Replanning: A New Perspective

William Cushing and **Subbarao Kambhampati**
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
{wcushing|rao}@asu.edu

Abstract

Replanning is generating a new plan in response to execution failures. Past work has procedurally characterized replanning as a special case of Plan Reuse: reuse the prior plan to solve the new problem. Just as in normal reuse, this type of "replanning" system focuses on minimally perturbing the prior plan while accounting for the alterations in current state and goal description. There are two immediately apparent limitations to this view of replanning. First, an adequate expression of plan failure is rarely as simple as altering the current state and goal descriptions. Second, plan reuse is motivated by efficiency considerations during plan generation, yet the ideal result from replanning is clearly to minimize the total residual *execution* cost.

In this paper, we argue that replanning should rightly be seen as a natural approximation to full multi-agent game-theoretic planning. Specifically, a natural step in modelling a complex domain with many external agents is to ignore them–by assuming default action. While this approximation allows computationally feasible approaches, execution failures will arise when external agents act unpredictably.

Replanning is precisely the problem of resolving such execution failures *without* "giving up" on the simple model.

Introduction

Replanning techniques resolve execution failures of prior plans. The literature has conflated replanning and plan reuse by viewing replanning as a special case of plan reuse: reuse the prior plan on altered current and goal states. This line of research presents *Minimal Perturbation Planning* as the panacea for both plan reuse and replanning(Kambhampati 1990; Hammond 1986; Simmons 1988). To start with, it is overly optimistic to assume that an execution failure can be represented by a new planning problem with altered current and goal states. This amounts to assuming that execution failures are independent of the agent's behavior. When the agent is caught in some kind of *trap*, this assumption leads to indefinitely repeating failure. It is natural to require that *correct* replanning behavior does not fall into any known traps.

Furthermore, preserving plan structure has little connection to resolving failures. To the contrary, execution failures imply that at least some part of the plan needs to be altered. Trying to preserve structure is, therefore, a handicap. Nonetheless, there is a connection. We view minimal perturbation planning as a heuristic approach to keeping overall execution costs low when replanning in the presence of collaborators. In the presence of collaborators, altering one's intentions arbitrarily can degrade overall execution performance: external agents could have based their own plans off of one's stated intentions. Such a dependency is a *commitment*, and thus we have that the *quality* of a replanning solution (overall execution performance) requires respecting commitments.

A Motivating Scenario: We take the perspective that replanning is the projection of a multi-agent planning and execution problem onto a single-agent. Our motivating example is $\mathrm{AltAlt}^{\mathrm{PDA}}$, a (hypothetical) personal digital assistant equipped with an automated planner. Its owner, Romeo, uses $\mathrm{AltAlt}^{\mathrm{PDA}}$ to (among other things) automatically produce efficient travel plans. Over time, we notice that not every travel plan executes in its entirety: some execution failure forces Romeo to take over the planning. Of course, Romeo is more than capable of dealing with such situations, but in order to enhance the value of the tool we will consider equipping $\mathrm{AltAlt}^{\mathrm{PDA}}$ with a replanning capability in order to support the dynamic execution of these travel plans by Romeo.

The full problem Romeo faces is very complex, as his travels take him to distant and unfamiliar places. This involves interacting with many other agents, directly and indirectly. Formalizing this situation as a full multi-agent problem would allow $\mathrm{AltAlt}^{\mathrm{PDA}}$ to (theoretically) produce optimal plans for Romeo – including branches for every conceivable contingency. Of course, solving such a problem automatically is far from feasible. Nonetheless, Romeo, or any other human, does not find travel planning unduly difficult. The secret here is project away all the other agents by assuming that they will do what they are "supposed" to do: thereby reducing an $\mathrm{AND}^{n-1} \times \mathrm{OR}$ search space (n agents) into a much simpler OR search space.

Contribution: In this paper, we consider how to ground the semantics of replanning in terms of the semantics of planning problems. Our approach is to encode trap avoidance and commitment respect as additional constraints by systematically altering the original problem description. Plausible failure scenarios tend to involve very complicated issues such as soft constraints, time, and resources. Our examples will thus be with respect to metric-temporal partial-satisfaction planning, although our approach to replanning is essentially orthogonal to the type of planning. We begin by exploring our motivating example to elucidate our perspective on what constitutes appropriate situations for replanning. After clarifying our specific assumptions about the kinds of domains where replanning would be successfully applied we concentrate on specific syntactic forms of traps and commitments that could easily be produced were there a mechanism for respecting such constraints. With respect to this subset of traps and commitments we demonstrate our systematic approach for altering the original planning problem.

Replanning is not without history in the literature, and our presentation differs radically at the surface level. Before concluding, we discuss how our perspective on replanning unifies the procedural definitions implicit in PRIAR(Kambhampati 1990), GORDIUS(Simmons 1988), CHEF(Hammond 1986), and SHERPA(Koenig, Likhachev, & Furcy 2004).

References

Hammond, K. J. 1986. Chef: A model of case-based planning. In *AAAI*, 267–271.

Kambhampati, S. 1990. Mapping and retrieval during plan reuse: A validation structure based approach. In *AAAI*, 170–175.

Koenig, S.; Likhachev, M.; and Furcy, D. 2004. Lifelong planning a. *Artif. Intell.* 155(1-2):93–146.

Myers, K. L. 1999. Cpef: A continuous planning and execution framework. *AI Magazine* 20(4):63–69.

Nebel, B., and Koehler, J. 1995. Plan reuse versus plan generation: A theoretical and empirical analysis. *Artif. Intell.* 76(1-2):427–454.

Pell, B.; Gat, E.; Keesing, R.; Muscettola, N.; and Smith, B. 1997. Robust periodic planning and execution for autonomous spacecraft. In *IJCAI*, 1234–1239.

Simmons, R. G. 1988. A theory of debugging plans and interpretations. In *AAAI*, 94–99.

Stentz, A. 1995. The focussed d* algorithm for real-time replanning. In *IJCAI*, 1652–1659.

van den Briel, M.; Nigenda, R. S.; Do, M. B.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) planning. In *AAAI*, 562–569.

van der Krogt, R., and de Weerdt, M. 2005. Plan repair as an extension of planning. In *ICAPS*.