Bounded Rationality in Multiagent Systems Using Decentralized Metareasoning

Shlomo Zilberstein
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
shlomo@cs.umass.edu

Alan Carlin
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
acarlin@cs.umass.edu

Abstract

Metareasoning has been used as a means for achieving bounded rationality by optimizing the tradeoff between the cost and value of the decision making process. Effective monitoring techniques have been developed to allow agents to stop their computation at the "right" time so as to optimize the overall time-dependent utility of the decision. However, these methods were designed for a single decision maker. In this paper, we analyze the problems that arise when several agents solve components of a larger problem, each using an anytime algorithm. Metareasoning is more challenging in this case because each agent is uncertain about the progress made so far by the others. We develop a formal framework for decentralized monitoring of decision making, establish the complexity of several interesting variants of the problem, and propose solution techniques for each case.

1 Introduction

The challenge of decision making with uncertain information and limited resources has attracted significant attention in philosophy, psychology, economics, and artificial intelligence. In the social sciences, the focus has been on developing *descriptive* theories of human decision making–theories that explain how people make decisions in the real world, coping with uncertainty and limited amount of time [6]. Work in artificial intelligence has produced several *prescriptive* theories and agent architectures that can take into account the computational cost of decision making [4, 9, 12, 16, 17]. The idea that the cost of decision making must be factored into the decision making process was introduced by Herbert Simon in the 1950's. His notion of "satisficing" has inspired research in many disciplines including AI. Much of the work so far has focused on a single decision maker–work on bounded rationality in group decision making has been relatively sparse.

To some extent, any approximate reasoning framework could be viewed as a form of bounded rationality. But unless one can establish some constraints on decision quality, such interpretations of bounded rationality are not very interesting. It seems more beneficial to define bounded rationality as an optimization problem constrained by the availability of knowledge and computational resources. One successful approach is based on decision-theoretic principles used to monitor the base-level decision procedure, structured as an *anytime algorithm*. Such decision procedures can be stopped at any time and provide an approximate solution, whose expected quality improves over time. It has been shown that the monitoring problem can be treated as a Markov decision process (MDP) and it can be solved optimally offline and used to optimize decision quality with negligible run-time overhead [8]. This approach to bounded rationality relies on *optimal metareasoning* [13]. That is, an agent is considered bounded rational if it monitors and controls its underlying decision making procedure optimally so as to maximize the comprehensive value of the decision. Additional formal approaches to bounded rationality have been proposed. For example, *bounded optimality* is based on a construction method that yields the best possible decision making program given a certain agent architecture [12]. The approach implies that a bounded rational agent will not be out-

performed by any other agent running on the same architecture. This is a stronger guarantee than optimal metareasoning, but it is also much harder to achieve.

Extending these computational models of bounded rationality to multiagent settings is hard. Even if one assumes that the agents collaborate with each other–as we do in this paper–there is an added layer of complication. There is uncertainty about the progress that each agent makes with its local problem solving process. Thus the metareasoning process inherently involves non-trivial coordination among the agents. One existing approach for meta-level coordination involves multiple agents that schedule a series of interrelated tasks [11]. As new tasks arrive, each agent must decide whether to deliberate on the new information and whether to negotiate with other agents about the new schedule. Each agent uses an MDP framework to reason about its deliberation process. However, the coordination across agents is handled by negotiation, not by the MDP policy.

In this paper, we extend optimal metareasoning techniques to collaborative multiagent systems. We consider a decentralized setting, where multiple agents are solving components of a larger problem by running multiple anytime problem solving algorithms concurrently. The main challenge is for each individual agent to decide when to stop deliberating and start taking action based on its own partial information. In some settings, agents may be able to communicate and reach a better joint decision, but such communication may not be free. We propose a formal model to study these questions and show that decentralized monitoring of anytime computation can be reduced to the problem of solving a decentralized MDP (Dec-MDP) [3]. Different monitoring strategies correspond to different types of Dec-MDPs with different computational complexity. Finally, we evaluate the performance of the approach on some decentralized decision making domains.

2 Decentralized Metareasoning

We focus in this paper on a multiagent setting in which a group of agents is engaged in collaborative decision making. Each agent solves a component of the overall problem using an *anytime algorithm*. While there is uncertainty about future solution quality, it increases with computation time according to some *probabilistic performance profile*. The purpose of metareasoning is to monitor the progress of the anytime algorithms and decide when to stop deliberation.

