Metareasoning and Bounded Rationality

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

Abstract

What role does metareasoning play in models of bounded rationality? We examine the various existing computational approaches to bounded rationality and divide them into three classes. Only one of these classes significantly relies on a metareasoning component. We explore the characteristics of this class of models and argue that it offers desirable properties. In fact, many of the effective approaches to bounded rationality that have been developed since the early 1980's match this particular paradigm. We conclude with some open research problems and challenges.

Computational models of bounded rationality

In the pursuit of building decision-making machines, artificial intelligence researchers often turn to theories of "rationality" in decision theory and economics. Rationality is a desired property of intelligent agents since it provides well-defined *normative* evaluation criteria and since it establishes formal frameworks to analyze agents (Doyle 1990; Russell and Wefald 1991). But in general, rationality requires making *optimal* choices with respect to one's desires and goals. As early as 1947, Herbert Simon observed that optimal decision making is impractical in complex domains since it requires one to perform intractable computations within a limited amount of time (Simon 1947; 1982). Moreover, the vast computational resources required to select optimal actions often reduce the utility of the result. Simon suggested that some criterion must be used to determine that an adequate, or satisfactory, decision has been found. He used the Scottish word "satisficing," which means satisfying, to denote decision making that searches until an alternative is found that is satisfactory by the agent's aspiration level criterion.

Simon's notion of satisficing has inspired much work within the social sciences and within artificial intelligence in the areas of problem solving, planning and search. In the social sciences, much of the work has focused on developing *descriptive* theories of human decision making (Gigerenzer 2000). These theories attempt to explain how people make decisions in the real-world, coping with complex situations, uncertainty, and limited amount of time. The answer is often based on a variety of heuristic methods that are used by people to operate effectively in these situations. Work within the AI community–which is the focus of this paper–has produced a variety of computational models that can take into account the computational cost of decision making (Dean and Boddy 1988; Horvitz 1987; Russell *et al.* 1993; Wellman 1990; Zilberstein 1993). The idea that the cost of decision making must be taken into account was introduced by Simon and later by the statistician Irving Good who used the term *Type II Rationality* to describe it (Good 1971). Good said that "when the expected time and effort taken to think and do calculations is allowed for in the costs, then one is using the principle of rationality of type II." But neither Simon nor Good presented any effective computational framework to implement "satisficing" or "type II rationality".

It is by now widely accepted that in most cases the ideal decision-theoretic notion of rationality is beyond our reach. However, the concept of satisficing offers only a vague design principle that needs a good deal of formalization before it can be used in practice. In particular, one must define the required properties of a satisficing criterion and the quality of behavior that is expected when these properties are achieved. AI researchers have introduced over the years a variety of computational models that can be seen as forms of bounded rationality. We start by dividing these models into three broad classes. We are particularly interested in the role that metareasoning plays in these theories.

Approximate reasoning

One of the early computational approaches to bounded rationality has been based on heuristic search. In fact, Simon has initially identified satisficing with heuristic search. In this context, heuristic search represents a form of approximate reasoning. It uses some domain knowledge to guide the search process, which continues until a satisfactory solution is found. This should be distinguished from *admissible* heuristic techniques such as A^* that are designed to always return the optimal answer. Admissible heuristic search is an important part of AI, but it has little to do with bounded rationality. The focus on optimal, rather than satisfying, solutions makes this type of heuristic search simply a more efficient way to find exact answers. Simon refers to another type of heuristic functions in which heuristics are used to select "adequate" solutions. Such heuristic functions are rarely admissible and the corresponding search processes are not optimal in any formal sense. Systems based on non-admissible heuristic functions are often harder to evaluate, especially when optimal decisions are not available.

