The Complexity of Multiagent Systems: The Price of Silence

Zinovi Rabinovich
School of Computer Science and Engineering
Hebrew University of Jerusalem
91904 Jerusalem, Israel
nomad@cs.huji.ac.il

Claudia V. Goldman
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
clag@cs.umass.edu

Jeffrey S. Rosenschein
School of Computer Science and Engineering
Hebrew University of Jerusalem
91904 Jerusalem, Israel
jeff@cs.huji.ac.il

ABSTRACT

In this work, we suggest representing multiagent systems using computational models, choosing, specifically, Multi-Prover Interactive Protocols to represent agent systems and the interactions occurring within them. This approach enables us to analyze complexity issues related to multiagent systems. We focus here on the complexity of coordination and study the possible sources of this complexity. We show that there are complexity bounds that cannot be lowered even when approximation techniques are applied.

Keywords

Computational complexity and agent systems, Markov Decision Processes

General Terms

Theory

Categories and Subject Descriptors

F [**Theory of computation**]; F.2 [**Analysis of algorithms and problem complexity**]

1. DISTRIBUTED MDPS

Multiagent systems consisting of non-communicating agents can be represented by Decentralized Partially Observable MDPs (*Dec_POMDP*s) introduced by Bernstein et al. [3]:

Definition 1. *Dec_POMDP_k is a tuple $\langle \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^k, \{\Omega_i\}_{i=1}^k, \mathcal{O}, p, r, \mathcal{T}\rangle$, where*

- *k is the number of agents involved in the process;*
- *$\mathcal{S}$ is the set of system states;*
- *$\mathcal{A}^i$ is the set of actions available to agent i;*
- *Ω_i is the set of agent i's observations;*
- *$\mathcal{O} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^k \to \Pi(\Omega_1 \times ... \times \Omega_k)$ is a probability distribution. $\mathcal{O}(s, a^1, ..., a^k, s', o^1, ..., o^k)$ is the probability of agent i observing $o^i \in \Omega_i$ when the system arrives at state s' after the actions $a^1, .., a^k$ have been performed at state s;*
- *$p : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^k \to \Pi(\mathcal{S})$ is the system transition probability distribution;*
- *$r : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^k \times \mathcal{S} \to \Re$ is the reward function;*
- *$\mathcal{T}$ is the set of decision epochs.*

Limiting a Dec_POMDP to be fully joint observable, i.e., the combined information held by all agents determines the system state completely, results in a decentralized Markov decision process, i.e., a Dec_MDP.

Controlling a system of cooperative agents in a decentralized way means finding a *decentralized joint policy*, that determines what policy will be followed by each agent. Bernstein et al. [3] showed that finding an optimal decentralized joint policy with respect to the expected total reward (and assuming no communication) is NEXP-hard.

At the other extreme, there are agents who can communicate at free cost, i.e., without any effective bound or cost on the communication. Decentrally controlling a system with free communication is known to be P-SPACE [5].

In this work, we extend those complexity results to include approximate solutions to decentralized control of cooperative agents, encompassing both no communication and absolutely free communication.

2. MULTI-PROVER INTERACTIVE PROTOCOLS

Multi-Prover Interactive Protocols (MIPs) [1] are a computational model used to characterize complexity classes of languages.

Definition 2. [2] *Let $P_1, \ldots, P_k$ be Turing machines (*the provers*) which are computationally unbounded, and let V (*the verifier*) be a probabilistic polynomial time Turing machine. All machines share a read-only input tape. In addition, the verifier shares a communication tape with each prover P_i. There is no communication among the provers. $(P_1, ..., P_k, V)$ is a k-prover Interactive Protocol.*

Given an Interactive Protocol, we can characterize its corresponding language using the following definition, where n denotes the size of an input word x.

Definition 3. *$(P_1, ..., P_k, V)$ form a Multi-Prover Interactive Protocol for a language L:*

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

1. *If $x \in L$ then*
   $Pr(P_1, ..., P_k$ *make* V *accept* $x) > 1 - 2^{-n}$*;*
2. *If $x \notin L$ then for all provers* $P'_1, ..., P'_k$*,*
   $Pr(P'_1, ..., P'_k$ *make* V *accept* $x) < 2^{-n}$.

