Learning to Integrate Reactivity and Deliberation in Uncertain Planning and Scheduling Problems

Steve A. Chien
Jet Propulsion Lab, M/S 525-3660
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
chien@aig.jpl.nasa.gov

Melinda T. Gervasio
Beckman Institute
University of Illinois
405 North Mathews Avenue
Urbana, IL 61801
gervasio@cs.uiuc.edu

Gerald F. DeJong
Beckman Institute
University of Illinois
405 North Mathews Avenue
Urbana, IL 61801
gervasio@cs.uiuc.edu

Abstract

This paper describes an approach to planning and scheduling in uncertain domains. In this approach, a system divides a task on a goal by goal basis into reactive and deliberative components. Initially, a task is handled entirely reactively. When failures occur, the system changes the reactive/deliberative goal division by moving goals into the deliberative component. Because our approach attempts to minimize the number of deliberative goals, we call our approach Minimal Deliberation (MD). Because MD allows goals to be treated reactively, it gains some of the advantages of reactive systems: computational efficiency, the ability to deal with noise and non-deterministic effects, and the ability to take advantage of unforseen opportunities. However, because MD can fall back upon deliberation, it can also provide some of the guarantees of classical planning, such as the ability to deal with complex goal interactions. This paper describes the Minimal Deliberation approach to integrating reactivity and deliberation and describes an ongoing application of the approach to an uncertain planning and scheduling domain.

INTRODUCTION

The AI problem of automatically achieving goals has been redefined in the last few years. The classical planning problem can be broadly characterized as finding a set of operators together with sufficient constraints such that when applied to some initial state the resulting state provably satisfies some goal relation. However, this is a narrow view of what is now seen as a more general problem. Recently, there has been a great deal of interest in reactivity as a model of action [Suchman87]. While the classical view of planning has been shown to have computational problems [Chapman87]; from a different perspective one might instead blame our failure to conceive of alternative frameworks for modeling world changes and formalisms for action selection.

Reactivity takes a different, more efficient view of action selection. Pure reactivity fundamentally gives up the idea of projecting the results of actions. Instead an agent reacts to the current state of affairs in the world as directly perceived by sensors. In a sense, reactivity is a hill-climbing action-selection model. The evidence taken into account in the selection of an action is necessarily local (i.e., the current readings of sensors). Based on this purely local information an action is taken that may have resounding global ramifications, fooling the agent into climbing to the top of a locally steep foothill from which state the goal is unachievable.

This phenomenon often occurs in the form of interacting sub-goals both in planning and scheduling. In a planning context, as you exit the parking lot on your way home from work you may prefer a right turn (it more directly leads toward your house, it is less expensive than a left turn across traffic, etc.). However, in the context of a second goal of picking up a loaf of bread, it may be better to turn left, taking you past a supermarket on the way. In a scheduling context, interactions occur through resource contention. A job may finish earlier if allowed to execute one of its subtasks at a certain time, but the overall schedule may suffer. Approaches that address managing such problems of purely reactive systems include: developing a theory of benign environments in which a reactive agent may be more certain that its reactive inclination will meet with success [Agre88, Hammond90]; the integration of classical planning with reactivity [Drummond90, Kaelbling86, Turney89]; and application of machine learning to this end [Gervasio90, Mitchell90, Laird90]. These approaches begin with what is essentially a classical planner and, guided by experience, result in the formulation of reactive components as well.

This research approaches planning and scheduling from a different point of view. Instead of learning to incorporate reactivity into a classical deliberative framework, we propose incorporating minimal classical deliberation into an initially purely reactive system. As failures are encountered, the system utilizes its world model to explain why the desired state of affairs was not brought about by the executed actions.

In the case of a failure of a reactive goal, the failure could be due to a faulty set of reactions or due to uncertainty in the effects of actions or schedules. In the case of failure of a deliberative goals, the failure must be due to interference from a reactive goal. In the case of uncertain effects causing the failure of a reactive goal, deliberation can be used to attempt to improve the plan. In the case of reactive interference in a reactive or deliberative goal, the offending reactions are inhibited by moving the associated goal into the deliberative component, where the negative goal interaction will be considered and avoided.