Although it is often assumed that approximate reasoning is aimed at finding approximate answers to a given problem, it can take different forms. For example, the initial problem can be reformulated in such a way that reduces its complexity. The reformulation process could be approximate, yielding a new problem that is easier to solve because it does not retain all the details of the original problem. The resulting problem could then be solved efficiently and perhaps optimally and the obtained solution can then be used as an *approximate* solution for the original problem. One example of such a process–also referred to as approximate modeling–is when deterministic action models are used in planning, ignoring the uncertainty about action failures. Combined with suitable runtime execution monitoring, such an approach could be beneficial. In fact, the winner of a recent probabilistic planning competition was a planner based on these principles. Regardless of the form of approximation, approximate reasoning techniques can be complemented by some form of explicit or implicit metareasoning. Metareasoning in this context is a mechanism to make certain runtime decisions by reasoning about the problem solving–or object-level–reasoning process. This can be done either explicitly, by introducing another level of reasoning as shown in Figure 1, or implicitly, by pre-compiling metareasoning decisions into the object-level reasoning process at design time. For example, metareasoning has been used to develop search control strategies–both explicitly and implicitly. In some cases, the goal is specifically to optimize the tradeoff between search effort and quality of results (Russell and Wefald 1991). Thus, metareasoning could play a useful role in certain forms of approximate reasoning, but it is not–by definition–a required component.

While it is clear that any form of bounded rationality essentially implies that the agent performs approximate reasoning, the opposite is not necessarily true. Generally, frameworks for approximate reasoning do not provide any formal guarantees about the overall performance of the agent. Such guarantees are necessary to offer a satisfactory definition of bounded rationality and thus restore some sort of qualified optimality. So, we do not view a heuristic rule that proves useful in practice to be in itself a framework for bounded rationality if it does not have additional formal properties. The rest of this section describes two additional approaches that offer such precise properties.

Optimal metaraesoning

Since metareasoning is a component that monitors and controls object-level deliberation, one could pose the question of whether the metareasoning process itself is optimal. Optimality here is with respect to the overall agent performance, *given* its fixed object-level deliberation capabilities. This is a well-defined question that sometimes has a simple answer. For example, metareasoning may focus on the single question of when to stop deliberation and take action. Depending on how the base-level component is structured, the answer may or may not be straightforward. Optimal metareasoning has been also referred to as *rational metareasoning* (Horvitz 1989) and *metalevel rationality* (Russell 1995) to distinguish it from perfect rationality. This offers one precise form of bounded rationality that we will examine in further details in the next section.

It should be noted that optimal metareasoning can result in arbitrary poor agent performance. This is true because we do not impose upfront any constraints on the object-level deliberation process, in terms of either efficiency or correctness. Nevertheless, we will see later that this presents an attractive framework for bounded rationality and that performance guarantees can be established once additional constraints are imposed on the overall architecture.

Bounded optimality

Bounded optimality techniques seek to restore a stonger notion of optimality in decision making in the face of computational complexity and limited resources. That is, instead of building systems that can find "sufficiently good" answers, the goal is to find a maximally successful program that can compute these answers. Optimality is defined with respect to a particular space of possible implementations (Russell and Wefald 1991; Russell 1995).

Russell and Wefald (1991) say that an agent exhibits bounded optimality "if its program is a solution to the constraint optimization problem presented by its architecture." This approach marks a shift from optimization over actions to optimization over programs. The program is bounded optimal for a given computational device for a given environment, if the expected utility of the program running on the device in the environment is at least as high as that of all other programs for the device. When the space of programs is finite, one can certainly argue that a bounded optimal solution exists. Finding it, however, could be very hard.

Russell, Subramanian, and Parr (Russell *et al.* 1993) give an efficient construction algorithm that generates a bounded optimal program for a particular restricted class of agent architectures, in which a program consists of a sequence of decision procedures. The decision procedures are represented using condition-action rules. The authors admit that bounded optimality as defined above may be hard to achieve for most problems. They propose a weaker notion of *asymptotic bounded optimality* as a more practical alternative. The latter case requires that the program performs as well as the best possible program on any problem instance, provided that its computational device is faster by a *constant* factor.

To establish bounded optimality, the designer of the system–not the agent itself, is responsible to identify the agent reasoning architecture and to prove that the program satisfies the optimality conditions. In that sense, metareasoning does not play any significant role is this framework. Certainly there is no requirement that the agent itself be engaged in any form of metareasoning. As long as the agent program is shown to satisfy the optimality conditions, we have a bounded optimal agent.

