

Identifying the Scope of Modeling for Time-Critical Multiagent Decision-Making*

Sanguk Noh

Computer Science Department
University of Missouri-Rolla
nohs@umr.edu

Piotr J. Gmytrasiewicz

Dept. of Computer Science and Engineering
University of Texas at Arlington
piotr@cse.uta.edu

Abstract

Decision-making in multiagent settings requires significant computational resources. Agents need to model each other to decide how to coordinate – this sometimes may require solving nested models of many other agents and may be impractical to perform in an acceptable time. In this paper, we investigate ways in which the agents can be equipped with flexible decision-making procedures to allow multiagent decision-making under time pressure. One of the techniques we implemented uses iterative deepening algorithm guided by performance profiles generated offline. When the interaction involves many agents, the algorithm iteratively enhances the quality of coordinated decision-making by incrementally adding the levels of nesting considered, but with the additional penalty of increased running time. To identify the appropriate scope of modeling online we use the concept of urgency which represents cost of delaying decisions. We validate our framework with experiments in a simulated anti-air defense domain. The contribution of our framework is that it endows our autonomous agents with flexibility to cope with time pressure in complex multiagent settings.

1 Introduction

In multiagent settings, deliberative decision-theoretic methods [Gmytrasiewicz *et al.*, 1998; Gmytrasiewicz and Durfee, 2000; Noh and Gmytrasiewicz, 1999a; 1999b; Vidal and Durfee, 1995] allow an agent to compute its best action, i.e., with the highest expected utility, given its available knowledge. The agent's knowledge may contain information about environment, about the agent itself, about other agents present, about what the other agents know, and so on. Therefore, when making coordinated decisions in complex scenarios with many agents

present, the agent may have to consider large and complex knowledge structures, which may lead to it failing to decide an action in an acceptable time.

In response to these issues we investigate methods which allow the agents to perform multiagent decision-making in time-critical situations. In our earlier work [Noh and Gmytrasiewicz, 1999b] we concentrated on compiling results of deliberative decision-making using machine learning and creating reactive rules that improved the agent's reactivity. This current paper focuses on limiting deliberation time by identifying the proper scope of knowledge used for decision-making. We wish to establish which information, assuming it is in the agent's possession in the first place, can be neglected and at what cost. In general, there are two approaches which we call simplification and approximation techniques:

- *Simplifications* involve neglecting a certain part of available information and computation associated with processing this information, while preserving the accuracy of deliberation. In general, there is no guarantee that a simplification technique will be applicable in any particular situation but, if it is, it can be used without sacrificing the quality of the result.
- *Approximations* involve limiting information considered while making decisions without the guarantee that the solution obtained is optimal.

In our framework, the simplification techniques remove computations that are found not to influence the quality of output. For example, if the agent finds that one of its actions is better than all other actions no matter what (we call it the dominant action) then any further computations of the values of actions become unnecessary. Other examples include memoization, and pruning certain parts of structured knowledge representation (defined rigorously in [Gmytrasiewicz and Durfee, 2000] and presented in Figure 1), if it can be proven that they do not influence the solution. All simplifications reduce the computation time needed for deliberation without significant overhead. The approximation techniques determine what computational resources should be used for agent's decision making. These methods consider a tradeoff between decision quality and computation time.

*This research has been sponsored by the Office of Naval Research Artificial Intelligence Program under contract N00014-95-1-0775 and by the National Science Foundation CAREER award IRI-9702132.

In other words, the objective of approximation techniques is to provide autonomous agents with the boundedly rational decision, given the real-time demands of the situation at hand. The approximation techniques include: (1) using compiled rules to constrain alternative actions, on which we reported in [Noh and Gmytrasiewicz, 1999b], and (2) using performance profiles and urgency of situation to determine the appropriate scope of modeling.

