On the Failure of Game Theoretic Approach for Distributed Deadlock Resolution

(Extended Abstract)

Nadav Sofy
Department of Computer Science
Bar Ilan University
Ramat Gan 52900, Israel

David Sarne
Department of Computer Science
Bar Ilan University
Ramat Gan 52900, Israel

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Performance, Human Factors

Keywords

Deadlock resolution, Equilibrium, Bounded rationality, Modeling the dynamics of MAS

1. INTRODUCTION

Deadlocks may occur in many multi-agent environments and in various contexts. In particular, deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource. As such, the problem has received much attention in Operating Systems and Databases literature, resulting in various mechanisms for avoiding, detecting and recovering from deadlock situations. Recent advances in deadlock research extend the deadlock model to distributed environments. Here, deadlocks are harder to manage since none of the participating agents have a full knowledge of the entire system. Consequently, a number of approaches were pursued for handling deadlocks in distributed systems. Still, all these studies assume that agents are cooperative and follow a dictated deadlock resolution protocol.

Nevertheless, in many deadlock situations occurring in multi-agent systems, agents are self-interested and a cooperative resolution scheme cannot be enforced. This situation is also likely to hold in future virtual environments where agents migrate between different platforms, communicating and negotiating with other agents autonomously, without the mediation of the hosting platform. In such environments, deadlocks can be resolved only if an agent willingly retracts from its deadlock-related requirements. The problem becomes even more complex if the agents are not fully rational or are pre-programmed with various deadlock handling logic. In this case, each individual agent needs to be incentivized to comply with the required behavior. Here, the vast deadlock management solutions designed for cooperative and fully rational agents may become irrelevant.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Game-Theory principles can be applied to construct a simple stable distributed solution to the problem which guarantees immediate deadlock resolution. While game-theoretic approaches are widely used in studying MAS conflict situations, there is extensive evidence in literature for the failure of such approaches where the main players are people or bounded rational agents [1]. On the other hand, there are works that report the successful use of game-theoretic approaches, in particular in repeated interaction domains [2]. Therefore, the success of such approaches in the deadlock domain is a priori inconclusive. In this paper we report the results of an experiment testing the effectiveness of a Game-Theoretic approach to the problem of distributed deadlock resolution of autonomous self-interested partially-rational agents. This is a part of an ongoing research aimed at studying distributed deadlock resolution in such settings and designing a restructuring heuristic that changes the input that each agent receives as a means for affecting the agents' decisions to better align with the desired solution.

2. THE DEADLOCK MODEL

This paper considers the standard Coffman deadlock model, commonly found in Operating Systems literature. The system is in a deadlock state if a circular chain $A = \{A_1, ..., A_N, A_1\}$ of agents exists, where each agent $A_i \in A$ attempts to acquire a resource held by agent A_{i+1} (A_1, in case $i = N$) in order to proceed with its execution.

Each agent A_i is associated with a processing time t_{A_i}, the time it needs to use the resource it requests from A_{i+1} before releasing the resource that A_{i-1} is waiting for. Each agent can also willingly release the resource it holds (opt out) at any time. In such case the agent needs to acquire both the resource released (now held by the previous agent in the chain) and the resource it was waiting for (held by the next agent) in order to proceed with its task. We assume that resources are acquired as soon as they are available.

We assume the existence of a central entity (e.g., operating system) that receives demands for resources and expected processing times and identifies deadlocks as they occur. The central entity supplies system-related information to the agents, though it cannot preempt the agents' hold on resources or enforce any particular behavior. In particular, since agents in a MAS can block their regular operation for various reasons, we assume that the system informs the agents once they are actually in a deadlock and supplies them with the deadlock description. This latter information includes the number of agents in the deadlock and their processing times. The agents are assumed to be homogeneous in the sense that each agent has an equal chance of being the i-th agent in any deadlock and its processing time is drawn from a common distribution of values. An agent's strategy is thus the mapping $S : A \rightarrow t$, where t is the time since the deadlock is first reported to the agent until the agent opts out. The agents are assumed to be self-interested and their goal is to minimize the time it takes to complete their task. Since no agent has control over its own processing time, this goal is equivalent to minimizing its overall waiting time. From the system's point of view, the goal is to minimize the average waiting time of the agents.

3. ANALYSIS

In this section we develop the dominant Nash-Strategy for the problem. For exposition purposes, we use $A_{sub}(A_i, A_j)$ to represent the subchain of agents in A positioned between A_i and A_j. Also, WLOG, the deadlock is taken to be formed at time $t = 0$. Once an agent $A_i \in A$ opts out, the deadlock is resolved. In this case, agent A_i will need to wait $\sum_{k \neq i} t_{A_k}$, while any other agent A_j will wait a time equal to the total processing times of all the agents along the subchain $A_{sub}(A_j, A_i)$ (formally given by $\sum_{A_k \in A_{sub}(A_j,A_i)} t_{A_k}$). Notice that once the deadlock is resolved by agent A_i, no agent $A_j \neq A_i$ can reduce the time it needs to wait by opting out as well. This is because opting out will not affect the time that any of the agents in $A_{sub}(A_j, A_i)$ will need to wait until gaining a hold of the resource they requested.