In this way the purely reactive system adopts just enough deliberation to avoid goal interaction pitfalls. Since deliberation occurs only in reaction to observed failures, (i.e. the resultant plan remains uncommitted on those goals not appearing in the failure trace) this approach will generally retain some level of flexibility by avoiding a rigid classical plan or schedule for all of the goals. Thid flexibility allows the MD approach will retain some of the benefits of reactivity: tolerance of noise, uncertainty, and incomplete knowledge as well as computational efficiency. Yet the MD approach also benefits from its ability to fall back upon traditional deliberative planning. It gains the ability to solve problems which require simultaneous consideration of multiple interacting goals. Additionally, through explanation-based learning(EBL), it gains the ability to cache and generalise decisions made in the plan construction process. As with traditional EBL, the learned deliberation molecules allow a system to find plans more quickly. But more importantly, these deliberation molecules allow a system to avoid repeating the failures resulting from the short-sighted decision of the reactive component.

These benefits of coordinating reactivity and deliberation are relevant to both planning and scheduling issues described in this paper. Reactivity can take advantage of unforseen opportunities. In, planning this is the ability to take advantage of fortuitous conditions in the world state. In scheduling, this is the ability to take advantage of unforseen resource availability. Another strength of reactivity is the capability to deal with uncertainty and noise. In planning this means the ability to deal with uncertain action effects and/or world state. In scheduling this means the ability to deal with uncertain resource consumptions and availabilities. A third strength of reactivity is its computational efficiency due to avoidance of explicit projection. In planning, this means not having to explicitly determine future world states. In scheduling, this means not having to explicitly determine future resource utilisation. The principal strength of deliberation is the ability to deal with arbitrary goal interactions by searching the space of possible plans and/or schedules. In planning this means being able to deal with complex precondition and effect interactions between goals. In scheduling, this means being able to deal with difficult resource interactions.

There are a number of assumptions underlying the MD approach. First, we assume that the cost of failures is sufficiently low so that the cost of failures incurred while acting reactively is outweighed by the overall gains in flexibility and efficiency from reactivity. A corollary to this assumption is that the reactive component is sufficiently competent to solve the majority of the goals. Without this constraint, the MD approach would incur the cost of numerous failures only to end up doing primarily deliberative planning. Second, we assume the presence of domain models to allow the system to fall back upon classical planning as well as permitting use of EBL. Third, the system must be allowed multiple attempts to solve a problem.

THE MD ARCHITECTURE

The system architecture advocated by the MD approach is that of an interacting set of components: a deliberative element, a reactive element, and a learning element. The deliberative element is a conventional planner which constructs classical plan/schedule molecules for goal conjuncts requiring deliberation. By analysing the precondition and schedule interactions and performing extensive search deliberation can resolve the goal interactions. The learning element uses EBL [DeJong86, Mitchell86] to learn general plan/schedule molecules which indicate how to achieve a set of goals by designating a reactive/deliberative goal allocation and a set of actions for the deliberative goals.

The reactive element proposes actions using a shallow decision model of reaction rules. Each reactive rule specifies a set of state conditions and resource requirements which specify an action as appropriate to execute. Multiple actions may be executed during a single timestep if resources allow. In most cases, failures in the reactive component will be due to goal interactions. Reaction rules consist of interrupt rules, which cause actions to be executed regardless of the other actions the agent is taking (i.e. actions determined by the deliberative component), and suggestion rules, which are executed when the system has no current pending actions. Thus, interrupt rules represent actions to take advantage of immediate opportunities or avoid dangerous situations regardless of the current deliberative plan, while suggestion rules direct activity when the system is confronted by a set of goals, and does not have a current plan.

Every reaction rule is defined with respect to a goal, and can only apply when its goal matches a reactive goal of the system. Thus, a reaction can be overridden by the deliberative component by removing the triggering goal from the set of reactive goals and planning for the goal deliberatively. Thus, in our architecture, there are three levels of priority: interrupt rules, the action advocated by the current plan, and suggestion rules. Within a given priority level, if more than one action is applicable, the system chooses one arbitrarily but deterministically (e.g. the same set of goals and state will produce the same action). For example, in a delivery domain, interrupt rules might trigger when the truck is at the location of one of its deliveries. This can occur in the midst of executing a decision molecule constructed by the deliberative component, and it results in actions other than those in the decision molecule. An example suggestion rule would be one which causes the truck to move towards the closest delivery site if it does not have a decision molecule to guide it otherwise.