In this paper, we will explore the second of the approximation techniques. We will identify the appropriate scope of modeling of other agents based on performance profiles and the calculation of urgency. Toward this end, we focus on the amount of information that can be included in modeling other agents. We are inspired by techniques used by humans in real-life situations when they make coordinated decisions with a large number of individuals. Frequently, we consider and think about only a few agents in our immediate vicinity for the purpose of coordination. Similarly, our agents start by considering their closest neighbors first, and then broaden the scope by including more distant agents (in Figure 1 this would correspond to broadening of, what we call the modeling structure.) At the same time, our agents need to decide on how deep the modeling should proceed. Should one consider how the other agents model others, for example? While reasoning at deeper levels results in better decisions, it also requires more computation time.

Thus, the entire modeling space has the dimension of the *breadth*, which corresponds to the number of agents considered for coordination, and the *depth*, which corresponds to the reasoning level. Based on the structure of the modeling space, we present an iterative deepening algorithm which provides our agents with preemptible results according to different breadths and depths of the modeling space. This algorithm can be used offline to generate *performance profiles* [Dean, 1990; Zilberstein and Russell, 1996], which illustrate the performance vs. computation time required by processing increasingly more information. Further, we introduce the concept of urgency which represents the value of time given a situation at hand. For example, if a situation is very urgent, the agent should be ready to execute an action quickly, without much reasoning efforts. When the situation is not urgent, on the other hand, the agent can afford to consider more information and spend more deliberation time looking for better decisions. Our agents decide whether to act or think by computing urgency.

2 Related Work

We explore the ways to reduce the complexity of deliberative decision-making procedures that might cause autonomous agents to fail in accomplishing their tasks under time pressure, and, at the same time, to preserve the accuracy of deliberation as much as possible. Due to the impracticality of an agent with perfect rationality, it has been proposed that bounded optimality can be a way to yield a solution within a specific time in spite

of sacrificing the optimality of the decision to some degree. Thus, our work builds on efforts by several other researchers who focus on making autonomous systems tractable when there is no sufficient time to produce optimal solutions in complex environments.

One approach uses a purely reactive reasoning procedure. The goal of reactive systems is to directly respond to the condition with a predefined action. Although such a system is able to react very quickly to environments, plans will be useless if problem varies and re-planning is needed. Further, if no protocol or overall plan can be established explicitly in advance, this reactive system is unrealistic or likely to lock the agents into suboptimal forms of behavior. However, our deliberative reasoning procedure provides rational behaviors in dynamic environments even when plans are not pre-defined.

Other rigorous efforts have focused on bounding the rationality of deliberative agents, which limits the resources available to agents. These approaches include work on the use of meta-reasoning procedures to control inference [Horvitz, 1988; Russell and Wefald, 1991], and anytime algorithms [Dean, 1990; Zilberstein and Russell, 1996]. Meta-reasoning and anytime algorithms balance the costs and the benefits of continued deliberation [Good, 1971]. They perform online computation of expected utilities of additional deliberation. In contrast, our approach incorporates offline empirical data (performance profiles) into the decision of whether to act or think. By using the information calculated offline, our meta-reasoning is faster than their meta-reasoning.

In the field of multiagent coordination, there have been several approaches to support practical decision-making. Lesser [Lesser, 1998] proposes a domain independent agent architecture for a large-scale multiagent environment, which deals with adaptation to resource availability, tradeoffs between resources and quality, complex interdependencies among agents, and so forth. For further research, the components of their software infrastructure should be implemented and tested in sophisticated, real-time multiagent applications. Durfee [Durfee, 1999] suggests strategies for making coordination practical when agents use the Recursive Modeling Method (RMM) [Gmytrasiewicz and Durfee, 2000] as a nested model framework. These strategies include simplifying utility computations, limiting possible choices, treating multiple agents as one, and so on. This work is similar to our current work in that both approaches use RMM to make agents' rational decisions. However, we identify the scope of modeling knowledge processed by an agent considering the urgency of the situation at hand whereas Durfee simplifies (or ignores) computational resources as much as possible. Further, our work concentrates on offline efforts to make real-time decision tractable but most of Durfee's strategies for practical coordination have to be executed online. Lastly, in our own work [Noh and Gmytrasiewicz, 1999b], we propose an adaptive and deliberative agent architecture consisting of compiled and deliberative decision procedures. We compile the results of deliberative decision-making into a set of reactive

condition-action rules with numerous machine learning algorithms. An autonomous agent can use the compiled knowledge and constrain the number of alternative actions considered by excluding the ones that are likely to be suboptimal.