From the system's point of view, regardless of the identity of the agent to opt out first, the optimum is achieved at $t = 0$. This is because all agents necessarily gain from an earlier deadlock resolution, given that all other parameters are fixed. If the deadlock is resolved by agent $A_i \in A$ at time $t = 0$, then the average waiting time is given by:

$$\frac{1}{N}\Big(\sum_{j \neq i} \sum_{A_k \in A_{sub}(A_j,A_i)} t_{A_k} + \sum_{k \neq i} t_{A_k}\Big) \qquad (1)$$

A lower bound for the expected average waiting time is obtained when agent A_i, for which Equation 1 is minimized, opts out at time $t = 0$. This solution can theoretically be achieved when each agent checks whether it is the agent to minimize Equation 1, and if so, opting out at $t = 0$. This solution can also be extended to the dominant Nash-Strategy when each agent waits indefinitely if it is not the agent that minimizes Equation 1. In such a case, none of the agents have an incentive to deviate from its strategy. Since none of the other agents ever opt out, the agent that is supposed to opt out at $t = 0$ will do better if it sticks with this strategy. Opting out at $t > 0$ is dominated by opting out at $t = 0$ for this agent and never opting out will necessarily result with an infinite waiting time. As for the other agents, since the deadlock is supposed to be resolved at time $t = 0$, none of them will have an incentive to deviate to a different strategy. In fact, based on the same argument, any protocol according to which one of the agents is selected to opt out by an external event (e.g., having the shortest or longest processing time) while none of the other agents ever opt out is in equilibrium.

4. EXPERIMENTS AND RESULTS

To enable the simulation of distributed deadlocks, we developed a system that simulates Coffman deadlocks. Agents in the system are put in a deadlock upon instantiation. The sole functionality of each agent is deciding whether to opt out of the deadlock based on the deadlock description and the time elapsed. The experiment was carried out with 28 agents programmed by computer science students in a core Operating Systems course. The goal set for the agents was to minimize their expected waiting time throughout the experiment. The students were given a detailed explanation about the game-theoretic-based solution to the problem. To simplify coding and assure that a deviation from the equilibrium strategy will not be a result of implementation bugs or difficulty, it was decided that the agent with the longest processing time in each deadlock will be the one to opt out. The drawbacks of deviating from this strategy, assuming everyone else is using it, were fully discussed and detailed in the task description. It was suggested that the students use this strategy, though it was made clear to them that there is no centralized mechanism enforcing it. In order to make the evaluation as realistic as possible, the experiment took place over the course of a few weeks, allowing the students to revise their strategies based on the results of thousands of deadlocks in which their agents participated. This process of repeated strategy updates and evaluations of performance was carried out until a week elapsed without any change made to the agents' strategies. The agents stored in the system at the end were considered the steady-state strategies. Deadlocks were generated automatically and randomly assigned with agents. The number of agents participating in each deadlock was uniformly drawn from a range of 2-10. The processing times were drawn from an Erlang distribution, which is the typical distribution of CPU burst times in operating systems, with parameters $\lambda = 0.01$ and $k = 1.5$ (yielding a mean of 150). Once an agent opted out, the system terminated and the waiting times of all the agents in the deadlock were calculated.

The results indicate that none of the subjects initially implemented the Nash-Strategy fully. Only 18 percent of the students have implemented the Nash-Strategy with an empty threat, by opting out after a constant time in deadlocks in which their agent is not the one with the longest processing time. The analysis of the steady-state set of distributions revealed that not only has no one changed her strategy to the game-theoretic one, even the number of partial implementations of the type described above decreased to only 3 percent of the strategies. In addition, the system's average waiting time in the steady-state was worse compared to the average obtained with the initial set of strategies.

The results demonstrate the failure of the Game-Theoretic approach in the distributed setting with self-interested bounded-rational agents. The main implication is that other approaches, such as input restructuring, should be considered for such settings.

Acknowledgments

This work was supported in part by ISF/BSF grants 1401/09 and 2008-404.

5. REFERENCES

[1] E. Manisterski, R. Lin, and S. Kraus. Understanding how people design trading agents over time. In *AAMAS*, volume 3, pages 1593–1596, 2008.

[2] M. Parameswaran, H. Rui, and S. Sayin. A game theoretic model and empirical analysis of spammer strategies. In *Collaboration, Electronic Messaging, AntiAbuse and Spam Conf.*, number 7, 2010.