THE MD APPROACH

In the MD approach, a system originally acts based upon a shallow, simple decision model. Through experience, the system gradually acquires a set of decision molecules which allow it to plan past local maxima encountered by the shallow decision model. Because of this progression, we describe the MD approach as "becoming decreasingly reactive", as the proportion of goals the system solves by deliberation increases (where we also consider as deliberative the compiled decision molecules created by the deliberative element). Eventually, for a fixed distribution of problems, the system will learn a set of decision molecules sufficient to allow it to solve the problems occurring in the distribution. Furthermore, because the MD approach uses EBL, the system also learns to avoid a general class of failures relevant to a particular plan, thus reducing the number of failures required to learn a satisfactory set of plans.

A problem consists of a conjunction of goals, and the task of a system in the MD approach is to divide the goals into a deliberative set and a reactive set such that the goals are all achieved with the minimum amount of deliberation and maximum amount of flexibility possible. A plan to solve a conjunction of goals is thus a composite plan/schedule which consists of a decision molecule, constructed by the deliberative component to solve the set of deliberative goals, and a set of reactive goals to be achieved by the reactive component. The MD algorithm is shown below:

```
Given a problem consisting of:

G - a set of problem goals
I - the initial state

REAC := G
DELIB := {}

loop
PLAN := Classical_Planner(DELIB,I)
Execute(PLAN,REAC)
if all goals achieved return SUCCESS
else if REAC = {} return FAIL
else
    for each goal in REAC
        if <goal not achieved> OR
           <reactive action in pursuit of goal
            interfered with another goal G'>
        then
           REAC := REAC - goal
           DELIB := DELIB + goal
go loop

if SUCCESS then generalize successful plan
```

The key to the MD approach is the blame assignment process. In general, failures are due to interactions between subgoals, as the reactive methods are intended to be sufficient to achieve goals without interference. Interference can occur at the planning level (due to an action in service of one goal clobbering a protection in service of another goal) and at the scheduling level (resource expenditures due to one goal causing a resource failure for another goal).

Blame assignment consists of determining which goals are involved and then using this information to reduce future failures due to goal interactions. In goal identification process. there are planning failures and scheduling failures. Each of these failure types (planning, scheduling) can cause a goal to be identified as relevant to a goal analysis. In the first way, a goal G fails, likely due to actions in service of another goal. This goal is called a conflictee and is considered in the analysis described below. This set of circumstances can be detected by checking if goals are achieved at the end of execution (infinite looping is detected by an execution limit). The second relevant goal type is a conflicter goal. A goal G is deemed a conflicter if an action A in service of G caused a failure of another goal H. In the context of planning, this occurs if the conflictee H is a deliberative goal and A clobbers a protection in the plan to achieve H. In a scheduling context a goal G is deemed a conflicter if an action A in service of G was the largest consumer/user of a resource R which caused a scheduling failure for a deliberative goal H.

We now describe how this determination of goal interference is used to modify the allocation of reactive and deliberative goals. If a reactive goal G1 fails without interference, it is moved to the deliberative component and thusly will be achieved by the classical planner and scheduler. A deliberative goal G1 cannot fail without interference as the planner performs full projection. In the case of a goal failing due to interference from a second goal G2, there are four cases, G1 and G2 reactive, G1 reactive and G2 deliberative, G1 deliberative and G2 reactive, and G1 and G2 both deliberative. How each of these cases is treated is described below.

1. Because the deliberative element performs full projection, two deliberative goals cannot interfere, thus the failure case of both G1 and G2 deliberative cannot occur.
2. If G1 is a reactive goal, and G2 deliberative, the MD approach will move G1 to the deliberative goal set and the classical planner will ensure that the negative goal interaction between G1 and G2 will be avoided.
3. If G1 is deliberative and G2 reactive, then due to the blame assignment scheme G2 will be moved into the deliberative component. In the next cycle both G1 and G2 will be delegated to the deliberative component and the interaction will be considered and avoided.
4. If G1 is a reactive goal and it has been thwarted by another reactive goal G2, the blame assignment scheme will move G1 to the deliberative component. If in the next cycle G2 still interferes, it is an example of case 3 above and will be treated accordingly.

Thus the process of moving more goals to the deliberative component continues until the system converges upon a set of deliberative goals for which the planner and scheduler constructs a plan and schedule which in combination with the reactive element achieves all of the problem goals.