3 Control of Reasoning: Act or Think

To show the correlation among the amount of information, decision quality, and computation time within our framework, we compare expected utilities of actions arrived at using different amounts of computational resources, and executed with various time delays. Let N be the set of agents and $A_i, i \in N$, be the set of actions of agent i . The expected utility of the best action, α , arrived at using the body of information E and executed at time t , is

$$EU(\alpha^E, t) = \max_{a_i^j \in A_i} \sum_k P(S_k | E, t, a_i^j) U(S_k) \quad (1)$$

where

- $P(S_k | E, t, a_i^j)$ is the probability over the possible outcome states, S_k , given the available evidence E and the action a_i^j executed at time t .
- $U(S_k)$ is the utility of the resulting state S_k .

Having arrived at the best action, the agent i can execute it immediately, or keep on computing by increasing the body of knowledge considered, and including additional information, e , hoping to arrive at better decision. If it were to keep computing with information $E + e$, the result, α^{E+e} would be available for execution at time $t + t'$, where t' is the computation time used. The decision of whether to keep computing with extra information or whether to execute currently best action comes down to the following comparison:

$$EU(\alpha^{E+e}, t + t') > EU(\alpha^E, t) \quad (2)$$

If the expected utility of possibly higher quality, but delayed, action is greater than the utility of acting on the current best, the agent should keep on computing based on the increased body of information. The challenge is to determine whether the above inequality holds before the computation of α^{E+e} is actually executed. Our approach to flexible decision-making is to use the performance profiles, generated offline, and the notion of value of time, or urgency, to determine whether to act or think. Using performance profiles one can estimate $EU(\alpha^{E+e}, t)$, i.e., the expected value of decision based on information $E + e$, executed currently. Since the actual α^{E+e} is not available, we postulate that the currently available α^E be used to estimate the $EU(\alpha^{E+e}, t + t')$ as follows:

$$EU(\alpha^{E+e}, t + t') = EU(\alpha^E, t) - [EU(\alpha^E, t) - EU(\alpha^E, t + t')] \quad (3)$$

In the above, the value of delayed action α^{E+e} , is estimated based on the assumption that its quality

deteriorates¹ at the rate similar to that of α^E , which has been computed. Substituting the above to Equation 2 and rearranging terms, we get:

$$\begin{aligned} EU(\alpha^{E+e}, t) - EU(\alpha^E, t) &> \\ EU(\alpha^E, t) - EU(\alpha^E, t + t') \end{aligned} \quad (4)$$

The left hand side of the above inequality is the benefit of further computation, if it can be carried out instantaneously. This quantity can be estimated from the performance profiles. The right hand side is the current estimate of the value of the extra computation time t' . It is closely related to what we define as *urgency*, or the *value of time*, expressed as the rate with which the expected utility of the currently best alternative deteriorates with time. For instance, if the system slowly processes the data and the urgency is high, then the cost of time needed for computation is greater than the benefit of computation, and it is better not to compute further and act immediately. But if the computation time is short, the value of a unit of time is moderate, and the performance profile rises sharply, the extra computation is preferred to immediate action.

4 Performance Profiles of Iterative Deepening Algorithm

Our deliberative decision-theoretic method is the Recursive Modeling Method (RMM). We have implemented a full-blown version of RMM which allows an agent to compute its best action given all that is known about the other agents, about their states of knowledge, and so on. An agent using RMM represents the hierarchy of its modeling knowledge as a nested modeling structure with: (1) the agent's own decision-making situation (represented as a payoff matrix or as an influence diagram), (2) the next level containing the agent's view of other agents' situations (also payoff matrices or equivalent influence diagrams), (3) the deeper level representing the agent's view of how other agents model others, and so forth (as depicted in Figure 1).

The RMM algorithm, operating on the structure represented in Figure 1, can iteratively deepen its scope by including lower modeling levels. Doing so enhances the quality of decision-making as models nested deeper decrease uncertainties about the agents' beliefs about others in a multiagent domain. In the worst case, when none of the simplification methods is applicable, the complexity and run time of decision-making grow exponentially with the number of agents and the level of reasoning. That is why we consider the tradeoff between decision quality and computation time.