Definition 1. *The* decentralized monitoring problem *(DMP) is defined by a tuple* $< Ag, Q, A, P, U, C_L, C_G, T >$ *such that:*

- Ag *is a set of agents.*
- $Q_1, Q_2, ..., Q_n$ *are sets of discrete quality levels for agents* $1..n$*. At each step* t*, we denote the vector of agent qualities by* $\vec{q}^{\,t}$*, or more simply by* $\vec{q}$*, where* $q_i \in Q_i$*. Components of* $\vec{q}^{\,t}$ *are qualities for individual agents. We denote the quality for agent* i *at time* t *by* q_i^t*.*
- $\vec{q}^{\,0}$ *is a joint quality at the initial step, known to all agents.*
- A *is a set of metalevel actions available to each agent: "continue", "stop", "monitorL", and "monitorG". The actions monitorL and monitorG represent "monitor locally" and "monitor globally" respectively.*
- T *is a finite horizon representing the maximum number of time steps in the problem.*
- P_i *is the transition model for the "continue" action for agent* i*. We will simply use notation* P *when* i *is implied by the context. For all* $i, t \in \{1..T-1\}$*,* $q_i^t \in Q_i$*, and* $q_i^{t+1} \in Q_i$*,* $P(q_i^{t+1}|q_i^t) \in [0,1]$*. Furthermore,* $\Sigma_{q_i^{t+1} \in Q_i} P(q_i^{t+1}|q_i^t) = 1$*. We assume that the transitions of any two agents* i *and* j *are independent of each other, that is,* $P(q_i^{t+1}|q_i^t, q_j^t) = P(q_i^{t+1}|q_i^t)$*.*
- $U(\vec{q}, t)$ *is a utility function that represents the value of solving the overall problem with quality vector* $\vec{q}$ *at time* t*.*
- C_L *and* C_G *are the costs of the local monitoring and global monitoring actions respectively.*

Each agent solves a component of the overall problem using an anytime algorithm. Unless a "stop" action is taken by one of the agents, all the agents continue to deliberate for up to T time steps. It is often useful to consider a special class of utility functions defined below.

Although the framework in this paper will apply to all DMPs, we are motivated by problems where the utility decreases as a function of time, but increases as a function of quality, and the transition model specifies that quality monotonically increases with each time step. Thus, agents must decide

whether to accept the current solution quality or continue deliberation, which will result in a higher solution quality but also a higher cost of time.

At each time step, agents decide which option to take, to "continue", "stop", or "monitor" globally or locally. If all the agents choose to "continue", then the time step is incremented and solution quality transitions according to P. If any agent chooses to "stop", then all agents are instructed to cease computation before the next time step, and the utility $U(\vec{q}, t)$ of the current solution is taken as the final utility. If an agent chooses "monitorL", then a cost of C_L is subtracted from the utility (for each agent that chooses monitorL). If any agent chooses "monitorG", a single cost of C_G is subtracted from the utility. After an agent chooses to monitor, it must then choose whether to continue or stop, at the same time step.

Agents are assumed to know the initial quality vector $\vec{q}^{\,0}$. An agent has no knowledge about quality in later time steps, unless a monitoring action is taken. The "monitorL" action monitors the local quality; when agent i takes the "monitorL" action at time t it obtains the value of q_i^t. However, it still does not know any component of $\vec{q}_{-i}^{\,t}$. A "monitorG" action results in communication among*all* the agents, after which they all obtain the global quality $\vec{q}^{\,t}$.

3 Local Monitoring

We start with the restricted case in which $C_G = \infty$, thus no global monitoring occurs. Each agent must decide whether to continue its anytime computation, stop the entire decision making process immediately, or to monitor its progress locally at a cost C_L, and then decide.

Complexity of local monitoring We have analyzed several scenarios involving local monitoring. When $C_L = 0$, each agent should choose to monitor locally on every step, since doing so is free. We show that even the simple case where $C_L = 0$, $C_G = \infty$, and number of agents is fixed, the problem of finding a joint optimal policy is NP-hard, and that with $C_L = k$, the DMP problem is NP-complete.

Lemma 1. *The problem of finding an optimal solution for a DMP with a fixed number of agents, $|Ag|$, $C_L = 0$ and $C_G = \infty$ is NP-hard.*

The proof is based on reduction to DMP from Decentralized Detection [15]. Note that making local monitoring decisions in this case is trivial, but the overall stopping problem is still NP-hard.