The initials MIP also denote the class of all languages that have Multi-Prover Interactive Protocols. The following terminology is standard:

Definition 4. *The functions $P_1, \ldots, P_k$, satisfying definition 3, will be called* the honest provers*. Any other collection of provers is* dishonest*;*

The provers win *indicates that the verifier "accepts" the word given as input, i.e., it decides that the word belongs to the language L;*

Perfect completeness means that for all $x \in L$ honest provers always win. The following should also be noted: if $x \in L$ then dishonest provers have a positive probability of winning, yet this probability is significantly smaller than the probability of the honest provers to win.

Theorem 1. *The following identities among classes of languages exist:*

1. $MIP_{k=2} = MIP_{k \geq 2} = NEXP$ [1]
2. $IP = MIP_{k=1} = PSPACE$ [7]

The languages that are included in the MIP class are also included in the complexity class NEXP, and vice versa: deciding whether a word is in a language that is in the MIP class will take non-deterministic exponential time. In particular, this is true for a Multi-Prover Interactive Protocol with only two provers. On the other hand, having a single prover, we can only decide PSPACE languages.

MIP retains its complexity under severe constraints on its components. The following result is due to Feige [4].

Theorem 2. *Let us denote by 'pfa' a verifier that uses O(1) working tape space. Then MIP(pfa)=MIP.*
That is, for any language decidable by a multi-prover protocol in polynomial time, there exists a protocol with a constant space-bounded verifier, that decides the same language.

3. APPROXIMATE SOLUTIONS TO DECENTRALIZED CONTROL

To define an approximate joint policy, we assume that we want to obtain a joint reward value R, and we assume that there is a joint policy π that achieves a joint reward r. Then, π is said to have relative error $E(R, \pi) = \frac{|r-R|}{\max\{r,R\}}$.

Definition 5. *A joint policy π is an ϵ-approximate policy if π yields an expected joint reward with relative error less than ϵ, i.e., $E(R, \pi) < \epsilon$.*

Theorem 3. *Given Dec_MDP_2, ϵ and R, deciding whether there exists an ϵ-approximate policy, i.e., a joint policy π with relative error $E(R, \pi) < \epsilon$, is NEXP-hard.*

Proof. A full proof of this theorem can be found in [6].
Assume a language L and its corresponding multi-prover protocol are given. We construct a transformation from the protocol into a Dec_MDP problem.

The reduction satisfies the following condition: if there exists an ϵ-approximate policy for the maximal reward value, then there exist provers who can convince the verifier with probability close to 1 that an input word x belongs to the language L. $\square$

4. DISCUSSION AND CONCLUSIONS

Multi-Prover Interactive Protocols were introduced as a model for multiagent systems. The modeling power of this approach is limited to systems with no inter-agent communication, and polynomial environmental complexity. A reduction between the MIP model and the Dec_MDP model was presented, borrowing from complexity results known for MIPs. This reduction provides a lower complexity bound for solving a Dec_MDP problem. It extends the results obtained in Bernstein et al. [3], concluding that both optimal and ϵ-approximate history-dependent joint policies are NEXP-hard to find in Dec_MDPs. We also showed that multiagent systems with collective partial observability and free communication are PSPACE-hard for approximate solutions.

Introducing MIPs as a model for cooperative multiagent systems demonstrates the complexity of information sharing scenarios at their extremes: when agents are completely isolated and when they fully share information, and when agents seek exact or approximate solutions. In both exact and approximate solutions, isolation among agents manifests itself as a jump in the complexity of system control. This complexity jump is an inherent property of the system. The complexity of finding a solution does not depend on whether we are seeking an exact or approximate result, but rather on a specific internal characteristic of the system, i.e., how information is distributed among the participating agents. An open question is whether the complexity jump is a true phase transition, and if so, how much information the agents have to share to prevent this increase in complexity.

Acknowledgments

C.V. Goldman is supported in part by AFOSR under Grant No. F49620-03-1-0090. Any opinions finding and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of AFOSR.

5. REFERENCES

[1] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[2] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of STOC-88*, pages 113–131, 1988.

[3] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[4] U. Feige. *Alternative models for zero knowledge interactive proofs.* PhD thesis, The Weizmann Institute of Sience, Rehovot, Israel, 1990.

[5] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16:389–423, 2002.

[6] Z. Rabinovich, C. V. Goldman, and J. S. Rosenschein. Non-approximability of decentralized control. Technical Report 2002-29, Leibniz Center for Computer Science, Hebrew University, Jerusalem, 2002.

[7] A. Shamir. IP=PSPACE. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Sciences*, pages 11–15, 1990.