Using iterative deepening RMM algorithm, we generate performance profiles, which plot expected output quality versus computation time in proportion to the

¹In an urgent situation it is better to execute action sooner rather than later. I.e., the expected utility of action decreases as the time passes.

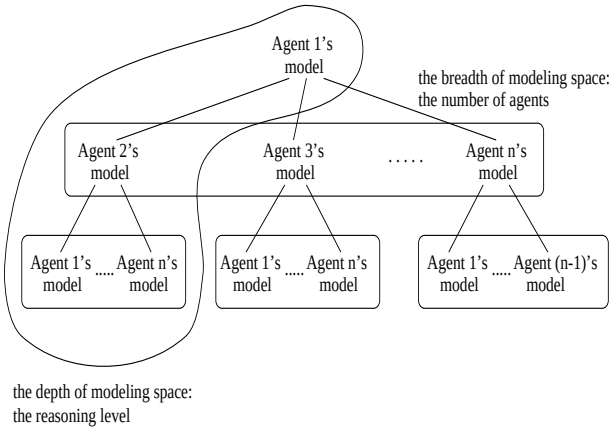


Figure 1: The space of recursively nested models.

breadth and the depth of modeling space. The data for performance profiles is collected offline through simulation of many specific decision-making situations.

Our model of multiagent decision-making under time pressure is the anti-air domain, which consists of a number of attacking targets and coordinating defense units, as depicted in Figure 2. The mission of the defense units is to attempt to intercept the attacking targets so as to minimize damages to own territory. The anti-air defense simulator [Noh and Gmytrasiewicz, 1999a] is written in Common LISP and built on top of the MICE simulator, running on a LINUX platform.

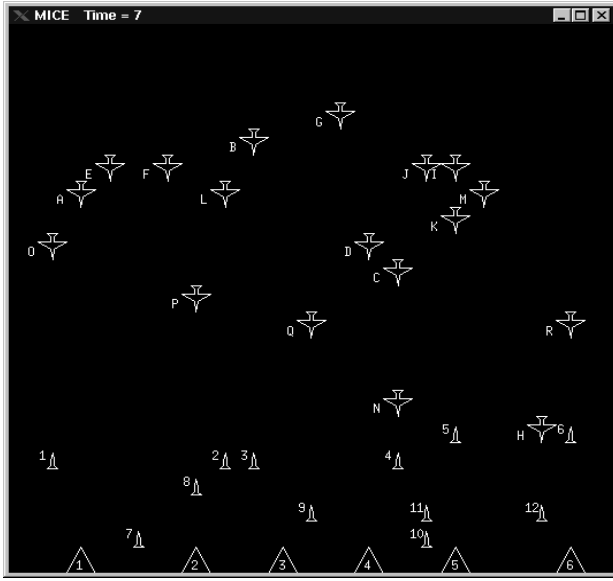


Figure 2: A complex anti-air defense scenario. There are six defense units and 18 incoming targets.

To construct performance profiles using the iterative deepening RMM, we consider six defense agents and six incoming targets.² The positions and warhead sizes of

²We simplified the scaled-up scenarios for clarity of pre-

targets are randomly generated, and the warhead sizes vary between 100 and 300. We measure the agent's performance in terms of total threat removed. The *total threat removed* is defined as the expected value of a sum of removed warhead sizes of attacking targets. Thus, if a target was chosen for interception, then it contributed (*interception_probability* $\times$ *warhead_size*) to the total threat removed. If a target was not intercepted, it did not contribute at all. The interception probability, $P(H_{ij})$, is dependent on the angle a_{ij} between the target j 's direction of motion and the agent i 's line of sight, as follows:

$$P(H_{ij}) = e^{-\mu a_{ij}} \quad (5)$$

where μ is an interceptor-specific constant (assumed here to be 0.01).