We further show that when local monitoring has an arbitrary cost, the DMP problem is NP-complete.

Theorem 1. *The problem of finding an optimal solution for a DMP with $C_L > 0$ and $C_G = \infty$ is NP-complete.*

The NP-completeness proof is based on a reduction of the DMP problem to a transition-independent Dec-MDP, known to be NP-complete [7].

Myopic greedy solution We first derive a simple polynomial solution to the local monitoring problem based on a greedy approach. According to this approach, each agent continues its local computation as long as the marginal value of continued problem solving is positive. It extends a similar approach that proved useful in monitoring single-agent deliberation [17, 18]. In our multiagent setting, the greedy approach considers the other agents to be part of the environment, assuming that they always continue, and never monitor or terminate. We describe the technique from a single agent's point of view, assuming that each agent is executing this algorithm simultaneously. The approach is myopic in the sense that it does not take into account future meta-level decisions about continuing/stopping the computation.

We first extend the single step probabilistic performance profile of each agent, into a multistep performance profile Pr, where $\Delta t \in [1..T{-}1]$ is some fixed duration. This extension is straightforward.

Definition 2. *A dynamic local performance profile of agent i, $Pr_i(q_i'|q_i, \Delta t)$, denotes the probability of agent i getting a solution of quality q_i' by continuing the algorithm for time interval Δt when the currently available solution has quality q_i.*

Because of the transition independence among the agents, the global multistep performance profile is simply the product of the local ones.

Definition 3. *A myopic greedy estimate of the expected value of computation (MEVC) for agent i continuing Δt steps at time t, given its local solution quality q_i^t, is:*

$$MEVC(q_i^t, t, \Delta t) = \sum_{\vec{q}^{\,t}} Pr(\vec{q}^{\,t}|q_i^t, t)\Big(\sum_{\vec{q}^{\,t+\Delta t}} Pr(\vec{q}^{\,t+\Delta t}|\vec{q}^{\,t}, \Delta t)U(\vec{q}^{\,t+\Delta t}, t+\Delta t) - U(\vec{q}^{\,t}, t)\Big)$$

The first probability is the expectation of the current global state, given the local state, and the second probability is the distribution of future global quality given the current global quality. Thus, *MEVC* is the difference between the expected utility after continuing for Δt more steps, and the current expected utility from the point of view of agent i.

When $C_L = 0$, agents could first monitor their local quality and then continue computation as long as MEVC > 0 for $\Delta t = 1$, and repeat this assessment in each step. An less myopic approach is to evaluate MEVC$(q_i^t, t, \Delta t)$ for every Δt and continue as long as it is positive for *some* Δt.

When $C_L > 0$, it is beneficial to postpone local monitoring. When the decision is to continue the computation, it is necessary to decide how many steps should be performed before the next local monitoring will occur. We call this policy a cost-sensitive monitoring policy.

Definition 4. *A cost-sensitive monitoring policy, $\Pi_i(q_i, t)$, is a mapping from time step t and local quality q_i to a monitoring decision $(\Delta t, m)$ such that Δt represents the additional amount of time to allocate to the anytime algorithm, and m is a binary variable that represents whether to monitor at the end of this time allocation or to stop without monitoring.*

It is relatively easy to extend the dynamic programming approach from [8] to this multiagent case and derive local monitoring policies that factor the cost of local monitoring.

Modeling the other agents The myopic greedy solution optimizes each agent's decision based on its local information, but it does not account for the fact that the other agents are also decision makers who monitor the situation. In fact, it is often beneficial to rely on other agents to stop the deliberation process. For example, consider a situation in which the stopping time depends highly on the completion of a critical task performed by agent j. It is better to let agent j–the only one who can monitor the progress made with the critical task–make the stopping decision.

To better address the interaction between the agents, we can exploit the fact that the local monitoring problem can be reduced to solving a transition-independent Dec-MDP (TI-Dec-MDP), based on the construction used to prove Theorem 1. Solving a TI-Dec-MDP is much harder than implementing the above myopic greedy approach, but it provides an *optimal* solution. Several algorithms have been developed in recent years for solving TI-Dec-MDPs. Among them, the coverage set algorithm [2] and the bilinear programming approach [10] are the most effective ones. In our experiments, we used the bilinear programming approach, which works particularly well and is easy to implement. We first convert the problem to a transition independent Dec-MDP, and then prune "impossible" state-actions, for example inconsistent states in which $t_i^0 > t_i$. Then we convert the resulting problem into a bilinear program and use the efficient succesive approximation algorithm developed by Petrik and Zilberstein [10]. Although bilinear problems are NP-complete in general, in practice performance depends on the sparsity of the reward structure, which the algorithm exploits.

