

An anytime computation approach to information gathering

Shlomo Zilberstein*
Computer Science Department
University of Massachusetts at Amherst

Abstract

Information gathering systems are operating in an ever growing environment of information sources. One of the main challenges of information gathering research is to generate a high-quality response to the information needs of the user. To achieve this goal, systems will have to trade off computational resources for quality of results. This paper shows how a special type of anytime algorithms can be used efficiently to construct and monitor information gathering systems.

1 Anytime algorithms and information gathering

The core of the problem of information gathering is how to generate a concise, high-quality response to the information needs of a user operating in an ever growing environment of information sources. In order to solve this problem, we can no longer rely strictly on information retrieval techniques, since the amount of "relevant" information to a given query may be too large for the user to handle. In order to reduce the amount of information and increase its precision, we need to perform some level of real-time problem solving and integration of information. The computational resources consumed by this operations will have to be balanced against the expected improvement in information quality. A similar view of information gathering has been advocated by Oates, Nagendra and Lesser [1994]. In travel planning, for example, a vacation destination is normally selected based on many factors

such as the cost of travel, the expected weather conditions, local events and activities, and the safety of the plan. Most of the information sources would provide only one type of information, leaving the task of integrating the pieces and matching them with the preferences of the user to the information gathering system. In addition, redundancy of information sources, communication and access costs, and limited availability of services require real-time planning and monitoring of the information gathering process.

Anytime algorithms have been used successfully to address several aspects of this planning and monitoring problem. Such algorithms offer a simple means by which a system can tradeoff quality of results for the resources used to produce those results [Boddy and Dean, 1989; Horvitz *et al.*, 1989; Zilberstein and Russell, 1993]. In previous work, we have developed an effective model for building complex systems using anytime algorithms as components [Zilberstein, 1993]. In particular, we have developed off-line compilation techniques and run-time monitoring strategies to maximize the utility of such systems. In this paper we examine the applicability of these results to the problem of information gathering and propose a particular solution.

An important observation that facilitates our modular solution to the composition problem is the distinction between contract and interruptible anytime algorithms. The original work on anytime algorithms used only algorithms whose execution can be interrupted at any time to produce less-than-optimal results. Contract algorithms offer a similar tradeoff between computation time and quality of results, but they must be informed of the total time allocation in advance. Not surprisingly, it is much easier to solve the composition problem with respect to contract algorithms and make the final product interruptible if necessary. This two stage approach to

*Author's address: University of Massachusetts, Computer Science Department, LGRC, Box 34610, Amherst, MA 01003-4610. Email: shlomo@cs.umass.edu. Phone: (413) 545-4189, Fax: 545-1249. URL: <http://anytime.cs.umass.edu/~shlomo>.

composition is one of the key aspects of our solution to the problem. Moreover, we found that in many domains it is undesirable to produce an interruptible algorithm, given the performance degradation that is involved in the transformation from a contract algorithm. Instead, we developed several monitoring strategies for contract algorithms [Zilberstein, 1993].

Information gathering is, in particular, a domain where contract algorithms can be used efficiently. Prior knowledge about the characteristics of different information sources and the time-dependent utility of the user can be used to optimally control the information gathering process. Most basic search algorithms to collect and integrate information can be implemented as anytime algorithms. The main problem addressed in this paper is how to monitor the execution of a complex information gathering task and respond to unexpected situations.

2 The components of information gathering tasks

A typical information gathering task is translated into a complex set of interrelated information retrieval and information integration operations. Such operations define alternative information sources (e.g. different airline reservation systems), components of information that must be collected independently (e.g. weather report and local events for potential vacation destinations) and domain-dependent operators that integrate pieces of information into concise responses (e.g. a procedure that determines how satisfactory is a potential vacation destination). Many of these information retrieval and information integration operators can be implemented as contract algorithms. The resulting task structures present a complex meta-level control problem related to the management of uncertainty and the execution of the components.