In the experiment, the breadth of modeling space ranges from two to six and the depth of modeling space varies between one and four. The algorithm begins with the simplest modeling space, that is, two agents and reasoning depth one and incrementally adds the amount of information up to six agents and depth four. For baseline performance, we include the performance of agents controlled by random and independent target selection (i.e., without modeling other agents) strategies. The experimental results of average performance (*total threat removed*) and computation time (*seconds* on a 500 MHz PentiumTM III machine) after 100 runs are summarized into Table 1 as discrete entries. In Table 1, the more total threat removed, the better performance.

The random agents, ignoring all information, showed the worst performance, 516.2, and spent the shortest computation time, 0.013. The performance (527.9) and the computation time (0.017) of the independent agents were similar to those of the random agents. The independent agent, maximizing its own expected utility without considering others, did not score well in this coordination experiment and obtained next to the worst performance. Compared the performance of the independent agent, the performance of the agent coordinating by considering the other neighboring agent, the amount of threat removed, 809.7, was drastically increased up by 53.4%. In this case, we clearly see that modeling other agents is advantageous for effective coordination.

As the depth or the breadth of modeling space increased, the performance of agents was improved and the computation time increased. When the number of agents considered for coordination was two or three, the performance gradually improved; the threat removed from our defense area increased from 809.7 to 859.9 when each defense agent considered the other agents, and increased from 906.2 to 1055.0 when it modeled two other agents. However, as more agents and further reasoning depth were included, the result of performance increase leveled off. The amount of improvement from 3 with depth 4 to

sentation here. This simplification is made possible by compiled rules we reported on earlier [Noh and Gmytrasiewicz, 1999b].

Table 1: Performance and computation time of the agents using random, independent and iterative deepening algorithm

Methods	Performance	Computation Time
Random	516.2 $\pm$ 126.6	0.013 $\pm$ 0.007
Independent	527.9 $\pm$ 175.3	0.017 $\pm$ 0.005
2 w/ depth 1	809.7 $\pm$ 105.1	0.040 $\pm$ 0.015
2 w/ depth 2	816.7 $\pm$ 108.9	0.056 $\pm$ 0.020
2 w/ depth 3	841.4 $\pm$ 82.1	0.058 $\pm$ 0.021
2 w/ depth 4	859.9 $\pm$ 57.9	0.062 $\pm$ 0.043
3 w/ depth 1	906.2 $\pm$ 98.3	0.364 $\pm$ 0.047
3 w/ depth 2	1001.3 $\pm$ 60.2	0.374 $\pm$ 0.053
3 w/ depth 3	1050.8 $\pm$ 46.3	0.381 $\pm$ 0.051
3 w/ depth 4	1055.0 $\pm$ 47.9	0.373 $\pm$ 0.054
4 w/ depth 1	1055.2 $\pm$ 48.9	3.036 $\pm$ 0.380
4 w/ depth 2	1055.4 $\pm$ 48.0	3.129 $\pm$ 0.501
4 w/ depth 3	1055.7 $\pm$ 50.8	3.168 $\pm$ 0.556
4 w/ depth 4	1055.9 $\pm$ 47.4	3.202 $\pm$ 0.570
5 w/ depth 1	1057.2 $\pm$ 48.0	32.533 $\pm$ 11.064
5 w/ depth 2	1057.4 $\pm$ 46.6	38.400 $\pm$ 7.026
5 w/ depth 3	1057.7 $\pm$ 46.3	37.675 $\pm$ 6.813
5 w/ depth 4	1058.1 $\pm$ 49.0	37.975 $\pm$ 7.108
6 w/ depth 1	1060.6 $\pm$ 45.3	144.776 $\pm$ 97.834
6 w/ depth 2	1060.8 $\pm$ 44.4	691.043 $\pm$ 166.245
6 w/ depth 3	1061.2 $\pm$ 44.3	689.943 $\pm$ 164.606
6 w/ depth 4	1061.5 $\pm$ 44.0	692.443 $\pm$ 164.804

6 with depth 4 in Table 1 was only 6.5(= 1061.5 – 1055.0) due to more information.

