Self-stabilization in cooperative multi-agent systems by a reset: Position paper

Rodion Podorozhny, Wuxu Peng, and Dewayne E. Perry
rp31@txstate.edu; wxp@cs.txstate.edu; perry@ece.utexas.edu

Texas State University, San Marcos, Texas; Texas State University, San Marcos, Texas; Empirical Software Engineering Laboratory, The University of Texas, Austin, Austin, Texas

Abstract. We believe it is important to be able to analyze and verify negotiation protocols as first class entities because they embody the reasoning mechanism responsible for agent interaction.
The underlying idea of our approach to self-stabuilization is a reset of the negotiation protocols once the negotiating agents either reach an invalid state or recognize the inevitability of reaching an invalid state.
By an invalid state we mean such a state where the combined utility of the actions of the negotiating agents decreases below a reasonable threshold making it impossible to accomplish the combined goal of the agents. Thus, the definition of an invalid state is dynamic in the sense that states can become valid or invalid based on the actions of other agents and on the environment state that affects the combined utility.
We use the formalization of a multi-agent system as a Markov decision process that separates the modeling of local action reasoning and the negotiation reasoning as in [19]. We extend the model with features that enable the detection of cases when a protocol results in an invalid state. Self-stabilization is then achieved either by resetting the initial states of the agents or by modifying the negotiation reasoning policy.

1 Introduction

Many multi-agent systems (MAS) rely on negotiation protocols as a mechanism for increasing the combined utility of agents' actions. The majority of verification techniques for the MAS suggested so far do not focus on verification of protocols. As a matter of fact, the majority of work on modeling, analysis, and verification of multi-agent systems do not distinguish between local action and negotiation reasoning. We believe it is important to be able to analyze and to verify negotiation protocols as first class entities because they embody the reasoning mechanism responsible for agent interactions. Joining the negotiation reasoning with local action reasoning in the analysis models can lead to unreasonable increase of the model's size and complicated specifications of desirable properties that pertain to the negotiation reasoning.

Our approach formalizes a multi-agent system as a Markov decision process that separates the modeling of local action reasoning and of negotiation reasoning as in [19]. We extend the model with features that enable the detection of cases when a protocol results in an invalid state. Self-stabilization is then achieved either by resetting the initial states of the agents or by modifying the negotiation reasoning policy.

2 Research Context

Multi-agent systems have been increasingly recognized as an efficient approach to the design of robust systems that can provide nearly optimal solutions under the conditions of uncertainty and limited resources.

While there are multiple definitions of an agent, there seems to be an agreement that the following main attributes define an agent: it is intelligent in the sense that it can optimize its actions; it is interacting in the sense that its actions are affected by actions of other agents; it is autonomous as the agent can choose to refuse execution of a requested task or it can choose a hard to predict solution from a great variety of options [17].

The notion of utility is quite often used as a measure of the "goodness" of a solution provided by a set of agents. The kind of interaction between the agents is a major contributor to an increase of the utility of agents' actions. While the interaction can be implicit we focus on explicit interaction reasoning which is quite often found in cooperative multi-agent systems. Thus we focus on the cooperative MAS even though our method for analysis, verification, and self-stabilization can be applied to competitive systems too.

Negotiation protocols are used so that agents can evaluate another agent's situation to find a solution that increases their combined utility. The negotiation may be initiated to arrive at a mutually acceptable agreement between agents on such issues as strong dependence between their activities ("enables" relationship), weak dependence ("facilitate" relationship), task allocation, and redundancy of activities [3].

A number of negotiation protocols and multi-agent systems that rely on negotiation have been suggested so far ([15], [13], [3], [16], [11], [2], [5]).

It is recognized that any multi-agent system is likely to use some negotiation techniques to increase the utility of their combined actions ([8]). While benefits of the multi-agent systems such as being able to effectively reason under uncertainty, being able to increase robustness, are recognized ,and the multi-agent approach gains popularity and becomes widespread, there is a growing need to be able to verify these multi-agent systems.

There is on-going work on the application of model-checking for the verification of multi-agent systems ([18], [7], [1]). Also, other approaches such as the application of situation logics ([4]) are used for the multi-agent systems verification. The vast majority of this work though does not address the issue of multi-agent negotiation as a first class entity. The models of multi-agent systems are quite often represented as a product of models of individual agents in the mentioned approaches. It is likely that for practical reasons, considering a wide range of protocol-based multi-agent systems, it is also important to be able to verify the properties of the negotiation protocols. There has also been some work on the application of Petri-nets for modeling and analyzing multi-agent systems ([10]).