Existing monitoring strategies for anytime algorithms address three important aspects of this meta-level control problem:

1. The representation of expectations regarding information quality from various sources and expectations regarding the quality of the results produced by each integration operator.
2. The combination of those expectations and the

allocation of computational resources to the primitive operators so as to maximize the utility of the information gathering system; and

3. The modification of expectations based on partial task execution and the response to abnormal operation.

The next two sections describe briefly the meta-level information structures and the monitoring approach that we propose.

3 Representing expectations with CPPs

The performance of elementary anytime algorithms can be characterized statistically using *performance profiles* [Boddy and Dean, 1989]. A performance profile describes how the quality of the output improves as a function of execution time. We have developed an off-line compilation process that constructs the performance profile of a composite system based on the performance profiles of its components. In order to solve the problem, we introduced the notion of *Conditional Performance Profile* (CPP) that captures the dependency of the output quality on the quality of the input. CPPs offer a useful mechanism to represent probabilistic expectations with respect to information gathering operations. For example, the quality of a vacation plan will depend on the precision of the weather report and the completeness of the cost projection. These dependencies can be described by the CPP of the planning algorithm. Through our compilation mechanism we can generate similar expectations for complex tasks that involve many interdependent operators.

We have constructed CPPs for sensing and planning algorithms and used them successfully in a path-planning application [Zilberstein and Russell, 1993]. Unfortunately, in standard knowledge-based systems it is hard to characterize the performance improvement as a function of time. However, recent work has shown that by structuring the available knowledge in a hierarchical way and by limiting the amount of data and knowledge used at each level, we can construct knowledge-based systems that have all the characteristics of “well-behaved” anytime algorithms [Mouaddib and Zilberstein, 1995]. Another

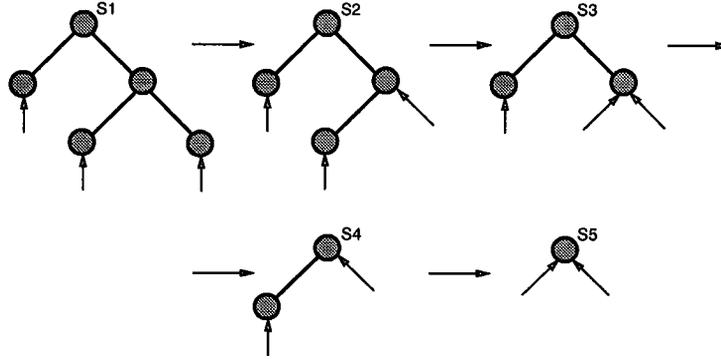


Figure 1: A sequence of residual sub-tasks

difficulty with information gathering systems is the limited applicability of *static* CPPs since information may be constantly added or deleted, the validity of data may expire, and computation load may limit the availability of information. The effect of these factors can be statistically summarized by dynamically adjusting CPPs. Each information source can maintain and “advertise” its current CPP with respect to different search operations. Reward and payment for services could be calculated accordingly (although this aspect is not discussed here).

4 The meta-level control problem

A utility maximizing approach to information gathering will normally generate responses that are based on partial and incomplete information because of the high cost of obtaining precise information. Given the CPPs of the information sources and of the complete information gathering task, we seek a meta-level monitoring technique that can maximize the user’s overall utility.

The first step in monitoring contract algorithms involves the calculation of the initial contract time. The expected quality of a contract algorithm \mathcal{A} at time t can be represented by:

$$U'_{\mathcal{A}}(S_t, t) = \sum_i Q_{\mathcal{A}}(q, t)[q_i] U_{\mathcal{A}}(S_t, t, q_i) \quad (1)$$