As more breadth and more depth were considered, our agents also spent more computation time. However, because of garbage collection in Lisp, the computation time in some cases did not increase in proportion to the reasoning depth: 3 with depth 4, 5 with depth 3, and 6 with depth 3. As more agents were added the run-times ranged over [0.040–0.062], [0.364–0.381], [3.036–3.202], and [32.533–38.400], respectively. When the six agents were considered for coordination, it should be noted that the computation time at reasoning depth two (691.043) took five times as long as that of reasoning depth one (144.776). Proceeding to more than depth one, the agents (6 with depth 2 – 4) needed to compute additionally five six-dimensional payoff matrices and so used relatively more runtime to construct and evaluate the payoff matrices. In that the dimension of payoff matrix increased according to the number of agents, the computation time spent depending on the number of agents was saliently different.

5 Determining the Scope of Modeling Using Urgency

Our definition of the urgency, as defined in Equation 4, gets at the intuition that an agent stands to lose utility as time progresses in an urgent situation if the agent is not doing anything to remedy the situation. For example, an air defense unit’s not trying to intercept the incoming threats may cause the increase in the probability of getting hit as the time progresses, because the

unit’s inaction may diminish its chances to successfully intercept the threats. Thus, if the agent’s expected utility is inversely proportional to the damages expected in the attack, the situation is urgent in that the defense agent loses utility as time progresses.

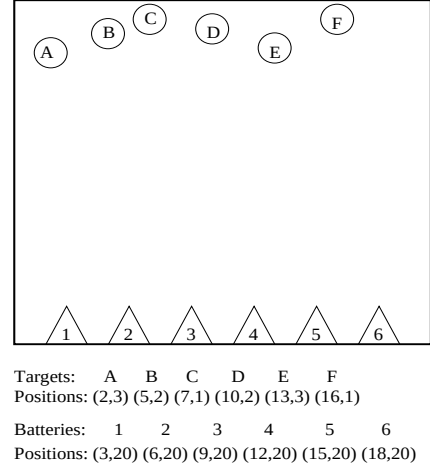


Figure 3: A simplified anti-air defense scenario. There are six defense units and six incoming targets.

As a simple example consider the scenario depicted in Figure 3. Assume that the targets proceed two grids down for one second. In this example, we focus on the agent 1’s decision-making situation. At the current time the average interception probability is

$$\text{At time } t, \Sigma_j P(H_{1j})/6 = 0.9499$$

After one second, the targets are coming closer to the defense agents by two grids, the angles between the battery’s line of sight and the targets increase, and the interception probability from the agent 1’s perspective decreases as follows:

$$\text{At time } t + 1, \Sigma_j P(H_{1j})/6 = 0.9483$$

In our settings as described in the previous section, the average total warhead size is 1,200(= 200 $\times$ 6). Given the interception probabilities at different time steps, the average total warhead size and the computation time, the urgency, or the cost of computation, is

$$\begin{aligned} EU(\alpha^E, t) - EU(\alpha^E, t + 0.017) &= \\ (0.9499 - 0.9483) \times 1,200 \times 0.017 &= 0.0326 \end{aligned}$$

Given the performance profiles in Table 1, the benefit of computation (the left hand side of Equation 4) is

$$EU(\alpha^{E+e}, t) - EU(\alpha^E, t) = 527.9 - 516.2 = 11.7$$

In this example scenario, therefore, since the benefit of computation is greater than the cost of computation, the agent should not take the immediately available action α of shooting at a randomly picked target at the initial time t , but it should consider the extra information

e (consisting of the agent's own decision-making situation), compute $EU(\alpha^{E+e}, t+t')$, and be ready to execute the best action at time $t + 0.017$.

It turns out that the break-even point, at which the agent should cease computation, is to model 3 agents with depth 3 in the example scenario. If the parameters of the particular situation at hand change, however, say when the missiles move faster, they are positioned closer, or if the computation is slower, the optimal scope of reasoning is lower and less computation will take place. In general, the factors of the calculation of value of time include the capability of data processing machine and the external factors imparting urgency to the situation.