There has been on-going work in application of reinforcement learning techniques for agents' learning ([9], [14], [6]). These techniques focus on learning local action strategies by agents, omitting the explicit representation of interaction reasoning between the agents.

Some of the work that recognize the negotiation reasoning as a first class entity include [19] and [12]. Their work is primarily focused on the evaluation of negotiation policies that decide whether an agent should communicate at a certain point in time. The work by Ping Xuan uses the modeling of multi-agent systems as decentralized Markov decision processes, introduces a separate representation for the local action reasoning and negotiation reasoning, and suggests approximations for optimal control of agent's interactions. The work by Pynadath evaluates a number of models based on Markov-decision processes,including the ones that treat negotiation explicitly, provides complexity analysis for the problem of finding optimal control (termed as Communicative Multi-agent Team Decision Problem or COM-MTDP), and provides a well-grounded characterization of the complexity-optimality trade-off among various means of team coordination.

The models for interaction reasoning in these approaches focus on the decision whether an agent should send a message, while a possible relation between the content, the size of the message and the frequency of communication is not taken into account. Neither do the models capture the possibility of improper negotiation - there is an assumption that negotiation reasoning will lead to a solution of the requested task by the agents with some non-zero utility. Thus there is no ground for reasoning about correctness of the negotiation protocols.

We propose to extend the Markov decision process-based model for multi-agent systems to accommodate the cases of improper negotiation and suggest a self-stabilization modification to the protocol that implements the negotiation.

3 Approach

In our discussion we will assume that the "goodness" of actions of all the agents is measured by the notion of utility that is a function of cost, duration, and quality of results of performing individual actions by the agents. Thus, the global utility is $GU = f(c, d, q)$. We assume that the greater the GU, the higher the "goodness" of combined actions by the agents. While the particular function for GU would depend on the needs of optimization (different ratios of emphasis between the contributions of cost, duration, quality) there is an underlying assumption that the GU is positively affected by quality and negatively affected by cost and duration [1]. Regardless of the particular problem domain, we assume that a successful operation of the MAS system is characterized by some non-zero

[1] For those problem domains where the duration might also be considered as a positive contribution, e.g. the actions of learning, we note that the quality would be a function

current utility. If the utility goes below a certain threshold ($GU \leq GU_{min}$) then the system enters an invalid state and the goal of the whole multi-agent system is considered unachievable.

In this work we propose the creation of methods that let us construct multi-agent systems capable of detecting and avoiding the possibility of entering such a state, thus ensuring that the system does accomplish its goal.

By definition agents interact in the sense that their actions are influenced by actions of other agents. If agents do not negotiate at all then it is very likely that they will not achieve their combined goal. On the contrary, if they either negotiate two frequently or they exchange all the available information about their states and policies of achieving the individual goals then they can incur unacceptable costs that would bring the GU below G_min. The action of negotiation can be treated as any other action, thus the GU will be negatively affected by the cost and duration of the negotiation action.

Let us now consider possible reasons for a failure in a multi-agent system ($GU \leq GU_{min}$). To keep this work focused on the reasons due to negotiation reasoning we make the following assumptions: there is no loss of messages due to unreliable communication; and agents are not malicious (i.e. they conscientiously attempt to increase the combined utility). We are not considering the cases of multiple parallel negotiations, neither do we consider possible complications due to different levels of autonomy among the agents. None of the agents possesses a global view (i.e. the knowledge of states of the rest of the agents and the knowledge of optimal policies) to accomplish the task at hand.

Thus the primary reason for a failure in an MAS is partial knowledge of agents about the environment and the optimal policies of achieving a combined goal. We propose a self-stabilizing MAS (SSMAS) by way of exchanging partial information about their previous experiences that capture policies that increase the combined utility. The example given below is illustrated using a model based on partially observable Markov decision processes (POMDPs) as introduced in (Ping Xuan et al.).

We assume that the reasoning of agents is divided into two categories: local action reasoning and negotiation reasoning. Local action reasoning is responsible for choosing a certain action that transitions the agents to another state. Negotiation reasoning is responsible for deciding whether to initiate negotiation and what kind of information to pass. In this model the agents reason about local actions and negotiation in turn. Thus an episode of agent's reasoning is represented as

$$s_0^X, m_0^X, a_0^X; s_1^X, m_1^X, a_1^X; ...s_{T-1}^X, m_{T-1}^X, a_{T-1}^X; s_T^X,$$

where T is the number of cycles in an episode, $s_i^X (i = 0, 1, ..., T)$ are the states for agent X, $m_i^X (i = 0, 1, ..., T)$ are the negotiation reasoning actions for agent X, $a_i^X (i = 0, 1, ..., T)$ are the local reasoning actions for agent X.

of the duration, thus the quality would still remain a positive contribution while the duration would be negative. That is doing the action of learning faster receives a higher reward.