One criticism of bounded optimality is that while the bounded rationality criterion is very well defined, it is very hard to achieve in practice as evidenced by the limited number of examples produced following the introduction of the

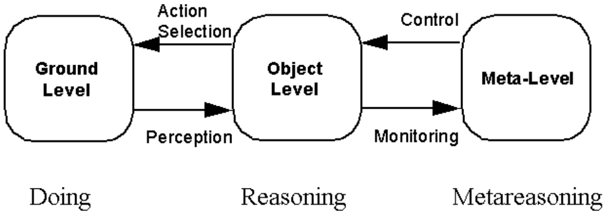

Figure 1: The view of metareasoning as monitoring and controlling object-level reasoning.

concept 15 years ago. Bounded optimality may well be the most precise formal approach to bounded rationality, but without further refinement it has little use in practice.

Bounded rationality as optimal metareasoning

We considered three basic approaches to achieve bounded rationality: approximate reasoning, optimal metareasoning, and bounded optimality. The latter two approaches represent very specific, well-defined solutions; everything else falls under the broad category of approximate reasoning. We argued that the first approach is under-constrained, essentially allowing any form of approximate reasoning to be a solution, and the third approach is over-constrained, being hard to implement in practice. This leaves us with optimal metareasoning as the most promising approach for further examination.

We adapt the view of metareasoning as a process that monitors and controls the object-level reasoning process as shown in Figure 1. This can take many different forms and can present metareasoning decisions of various complexities. We will consider an agent to be bounded rational when its metareasoning component is optimal. That is, given a particular object-level deliberation model, we look for the best possible way to control it so as to optimize the actual ground-level performance of the agent.

There are a number of key questions to answer with respect to any implementation of this framework. The answers to these questions can help establish additional desired properties of the overall system that are not directly implied by optimal metareasoning.

1. What object-level decision making architecture is employed? Is it complete? Is it sound? What tradeoffs it offers between computational resources and quality of results?
2. How does the metareasoning component model the object-level reasoning process? What kind of prior knowledge is available about the efficiency and correctness of the object-level component?
3. What run time information about the state of the object-level reasoning process is being monitored? What is known about the external environment?
4. What control decisions are being made by the meta-level reasoning process? How do these decisions affect the object-level component?
5. When and how does execution switches between the object-level and the meta-level?
6. How much time is consumed by the meta-level reasoning process? How much of the metareasoning strategy is precomputed offline? What is the online overhead?
7. Is metareasoning optimal? What assumptions are needed to establish optimality?
8. What can be said about the overall performance of the agent? Can a bound be established on how close it is to an ideal perfectly rational agent?

Over the 1980's, a number of decision making frameworks have been developed that match this form of bounded rationality. In the next section, we describe one such framework in detail and examine the answers to the above questions in that particular case. We then survey briefly a number of additional frameworks.

Optimal metareasoning with anytime algorithms

One general approach to bounded rationality is based on composition and monitoring of anytime algorithms. Methodologically, problem solving with anytime algorithms is based on dividing the overall problem into four key subproblems: elementary algorithm construction, performance measurement and prediction, composability, and meta-level control of computation.

Elementary algorithm construction covers the problem of introducing useful tradeoffs between computational resources and output quality in decision making. This fundamental problem has been studied by the AI community resulting in a variety of "anytime algorithms" (Dean and Boddy 1988) or "flexible computation" methods (Horvitz 1987) whose quality of results improves gradually as computation time increases. The same problem has been studied within the systems community in the area of "imprecise computation" (Liu *et al.* 1991). While iterative refinement techniques have been widely used in computer science, the construction of "well-behaved" anytime algorithms is not obvious. To serve as useful components of a resource bounded reasoning system, such algorithms should have certain properties: measurable objective output quality, monotonicity and consistency of quality improvement, and marginal decrease in the rate of quality improvement over time. Constructing good, reusable anytime algorithms is an important active research area. There are now many existing anytime algorithms for standard heuristic search, planning and reasoning tasks.

Performance measurement and prediction covers the problem of capturing the tradeoff offered by each system component using a "performance profile". A good performance profile is a compact probabilistic description of the behavior of the component. A typical representation is a mapping from run-time to expected output quality. Recent results show that conditioning performance profiles on input quality and other observable features of the algorithm can improve the precision of run-time quality prediction.

Composability covers the problem of building modular resource bounded reasoning system with anytime algorithms as their components. The fundamental issue is that composition destroys interruptibility–the basic property that defines anytime algorithms. A two step solution has been developed to this problem that makes a distinction between "interruptible" and "contract" algorithms (Zilberstein 1993). Contract algorithms offer a tradeoff between output quality and computation time, provided that the amount of computation time is determined prior to their activation. The idea is to first compose the best possible contract algorithm and then make it interruptible with only a small, constant penalty (Zilberstein and Russell 1996).