This classical plan is then generalised using EBL, with the reactive goals being generalised to a default level. This resultant plan structure (and reactive/deliberative division) can then be used to solve future problems as follows. When problems are initially posed to MD, it begins by attempting to match the goals and initial conditions to an existing decision molecule. If a matching decision molecule exists, it is used in an attempt to solve the plan. If all such matching molecules fail, the system attacks the problem entirely reactively and the entire MD approach is called from scratch.

EVALUATION

The MD approach has been implemented for a simple delivery planning domain [Chien91]. We have extended the failure analysis algorithm and are currently implementing this newer version of MD for a more complex mathematical planning and scheduling domain. This ongoing implementation is the one described in this paper. In this mathematical domain, each goal can be achieved by the execution of a number of actions. Each action has a randomized number of resource requirements, and possibly state requirement preconditions for each of the resources (e.g., a value for a predicate on the resource). Planning goal conflicts occur through incompatible resource state requirements. Scheduling resource contention occurs through goals competing for resources. Uncertainty exists through a random element in duration of primitives (and thus resource usage).

We plan to test our architecture by generating domain theories which vary a number of parameters which will affect the overall scheduling and planning goal interaction rate. The domain parameters are: 1) the # of resource types (affects resource and conflict rate); 2) average number of resources each action uses (affects resource conflict rate); 3) frequency and types of resource conditions (affects planning conflict rate); and 4) # of preconditions per primitive (affects planning conflict rate). Finally, we plan to vary the amount of action duration uncertainty, which affects the amount of benefit gained by deferring decision-making.

In order to compare with the MD approach, we are currently implementing a fully deliberative planner and scheduler. This comparison classical system simply delegates all of the goals to the deliberative component.

The metrics which we plan to use to evaluate the plans produced by the two systems are: 1) total CPU time required for decision-making; 2) robustness of the schedule (% of goals achieved by deadlines); 3) average time to completion of individual goals; and 4) average time to completion of all goals. These metrics will be evaluated for different combinations of the domain parameters described above.

DISCUSSION

This research is preliminary, and there are a number of outstanding research issues. One difficult issue is determining the correct level of generalization for the reactive portion of any plan/schedule. Because reactive actions are undetermined, analysing generality of the goal achievement methods is difficult. While committing the planner to the same general set of actions used by the reactive component in the current problem would allow EBL on the action trace, it commits the planner to the same general set of actions - losing the flexibility allowed by reactivity and forcing a possibly expensive causal analysis of the example. Yet another approach would be to generalise the reactive portion aggressively and allow later learning to either reduce the level of generality or learn more specific plans which would shadow the over-general plan in cases where it was inappropriate.

One view of the MD approach is that of using deliberation to learn patches to a set of reactive rules. In this view our techniques allow for encoding of a quick and dirty set of reactive rules which solve the majority of problems. Through learning, a set of patches can then be constructed to allow these imperfect rules to solve a given distribution of problems.

Another interesting issue for examination is the tradeoff between reactivity and deliberation in the purely reactive component. Currently, the reactive component does no projection before interrupting the current plan and the deliberative element performs full projection. While ideally both approaches components would be less extreme, the same general mechanisms for integrating deliberation and reactivity would apply.

Another possible approach to integrating deliberation and reactivity is to use the same failure-driven method for splitting goals between the reactive and deliberative component to learn control rules specifying allocation of goals to the deliberative and reactive components. While we feel that the current MD macro-based approach better preserves the notion of a plan/schedule context in that the deliberative actions selected may impact the success of the reactive component, this is a larger issue involving the operationality/generality tradeoff.

Another issue is that of controlling moving interacting goals into the deliberative component. Managing the tradeoff between more expensive (and likely more accurate) failure analyses and more heuristic (and likely less accurate) goal analyses is an issue for future work.

RELATED WORK

Drummond and Kaelbling [Drummond90, Kaelbling86] describe anytime approaches wherein planning is used to constrain the reactions, which are always available for deciding on actions. [Turney89] interleaves planning and execution by allocating some predetermined amount of time to each phase in turn, while [Hanks90] uses the constraints of urgency and insufficient information to determine when to pass control to the reactive component. In these approaches, any goal may thus be addressed reactively or deliberatively. In contrast, a system in the MD approach initially addresses all its goals reactively but incrementally learns which goals require deliberation to avoid negative interactions and which goals can be addressed reactively without preventing the achievement of other goals. Thus, the MD approach can guarantee the achievement of its goals, which the others in general cannot.