4 Global Monitoring

Next, we examine the case where agents can perform global monitoring by communicating with each other (imposing cost to the network). We analyze the case where $C_L = 0$ and $C_G = k$, where k is some constant. We show that this problem is NP-complete as well, by reducing it to a Dec-MDP-Comm-Sync [1]. A Dec-MDP-Comm-Sync is a transition-independent Dec-MDP with an additional property: after each step, agents can decide whether to communicate or not. If they do not communicate, agents continue onto the next step as with a typical transition-independent Dec-MDP. If any agent decides to communicate, then all the agents exchange knowledge and learn the global state. However, a joint cost of C_G is assessed for communicating. Agents form joint plans after communication. The portion of the joint plan formed by agent i after step t is denoted π_i^t.

Theorem 2. *The DMP problem with $C_L = 0$ and some constant C_G is NP-complete.*

The proof of NP-hardness is similar to Lemma 1. To show that the problem is in NP, we reduce the problem to that of finding the solution of a Dec-MDP-Comm-Sync [1]. In particular, the following Dec-MDP-Comm-Sync can be created from a DMP with $C_L = 0$:

- F^i is a new "terminal" state for each time step for agent i.
- S^i is the set of $\{q_i^t\}$ levels for agent i; the global state space is $\prod_i (S^i \cup \{F^i\})$
- A^i is { "continue", "terminate" }; the joint action set is $\prod_i A^i$
- The transition model:
  $P(q_i^t, \text{continue}, q_i^{t+1}) = P(q_i^{t+1}|q_i^t)$; $P(q_i^{t_1}, \text{continue}, q_i^{t_2}) = 0, \forall (t_2 \neq t_1 + 1)$
  $P(q_i^t, \text{terminate}, f_i) = 1, \forall q_i^t \in S_i$.
- The reward function $R(\vec{q}^t, \{A_i\}) = U(\vec{q}, t)$ if $A_i =$terminate for some i; 0 otherwise.
- The horizon T is the same as T from the DMP.
- The cost of communication is C_G.

It is straightforward to verify that this reduction is polynomial. Having represented the DMP problem as a Dec-MDP-Comm-Sync, we can use solution techniques from the literature to solve the problem [1]. One effective myopic approach to making communication (i.e., global monitoring) decisions is based on computing the net value of communication actions. The net value is based on the value of information obtained (the increase in expected utility of the new policy) minus the associated cost (C_G). An improvement of this myopic approach that considers postponing communication as well as the fact that other agents may initiate communication has shown to produce very good results [1]. We extended this technique to compute policies for global monitoring.

5 Experiments

We experimented with two different decentralized decision making scenarios. The first scenario involved a decentralized maximum flow problem where two entities must each solve a maximum flow problem in order to supply disparate goods to a customer. To estimate the transition model P in the DMP, we profiled the performance of an anytime maximum flow algorithm (based on Ford Fulkerson [5], through Monte Carlo simulation). The flow network was constructed randomly on each trial, with each edge capacity in the network drawn from a uniform distribution. The second example was the Rock Sampling domain, borrowed from the POMDP planning literature [14]. In this planning problem, two rovers must each form a plan to sample rocks, maximizing the joint value of the samples. However, the location of the rocks are not known until runtime, and thus the plans cannot be constructed until the rovers are deployed. We used the HSVI algorithm for POMDPs as the planning tool [14]. HSVI is an anytime algorithm whose performance (error bound) is constructed and reported at runtime.

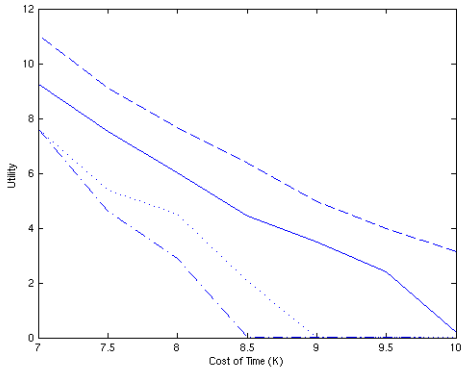

Figure 1: Expected utility vs. cost of time of non-myopic local monitoring. The costs of monitoring for the plots (top to bottom) are .5, 4, 7, and 10. As cost of time increases, utility decreases. The slope is more negative for higher costs of monitoring.