Where $Q_{\mathcal{A}}(q, t)$ is the CPP, $Q_{\mathcal{A}}(q, t)[q_i]$ denotes the probability of output quality q_i , and $U_{\mathcal{A}}(S, t, q)$ represents the utility (from the user’s perspective) of a

result of quality q in state S at time t . Hence, an initial contract time, t_c , can be determined before the task is activated by solving the following equation:

$$t_c = \arg \max_t \{U'_{\mathcal{A}}(S_t, t)\} \quad (2)$$

Once an initial contract time is determined, several monitoring policies can be applied. The most trivial one is the *fixed contract* strategy that leads to a passive monitoring scheme. One improvement of this monitoring strategy involves reallocation of residual time among the remaining contract algorithms. Given a complex information gathering task, the execution of any elementary operator can be viewed as a transformation of a node in the graph representing the task from a computational node to an information node of a certain quality. This transformation is shown in Figure 1. The *actual* quality of this information is only known after the corresponding operation is performed. Based on the actual quality of information, the remaining time (with respect to the global contract) can be reallocated among the remaining computational components to yield significant performance improvement. In order to be able to allocate time optimally to each operator, the monitor needs not only the CPP of the complete task, but also the CPPs of all the *residual sub-tasks*. Hence, the off-line compilation problem has to be solved for each residual sub-task (whose number is linear in the size of the overall task).

The second possible improvement over the fixed contract strategy involves adjustments to the original contract time. As before, once an elementary operator is executed, the monitor can consider its output as an input to a smaller residual task composed

of the remaining anytime algorithms. By solving the previous equation (that determines the contract time) for the new residual task, a better contract time can be determined that takes into account the actual quality of the intermediate results generated so far.

5 Summary

This paper examines the applicability of recent results in the area of anytime computation to information gathering tasks. We believe that several key problems in information gathering can be effectively solved by existing techniques for meta-level control of anytime algorithms. In particular, we propose to use contract algorithms for implementing the basic information retrieval and information integration operators and characterize their performance using dynamic CPPs. Then, existing off-line compilation and a run-time monitoring techniques can be used to construct a utility maximizing model of information gathering. Early experience with these techniques (using static CPPs) in other domains has produced successful systems. In information gathering tasks however there is a need to study the construction of dynamic performance profiles to reflect constant changes in information quality and availability. We are currently studying new mechanisms for real-time adjustment of performance profiles and their effect on the design of the monitoring component.

References

- [Mouaddib and Zilberstein, 1995] A. I. Mouaddib and S. Zilberstein. Knowledge-Based Anytime Computation. In preparation, 1995.
- [Boddy and Dean, 1989] M. Boddy and T. Dean. Solving time-dependent planning problems. Technical Report CS-89-03, Department of Computer Science, Brown University, Providence, 1989.
- [Dean and Boddy, 1988] T. Dean and M. Boddy. An Analysis of Time-Dependent Planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 49–54, Minneapolis, Minnesota, 1988.
- [Horvitz *et al.*, 1989] E. J. Horvitz, G. F. Cooper and D. E. Heckerman. Reflection and action under scarce resources: Theoretical principles and empirical study. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 1121–1127, Detroit, Michigan, 1989.
- [Oates *et al.*, 1994] T. Oates, M. V. Nagendra Prasad and V. R. Lesser. Cooperative Information Gathering: A Distributed Problem Solving Approach. Technical Report 94-66, Computer Science Department, University of Massachusetts at Amherst, 1994.
- [Smith and Liu, 1989] K. P. Smith and J. W. S. Liu. Monotonically improving approximate answers to relational algebra queries. *COMPSAC '89*, Orlando, Florida, 1989.
- [Zilberstein, 1993] S. Zilberstein. Operational Rationality through Compilation of Anytime Algorithms. Ph.D. dissertation, Computer Science Division, University of California at Berkeley, 1993.
- [Zilberstein, 1995] S. Zilberstein. On the Utility of Planning. In M. Pollack (Ed.), *SIGART Bulletin Special Issue on Evaluating Plans, Planners, and Planning Systems*, Vol. 6, No. 1, 1995.
- [Zilberstein and Russell, 1993] S. Zilberstein and S. J. Russell. Anytime Sensing, Planning and Action: A Practical Model for Robot Control. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1402–1407, Chambéry, France, 1993.
- [Zilberstein and Russell, 1995] S. Zilberstein and S. J. Russell. Optimal Composition of Real-Time Systems. *Artificial Intelligence*, forthcoming, 1995.