Guaranteed goal achievement is similar to ideas presented in [Gervasio90, Martin90]. In [Gervasio90], the a priori (deliberative) planner must construct an achievability proof for each deferred goal, while in [Martin90], the strategic (deliberative) planner assigns the reactive planner those goals which the reactive planner has proven itself capable of handling. In contrast, in the MD approach, each goal is considered achievable during execution until experience shows otherwise. The MD need not prove achievability but instead incurs failures to determine which goals must be deliberated upon.

In [Mitchell90, Laird90] systems become increasingly reactive by compiling deliberative decisions into stimulus-response rules/chunks. As the decision molecules learned by MD are compiled schemata, MD becomes increasingly reactive in the same sense. However, it becomes decreasingly reactive in the sense that it initially addresses all goals reactively, but gradually learns to address particular goals deliberatively. In contrast, since Theo-Agent and SOAR derive all their rules/chunks from deliberative plans, they always address their goals purely deliberatively.

TRUCKER [Hammond88] learns to optimise its planning from successful opportunistic problem-solving. While in the MD approach, a system learns which goals interact negatively and modifies its planning behavior to deliberate over these goals and avoid the interaction, TRUCKER learns which goals interact positively and modifies its planning behavior to take advantage of this interaction. Other work on learning from failure deals with purely deliberative plans, in contrast to the composite plans in the MD approach.

CONCLUSION

This paper has presented an approach to integrating reactivity and deliberation in planning and scheduling in uncertain domains. In this approach, called Minimum Deliberation (MD), the problem-solver initially attempts to solve all goals reactively. When the system encounters failures it responds by moving reactive goals into the deliberative component. By performing this refinement, the system extends its analysis of the problem minimally until the reactive component can solve the remainder of the goals. Resultant successful plans are then generalised using a combination of EBL and default generalisation information. By introducing deliberation minimally, the MD approach retains some of the benefits of reduced computation and flexibility from reactivity while still being able to fall back upon deliberation to solve complex interactions.

Acknowledgements Portions of this research were performed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Other portions of this work were performed at the Beckman Institute of the University of Illinois and were supported by the Office of Naval Research under Grants N-00014-86-K-0309 and N-00014-91-J-1563.

References

P. E. Agre, "The Dynamic Structure of Everyday Life," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, Oct 1988.

D. Chapman, "Planning for Conjunctive Goals", *Artificial Intelligence 32*, 3 (1987).

S. A. Chien, M. T. Gervasio, and G. F. DeJong, "On Becoming Decreasingly Reactive: Learning to Deliberate Minimally," *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, Jun 1991.

G. F. DeJong and R. J. Mooney, "Explanation-Based Learning: An Alternative View," *Machine Learning 1*, 2 (Apr 1986).

M. Drummond and J. Bresina, "Anytime Synthetic Projection: Maximising the Probability of Goal Satisfaction," *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, Aug 1990.

M. T. Gervasio, "Learning General Completable Reactive Plans," *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, Aug 1990.

K. Hammond, T. Converse and M. Marks, "Learning from Opportunities: Storing and Re-using Execution-Time Optimisations," *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, MN, Aug 1988.

K. Hammond, T. Converse and C. Martin, "Integrating Planning and Acting in a Case-Based Framework," *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, Aug 1990.

S. Hanks and R. J. Firby, "Issues and Architectures for Planning and Execution," *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, CA, Nov 1990.

L. P. Kaelbling, "An Architecture for Intelligent Reactive Systems," *Proceedings of the 1986 Workshop on Reasoning About Actions & Plans, Timberline*, OR, Jun 1986.

J. Laird and P. Rosenbloom, "Integrating Execution, Planning, and Learning for External Environments," *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, Aug 1990.

N. G. Martin and J. F. Allen, "Combining Reactive and Strategic Planning through Decomposition Abstraction," *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, CA, Nov 1990.

T. M. Mitchell, R. Keller and S. Kedar-Cabelli, "Explanation-Based Generalisation: A Unifying View," *Machine Learning 1*, 1 (Jan 1986).

T. M. Mitchell, "Becoming Increasingly Reactive," *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, Aug 1990.

L. A. Suchman, Plans and Situated Actions, Cambridge University Press, Cambridge, 1987.

J. Turney and A. Segre, "SEPIA: An Experiment in Integrated Planning and Improvisation," *Proceedings of The AAAI Spring Symposium on Planning and Search*, Mar 1989.