The policy for local action of agent X is denoted as $\pi^{X,a}$, the policy for negotiation reasoning for agent X is denoted as $\pi^{X,m}$. Thus $\pi^{X,a_j}(s_i)$ determines the next state s_{i+1} that agent X takes after performing action a_j while in state s_i based on the policy captured by weights $w(s_i, a_k)$ where $k = 0, 1, ...K - 1$ covers the K actions possible at state s_i.

Similarly $\pi^{X,m_j}(s_i)$ would determine the next message m_j while the agent is in state s_i. The message m_j can be *null* corresponding to no message sent. The negotiation action does not change the agent's states in this model. If an agent determines that its next local action according to its policy $\pi^{X,a_j}(s_i)$ would make $GU \leq G_{min}$ then it resets, moving to its local state s_0. A more reasonable solution would be to reset the agent whose anticipated action brings GU below the threshold to a previous state that is recognized as the last valid state from which some other route via the state space can still bring the agent to the accomplishement of the goal without damaging the GU. Finding such a state for a reset can be done based on "checkpoints" and the reevaluations of the weights for actions leading out of the "checkpoints". For the initial version of the approach we suggest resetting the agent to state s_i. The rest of the agents can still continue operation. The reset is done only by the agent that recognized individually that its next action will inevitably bring the combined utility below the desired threshold.

4 Motivating example

For a motivating example let us suppose a multi-agent system whose goal is to increase the savings while buying products. Each agent has a limited view in that it cannot know the sets of products available at all the stores. The states correspond to sets of products, while each agent has a task to buy a certain subset of the products $prodList^X \subset products$. Let us assume that the product lists of agents are not intersecting, i.e. they do not have to compete with each other. The local actions correspond to buying a product or returning a product. On buying a product the money amount of an agent taking this action is reduced by the price of the product bought and its savings are increased by the savings associated with the product at the store. On returning a product the money amount is increased by the price of the product returned and the savings are decreased by the savings associated with the product at the store. The negotiation reasoning decides whether to communicate and it communicates the prices and savings of products available at a store local to a specific agent. Every instance of negotiation has a communication cost associated with it: c_{m_i}. Every product l out of the whole set of products has savings associated with it: $save^{prod_l}$. Each agent has a limited amount of money to spend. There is a certain threshold of the money amount below which an agent cannot go - it corresponds to the price of a ride home. Let us say, for an agent X the price of a ride home is $retHome^X_{min}$. The global utility is expressed as a sum of savings achieved by all the agents.

Deciding whether to communicate influences the trade-off between the increase of the combined utility and the cost of an agents' actions. The reasoning approach to this decision that we envision is based on the level of confidence in the estimate of the GU (LC^i for agent i) and the level of constraints on agents' actions. If agents had known each others policies at any state throughout an episode then there would have been no need to communicate. Each agent could predict the actions of other agents, which would let them arrive at a very accurate estimate of the GU. That, in turn, would let them predict and avoid invalid states (i.e. such states that would bring the GU below the predefined threshold). Such a situation is not likely, especially for large state spaces. Even if the agents start with the same knowledge of the policies ($\pi^{i,a}$ and $\pi^{i,m}$ are the same for any agent i out the set of agents A) they would learn more information about the subsets of states in which they operate which would lead to different levels of knowledge of individual agents about different subsets of states. If the level of confidence in the estimate of the GU by an agent is based on the time passed since the previous communication then the timings for communication would be periodical. If the level of confidence in the estimate of the GU by an agent is based on the level of constraints on an agent's actions then the timings would depend on the characteristics of the state space and the paths of individual agents through the state space governed by their policies.

A prediction is attributed with a confidence level that is based on the number of cycles (modeled amount time) passed since the previous communication between the agents: $(t_{cur} - t_{lastComm})$. Let us assume the confidence level decreases with time according to a certain time decay function. There is a certain threshold for the confidence level (LC_{min}) below which the agent decides to communicate, requesting the information about sets of products and savings available at the stores known to other agents.