Finally, meta-level control covers the problem of runtime allocation of computational resources (or "deliberation scheduling" (Dean and Boddy 1988)) so as to maximize the overall performance of the system. In general, meta-level control involves modeling both the internal problem solving process and the external environment and manage computational resources accordingly. In domains characterized by high predictability of utility change over time, the monitoring problem can be solved efficiently using contract algorithms and a variety of strategies for contract adjustment. In domain characterized by rapid change and a high level of uncertainty, monitoring must be based on the use of interruptible algorithms and the marginal "value of computation" (Russell and Wefald 1991). An approach to monitoring have been developed that is sensitive to both the cost of monitoring and to how well the quality of the currently available solution can be estimated by the run-time monitor. The technique is based on modeling anytime algorithms as Markov processes and constructing an off-line monitoring policy based on a stochastic model of quality improvement (Hansen and Zilberstein 1996).

Work on anytime algorithms has produced all the key ingredients for building bounded rational agents. In order to relate this work to the model of bounded rationality we discussed in the previous section, we answer the eight key questions that are associated with that model. We use the work on anytime algorithms to illustrate the framework and refer to the simple case involving an object-level component composed of a single anytime algorithm.

1. What object-level decision making architecture is employed?
   The basic assumption about the object-level is that it is an anytime algorithm, normally an interruptible one. Some anytime algorithms, such as anytime A* (Hansen *et al.* 1997), guarantee convergence on the optimal solution, but this is not generally required.

2. How does the metareasoning component model of the object-level reasoning process? What kind of prior knowledge is available about the efficiency and correctness of the object-level component?
   Some form of a performance profile is normally used as prior knowledge. It characterize the tradeoff between runtime and quality of results. Both deterministic and probabilistic models have been developed.

3. What run time information about the state of the object-level reasoning process is being monitored? What is known about the external environment?
   Typically the two key parameters are total runtime and the quality of the current solution. In some cases, the quality of the current solution can only be estimated using certain features of the solution. In that case, the metareasoning component need to estimate the quality of the solution using the available features (Hansen and Zilberstein 2001)

4. What control decisions are being made by the meta-level reasoning process? How do these decisions affect the object-level component?
   The most simple example is when the metareasoning component only decides when to stop the anytime algorithm and return the current solution. When each activation of the metareasoning component takes non-negligible amount of time, the decision could also include the frequency of monitoring (Hansen and Zilberstein 2001).

5. When and how does execution switches between the object-level and the meta-level?
   In most cases, monitoring of anytime algorithm is done periodically at some fixed intervals, although the techniques mentioned above can also optimize the frequency of monitoring and vary it at runtime.

6. How much time is consumed by the meta-level reasoning process? How much of the metareasoning strategy is precomputed offline? What is the online overhead?
   Work on anytime algorithms often relies on pre-computed control strategies that are generated offline using the performance profile of the algorithm. Simple control strategies, such as a myopic approaches that stops the computation when the marginal value of computation becomes negative–can be computed online with little overhead. When solution quality must be estimated at runtime, this needs to be done online and could introduce a significant overhead.

7. Is metareasoning optimal? What assumptions are needed to establish optimality?
   Optimal metareasoning has been introduced for a wide range of scenarios involving anytime algorithms using certain assumptions about the performance profile. Typical assumptions are that the performance profile satisfies the diminishing returns property. A range of situations in which optimal metareasoning can be established is described in (Zilberstein 1996; Hansen and Zilberstein 2001).

8. What can be said about the overall performance of the agent? Can a bound be established on how close it is to an ideal perfectly rational agent?
   It depends on how the quality of results produced by the anytime algorithm is measured. Generally, no performance bound exist because the anytime algorithm being monitored is not subject to any constraints in terms of its efficiency or correctness. But when the quality measure provides an error bound on how close the results is to the optimal answer, we can establish a worst-case bound.

To summarize, there are many instances of optimal metareasoning involving anytime algorithms as object-level deliberation method. There are also examples of optimal metareasoning with respect to other object-level components such as algorithm portfolios (Petrik and Zilberstein 2006), and contract algorithms (Zilberstein *et al.* 2003). These examples illustrate that this well-defined model of bounded rationality can be implemented in practice in many domains.

