swarm is hard to achieve with a low number of agents.

6. References

Barbara Wasson Barbara (1998). Identifying Coordination Agents for Collaborative Telelearning. International Journal of Artificial Intelligence in Education (IJAIED), 1998, 9, pp.275-299. Https://telearn.archives-ouvertes.fr/hal-00197381/document

Gonzalez, C., Burguillo, J., Llamas, M., & Laza, R (2013). Designing Intelligent Tutoring Systems: A Personalization Strategy using Case-Based Reasoning and Multi-Agent Systems. ADCAIJ: Advances In Distributed Computing And Artificial Intelligence Journal, 2(1), 41-54. doi:10.14201/ADCAIJ2013244154

Heras, Stella; de la Prieta Pintado, Fernando; Julian Inglada, Vicente; Rodríguez González, Sara; Botti, Vicente; Bajo Pérez, Javier;Corchado Rodríguez, Juan M (October 2012). Agreement technologies and their use in cloud computing environments. Progress in Artificial Intelligence. Springer

Huhns M. Andy L. Stephens L (1999), “Multiagent system and societies of agents”, In Multiagent Systems, pp. 79-118, 1999.

Julián, V., Navarro, M., Botti, V., & Heras, S (2015). Towards Real-Time Argumentation. *ADCAIJ: Advances In Distributed Computing And Artificial Intelligence Journal, 4*(4), 35-58. doi:10.14201/ADCAIJ2015443558.

Maes, P (1989). The dynamics of action selection. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 991-997, Detroit, MI.

Muller Jorg P., Jörg P. Müller Jörg P (1996) ,The Design of Intelligent Agents: A Layered Approach

Nash J. F (1950) , "The bargaining problem", Econometrica, pp. 155-162, 1950.

Neumann J. Von y O. Morgenstern O (1947), "Theory of games and economic behavior", Princeton University Press, NJ, 1947.

Parsons S., Sierra C. Y Faratin P. Jennings N. R (2000) "Automated negotiation", in Proc. 5th Int. Conf. On the Practical Application of Intelligent Agents and Multi- Agent Systems, Manchester, UK, 2000, pp. 23-30

Parsons S., Faratin P., Lomuscio A. R. Jennings N. R (2001), "Automated negotiation: prospects, methods and challenges", International Journal of Group Decision and Negotiation, pp. 199–215, 2001.

Ramchurn S. D., Jennings N. R., mcburney P., Parsons S. And Sonenberg L. Rahwan I (2004), "Argumentation-based negotiation", in The Knowledge Engineering Review, 2004, pp. 343-375.

Rosenschein . Andy G. Zlotkin G (1994), "Rules of encounter: designing conventions for automated negotiation among computers, MIT Press Cambridge, MA, USA ©1994, ISBN:0-262-18159-2

Sasanka Nagavalli Sasanka, Ramitha Sundar Ramitha, Katia Sycara. Katia (2018) "Exploiting Asynchrony in Multi-agent Consensus to Change the Agreement Point" AAMAS, July. 2018.

Virginia Dignum Virginia (2004), A Model for Organizational Interaction: Based on Agents, Founded in Logic, Utrecht University

Xiao F., L. Wang L (2006). State consensus for multi-agent systems with switching topologies and time-varying delays.

Yin Chen Yin, Roberto Tron Roberto, Andreas Terzis, and Rene Vidal Rene (2010). Corrective consensus: Converging to the exact average. In Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, 1221–1228.

Yuting Feng Yuting, Zhisheng Duan Zhisheng, Guanrong Chen Guanrong and, Wei Ren Wei (2017). Consensus of multi-agent systems with fixed inner connections.