Thus the self-stabilization of a multi-agent system can be described as making a decision about the communication and exchanging information about the states and policies of other agents. In the case of this example, the maximum $save^{prod_l}$ associated with buying product $prod_l$ ($l = 0, 1, ...|P_i| - 1$, where P_i is a set of products available at state s_i) is a deciding factor for performing the buying action a.

When deciding on a negotiation action m_i an agent chooses one of the following options:

- if $LC \leq LC_{min}$ then m_i is a $requestForOtherPrices$
- otherwise m_i is a $null$

When deciding on a local action a_i an agent chooses whether to reset or do the action:

- if $LC \leq LC_{min}$ then move to state s_0
- otherwise perform the action a_i and move to the state to which the transition by a_i leads

5 Future work

We have introduced an approach to self-stabilization of cooperative multi-agent systems by a reset. Also, we provided a model and an example that illustrate our approach.

We still need to evaluate the self-stabilization by a reset using a testbed. The problems to be solved include

- finding a method for determination of a state to reset to (other than s_0),
- evaluating various assessment methods for the level of confidence in the individual agent's estimate of the global utility (GU)
- evaluating various time decay functions for the estimate of the GU and
- striking a trade-off between the communcation cost and the accuracy of the assessment.

We envision creation of a method for updating existing protocols with reset capability to make them more robust. A long term goal of this line of work is to arrive at an underlying model for an optimal reset, such that helps to choose agents to be reset and optimal states they need to reset to.

References

1. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
2. S. E. Conry, R. A. Meyer, and V. R. Lesser. Multistage negotiation in distributed planning. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 367–384. Kaufmann, San Mateo, CA, 1988.
3. K. Decker and V. Lesser. Designing a Family of Coordination Algorithms. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 73–80, January 1995.
4. Gunter Gans, Matthias Jarke, Gerhard Lakemeyer, and Thomas Vits. SNet: A modeling and simulation environment for agent networks based on i* and ConGolog. In *CAiSE02*, pages 328–343, Canada, Toronto, May 2002. Springer.
5. John R. Graham, Keith Decker, and Michael Mersic. Decaf - a flexible multi agent system architecture. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):7–27, 2003.
6. Junling Hu and Michael P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *ICML*, pages 242–250, 1998.
7. Magdalena Kacprzak, Alessio Lomuscio, and Wojciech Penczek. Verification of multiagent systems via unbounded model checking. In *AAMAS*, pages 638–645, 2004.
8. S. Kraus and D. Lehmann. Using interacting goal-processes for automated negotiation. In Z. W. Ras, editor, *Methodologies for Intelligent Systems, 4: Proc. of the Fourth International Symposium on Methodologies for Intelligent Systems*, pages 209–219. North-Holland, New York, 1989.
9. Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th Int. Conf. on Machine Learning (ML-94)*, New Brunswick (NJ, USA), 1994.

10. Irina A. Lomazova. Nested petri nets - a formalism for specification and verification of multi-agent distributed systems. *Fundam. Inform.*, 43(1-4):195–214, 2000.
11. T. Moehlman, V. Lesser, and B. Buteau. Decentralized Negotiation: An Approach to the Distributed Planning Problem. *Group Decision and Negotiation*, 1(2):161–192, January 1992.
12. David V. Pynadath and Milind Tambe. Multiagent teamwork: analyzing the optimality and complexity of key theories and models. In *AAMAS*, pages 873–880, 2002.
13. T. Sandholm and V. Lesser. Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. *First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 328–335, January 1995.
14. Yoav Shoham, Rob Powers, and Trond Genager. Multi-Agent Reinforcement Learning: a critical survey. Technical report, Stanford University, Stanford, CA 94305, 2003.
15. R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 357–366. Kaufmann, San Mateo, CA, 1988.
16. Katia Sycara, Joseph Andrew Giampapa, Brent K Langley, and Massimo Paolucci. The retsina mas, a case study. In Alessandro Garcia, Carlos Lucena, Franco Zambonelli, Andrea Omici, and Jaelson Castro, editors, *Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications*, volume LNCS 2603, pages 232–250. Springer-Verlag, Berlin Heidelberg, July 2003.
17. Gerhard Weiss, editor. *Multiagent systems : a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, Mass., 1999.
18. Michael Wooldridge, Michael Fisher, Marc-Philippe Huget, and Simon Parsons. Model checking multi-agent systems with mable. In *AAMAS*, pages 952–959, 2002.
19. Ping Xuan, Victor Lesser, and Shlomo Zilberstein. Modeling Cooperative Multiagent Problem Solving as Decentralized Decision Processes. *Autonomous Agents and Multi-Agent Systems*, 2004.