Conclusion

We examined three different formal approaches to bounded rationality. One approach–based on optimal metareasoning–seems particularly attractive because it is both relatively easy to implement and it provides some formal guarantees about the behavior of the agent. We examined several instantiations of this approach using anytime algorithms and provided a characterization of the relationship between the object-level and metareasoning components. These examples show that metareasoning plays an important role in one of the key approaches to bounded rationality.

One interesting challenge is to try to establish mechanisms to bound the performance different between the more practical approach based on optimal metareasoning with a given object-level component and a bounded optimal agent, using the same architecture. Creating a bounded optimal agent is hard, but bounding the performance gap might be possible.

Another challenge is to develop models of bounded rationality suitable for multiple decision makers in either cooperative or competitive settings. When agents operate independently and cannot be controlled in a centralized manner, their metareasoning components need to coordinate. A simple example is when one agent decides to stop thinking and take action, but the other may see a need to continue deliberation. There has been little work so far on coordination between the metareasoning components of collaborative agents. The situation is even more complicated in competitive settings when agents need to monitor the deliberation processes of other agents, on which they have very little knowledge.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant No. IIS-0535061 and by the Air Force Office of Scientific Research under Grant No. FA9550-08-1-0181.

References

T. Dean and M. Boddy. An analysis of time-dependent planning. *Proceedings of the Seventh National Conference on Artificial Intelligence*, pp. 49–54, Minneapolis, Minnesota, 1988.

J. Doyle. Rationality and its roles in reasoning. *Proceedings of the Eighth National Conference on Artificial Intelligence*, pp. 1093–1100, Boston, Massachusetts, 1990.

G. Gigerenzer. *Adaptive Thinking: Rationality in the Real World.* Oxford University Press, 2000.

I. J. Good. Twenty-seven principles of rationality. In Godambe, V. P. & Sprott, D. A. (Eds.), Foundations of Statistical Inference, pp. 108–141. Toronto: Holt, Rinehart, Winston, 1971.

E. A. Hansen and S. Zilberstein. Monitoring the progress of anytime problem solving. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1229–1234, Portland, Oregon, 1996.

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

E. A. Hansen, S. Zilberstein, and V. A. Danilchenko. Anytime heuristic search: First results. Technical Report 97-50, Computer Science Department, University of Massachusetts, 1997.

E. J. Horvitz. Reasoning about Beliefs and Actions under Computational Resource Constraints. *Proceedings of the 1987 Workshop on Uncertainty in Artificial Intelligence*, Seattle, Washington, 1987.

E. J. Horvitz. Rational metareasoning and compilation for optimizing decisions under bounded resources *Proceedings of Computational Intelligence*, Milan, Italy, 1989.

J. W. S. Liu, K. J. Lin, W. K. Shih, A. C. Yu, J. Y. Chung and W. Zhao. Algorithms for scheduling imprecise computations. em IEEE Computer, 24:58-68, 1991.

M. Petrik and S. Zilberstein. Learning parallel portfolios of algorithms. *Annals of Mathematics and Artificial Intelligence*, 48(1-2):85-106, 2006.

S. J. Russell, D. Subramanian and R. Parr. Provably bounded optimal agents. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 338–344, Chambery, France, 1993.

S. J. Russell and E. H. Wefald. *Do the Right Thing: Studies in limited rationality*. Cambridge, Massachusetts: MIT Press, 1991.

S. J. Russell. Rationality and intelligence *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 950–957, 1995.

H. A. Simon. *Administrative Behavior*. New York: Macmillan, 1947.

H. A. Simon. *Models of bounded rationality, Volume 2*. Cambridge, Massachusetts: MIT Press, 1982.

M. P. Wellman. *Formulation of Tradeoffs in Planning Under Uncertainty*. London: Pitman, 1990.

S. Zilberstein. Operational Rationality through Compilation of Anytime Algorithms. Ph.D. dissertation, Computer Science Division, University of California at Berkeley, 1993.

S. Zilberstein. The use of anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.

S. Zilberstein and S. J. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82:181–213, 1996.

S. Zilberstein, F. Charpillet, and P. Chassaing. Optimal sequencing of contract algorithms. *Annals of Mathematics and Artificial Intelligence*, 39(1-2):1–18, 2003.