For each of these examples, we conducted separate experiments for local and global monitoring, using $C_G = \infty$ in local monitoring and $C_L = 0$ in global monitoring. For local monitoring, the decentralized anytime problem was converted to a Dec-MDP and solved using the bilinear program. For global monitoring, the problem was converted to a Dec-MDP-Comm-Sync and solved. We also ran experiments based on a direct extension of [8], where each agent chooses its cost-sensitive monitoring policy by treating the other agents as if they were part of the environment.

Due to space limitation, we only present a small sample of the results. Figure 1 plots value versus the cost of time (K) for 4 different costs of local monitoring on the Rock Sampling problem. The dashed line (lowest) represents a cost of monitoring of 10. The dotted-dashed line (highest) represents a cost of monitoring of 0.5. As expected, for a constant cost of time, a higher cost of monitoring results in a lower quality solution. The drop-off is monotonically decreasing and roughly linear, with higher cost of monitoring resulting in a more negative slope. The extreme end points of these graphs represent simple cases. As one proceeds leftwards, the cost of time goes down, ultimately reaching the point at which the agents should always continue the computation until completion and no monitoring decisions are needed. As one proceeds rightwards, the cost of time grows, ultimately reaching the point at which agents should stop on the first step and again, no monitoring decisions are needed.

The monitoring approach produces a policy whose value changes roughly linearly between these two extreme points, suggesting that it factors effectively the costs of time and monitoring.

6 Conclusion

We analyze in this paper the problem of coordination of deliberation interruption among multiple agents, when each agent solves a component of the overall decision problem using an anytime algorithm. To optimize the overall decision quality, agents need to monitor their own progress and occasionally monitor the global progress by entire team. We show that the local and global monitoring problems can be reduced to stochastic planning problems that can be represented as different variants of Dec-MDPs. We use these reductions to establish the computational complexity of monitoring and in order to solve the problem in two realistic scenarios. The results show that existing Dec-MDP solution techniques can be used effectively for decentralized meta-level control of deliberation processes in multiagent settings.

Currently, only the myopic greedy approach to local monitoring works well for more than two agents. Extending these techniques, particularly the bilinear solver, to settings that involve more than two agents remain an important challenge for future work. We are also interested in solving the monitoring problem when local or global solution quality are only partially observable. General Dec-POMDP algorithms that we are currently developing offer promising solution methods for this more general case.

References

[1] R. Becker, A. Carlin, V. Lesser, and S. Zilberstein. Analyzing myopic approaches for multi-agent communication. *Computational Intelligence*, 25(1):31–50, 2009.

[2] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized Markov decision processes. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.

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

[4] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, 49–54, 1988.

[5] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[6] G. Gigerenzer, P. M. Todd, and ABC Research Group. *Simple Heuristics That Make Us Smart*. Oxford: Oxford University Press, 1999.

[7] C. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[8] E. Hansen and S. Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126(1-2):139–157, 2001.

[9] E. Horvitz. Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of Third Workshop on Uncertainty in Artificial Intelligence*, 429–444, 1987.

[10] M. Petrik and S. Zilberstein. A bilinear approach for multiagent planning. *Journal of Artificial Intelligence Research*, 35:235–274, 2009.

[11] A. Raja and V. Lesser. A framework for meta-level control in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 15:147–196, 2007.

[12] S. Russell, D. Subramanian, and R. Parr. Provably bounded optimal agents. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 575–609, 1993.

[13] S. Russell and E. Wefald. Principles of metareasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 400–411, 1989.

[14] T. Smith and R. Simmons. Heuristic search value iIteration for POMDPs. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 520–527, 2004.

[15] J. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, 30(5):440–446, 1985.

[16] M. P. Wellman. *Formulation of Tradeoffs in Planning under Uncertainty*. London: Pitman, 1990.

[17] S. Zilberstein. Operational rationality through compilation of anytime algorithms. Ph.D. Dissertation, Computer Science Division, University of California, Berkeley, 1993.

[18] S. Zilberstein and S. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82(1-2):181–213, 1996.

[19] S. Zilberstein. Metareasoning and bounded rationality. In M. Cox and A. Raja (Eds.), Metareasoning: Thinking about Thinking, MIT Press, forthcoming.