We tested the notion of urgency in the anti-air defense settings which contain six defense units and six incoming targets. To identify the appropriate scope of modeling given the offline pre-computed performance profiles (Table 1), each defense agent calculated the benefit and the cost of computation for a level. As a result, after 100 runs, the defense agents stopped reasoning when the average breadth of modeling space was 3 and the average depth of modeling space was 3.017. That is, they considered the adjacent two agents with the reasoning depth just over three, on the average, for their coordinated decision-making. When the agents were under time pressure, thus, they could decide the proper modeling space by calculating their urgency, and reduce the scope and processing time to react faster.

6 Conclusions

The iterative deepening algorithm can be a useful tool that generates partial results according to the breadth and the depth of the modeling space. We generated offline performance profiles by using the iterative deepening algorithm in an anti-air defense domain. We think that the performance profiles for other coordination domains may be similar, but should be obtained separately offline. As the amount of breadth and depth of modeling information increased, the quality of decision was improved. We also extended our multiagent interaction model by introducing the concept of urgency. This notion presents the value of time in a specific situation, when decisions have to be made under time pressure and uncertainty. We integrated this concept into the flexible decision-making framework for identifying the proper scope of modeling information. The most promising aspect of this method is that the reasonable model structure can be employed to avoid wasteful computations and produce a practical decision during real-time.

We experimented with the iterative deepening algorithm and the concept of urgency in the anti-air defense domain to assess agent's decision-making quality. Anti-air defense is certainly a real-time task, and any overhead or loss of timely response can result in the destruction of coordinating agents and areas they are defending. The iterative deepening algorithm and the calculation of value of time prevented catastrophic failure and provided the best decisions in the high-stakes anti-air defense.

We will address the problem of what and how much offline performance profiles are necessary to gain the real-time advantage. For a more systematic evaluation of the framework, the performance profiles using the iterative deepening algorithm will be generated in various domains. In parallel, when insufficient data is available for offline simulation, we will test whether or not our framework produces deterioration in performance of the real-time deliberation.

Another problem to be answered is that we need to validate how each flexible decision-making method proposed can be integrated into a whole procedure. Even though we already tested whether the respective methods can be successfully applied to the time critical situation, we still need to experiment with the whole procedure selectively combined for various situations.

References

- [Dean, 1990] T. Dean. Decision-theoretic control of inference for time-critical applications. *Artificial Intelligence*, pages 1–28, November 1990.
- [Durfee, 1999] E. H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.
- [Gmytrasiewicz and Durfee, 2000] P. J. Gmytrasiewicz and E. H. Durfee. Rational coordination in multi-agent environments. *Autonomous Agents and Multiagent Systems Journal*, 3:319–350, 2000.
- [Gmytrasiewicz et al., 1998] P. J. Gmytrasiewicz, S. Noh, and T. Kellogg. Bayesian update of recursive agent models. *User Modeling and User-Adapted Interaction: An International Journal*, 8(1/2):49–69, 1998.
- [Good, 1971] I. J. Good. Twenty-seven principles of rationality. In V. P. Godambe and D. A. Sprott, editors, *Foundations of Statistical Inference*, pages 108–141. Holt, Rinehart, and Winston, 1971.
- [Horvitz, 1988] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the 7th AAAI*, pages 111–116, August 1988.
- [Lesser, 1998] V. R. Lesser. Reflections on the nature of multi-agent coordination and its implications for an agent architecture. *Autonomous Agents and Multi-Agent Systems*, 1(1):89–111, 1998.
- [Noh and Gmytrasiewicz, 1999a] S. Noh and P. J. Gmytrasiewicz. Implementation and evaluation of rational communicative behavior in coordinated defense. In *Proceedings of the 3rd Autonomous Agents*, pages 123–130, May 1999.
- [Noh and Gmytrasiewicz, 1999b] S. Noh and P. J. Gmytrasiewicz. Towards flexible multi-agent decision-making under time pressure. In *Proceedings of the 16th IJCAI*, pages 492–498, August 1999.
- [Russell and Wefald, 1991] S. J. Russell and E. H. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.
- [Vidal and Durfee, 1995] J. Vidal and E. Durfee. Recursive agent modeling using limited rationality. In *Proceedings of the 1st ICMAS*, June 1995.
- [Zilberstein and Russell, 1996] S. Zilberstein and S. J. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82(1):181–213, 1996.