

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 involves generating a new plan to fix execution failures. Past work has often characterized replanning procedurally as a special case of Plan Reuse: reuse the current plan to solve a new problem with altered initial and goal states. Just as in normal reuse, these replanning systems focus on minimally perturbing the plan while accounting for the alterations in the problem. There are two important limitations to this view of replanning. First, many failures cannot be represented as alterations to initial and goal states. Second, plan reuse is motivated by efficiency considerations during plan generation, while replanning attempts to minimize the residual *execution* costs of the partial execution.

In this paper, we argue that replanning should rightly be seen as solving a new planning problem that not only captures *general execution failures*, but also the *commitments* incurred during the partial execution. Failures are modeled as fine-grained modifications to operator descriptions, and commitments are modeled as soft constraints. From this general perspective, minimal perturbation planning can be understood as a crude heuristic for respecting commitments: if we keep the plan the same, then we are likely to respect the commitments as well.

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 current plan on altered initial 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 initial and goal states. This boils down to assuming that execution failures are independent of the agent's behavior. When this is not the case, making the assumption can lead to indefinitely repeating the failure. To avoid this, we say that a *correct* replanning solution avoids repeating execution failures.

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. Yet, there is a connection: minimal perturbation plan-

ning can be seen 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 can degrade overall execution performance: external agents could have based their own plans off of one's stated intentions. We refer to such a dependency as a *commitment*, and say that the *quality* of a replanning solution is in terms of respecting commitments.

A Motivating Scenario: In this paper, 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 $\text{AltAlt}^{\text{PDA}}$, a hypothetical personal digital assistant equipped with an automated planner. Its equally hypothetical user, Romeo, uses $\text{AltAlt}^{\text{PDA}}$ to automatically produce efficient travel plans. We will consider equipping $\text{AltAlt}^{\text{PDA}}$ 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 $\text{AltAlt}^{\text{PDA}}$ to produce globally optimal plans. Of course, Romeo's colleagues may not be very impressed with the endless negotiations carried out on Romeo's behalf. We presume that even though there are other agents, it is unnecessary to take them into consideration in order to synthesize efficient travel plans. However, once a plan has been published, it is no longer possible to completely ignore the external agents: the commitments induced by the published plan must be considered. Therefore, we project the multi-agent problem onto a single-agent problem by ignoring everything except the static impact of the external agents: the projection abstracts every aspect of the external agents except the current set of commitments.

Contribution: In this paper, we consider how to update the original planning problem so that solutions resolve failures while respecting commitments. Doing so involves two distinct modeling issues: a language for failure representation, and a language for commitment representation. We present a particular language for representing failures, and define the correctness of a replanning solution with respect to it. From there, we consider the somewhat orthogonal is-

sue of representing commitments. We model a commitment as a soft constraint, and thereby reduce replanning solution quality to partial-satisfaction planning solution quality. Starting from this high level discussion, we present a formal model of replanning with failures and commitments in terms of metric-temporal partial-satisfaction planning. Before concluding, we discuss some of the connections to related work in plan reuse and real-world systems.

Correctness: Execution Failure

The ultimate goal in replanning is to produce a new executable, relevant, failure-resolving plan. Unfortunately, traditional planning theories only verify two of these properties: executability from an initial state, and relevancy to a goal description. To ensure that solutions further resolve failures, some extension of planning theory is required. Here, we are motivated by the concept of replanning as “planning again”. That is, while some researchers (Hammond 1986; Simmons 1988) consider augmenting traditional planning theories with techniques for directly resolving failures, we instead consider reducing the replanning problem to a modified planning problem.

These modifications to the planning problem are to ensure that executability and relevancy with respect to the modified instance, together, implies executability, relevancy, and failure-resolution with respect to the executor’s updated theory. This is trivial when the executor has no ability to learn from its mistakes; unexpected dynamics are simply represented as changes to the initial state and to the goal description. Given that we have assumed an environment where failures occasionally occur, executors incapable of learning are doomed for all but the most restricted case: those environments where failures are entirely independent of the behavior of the agent.

For example, let’s suppose that Romeo has used AltAlt^{PDA} to synthesize a travel plan to Los Angeles, specifically, one which involves driving. Suppose Romeo attempts to start the car, and that the attempt fails. If we restrict ourselves to representing failures as changes in the initial state and the goal description, then Romeo has a dilemma: the initial state is unchanged, as is the goal. The old plan would still be valid! It is nonetheless clear that the old plan is inappropriate; specifically, Romeo would possess the knowledge that future attempts to start the car will continue to fail.

For these reasons, we describe a failure as a fine-grained modification to the planning instance. The semantics of traditional planning theories are typically given in terms equivalent to Markov Decision Processes, where the planner chooses among several transition matrices at each state. In the case of deterministic planning, for example, such matrices consist of only 0’s and 1’s, whereas in probabilistic planning, rows encode the posterior distribution of states given the execution of an action. For our purposes, we say a failure can be described as a set of modifications to such matrices. As an example, the transition matrix for driving, in the replanning scenario described above, would be altered by the failure to the identity matrix: every attempt at driving will

fail to achieve a change in state. After applying such a set of modifications, the definition of executability further entails failure-resolution: so we have reduced replanning correctness to planning correctness.

Quality: Commitments

In many, if not all, potential applications of automated planning, plan quality is just as important as plan correctness. We can assume that an appropriate quality metric has been provided for the original planning domain, but after an execution failure, it is not clear if that metric is still appropriate. For example, many authors claim that a more appropriate quality metric is preservation of prior plan structure. Some high-level justifications for ignoring the old quality metric in favor of prior plan syntax include rescheduling cost, commitment preservation, and user acceptance in mixed-initiative planning. In fact, these three explanations are really but different facets of the presence of external, collaborative, unmodeled agents.

This raises some important questions:

1. Why did the original model omit the external agents?
2. How does plan failure invalidate that reason?
3. What is the right model of this expanded quality metric?

Full multi-agent planning and execution is very complex. It is difficult to formalize, acquire models, and find solutions. Moreover, in many settings, the presence of other agents ends up being irrelevant, as an efficient plan exists to achieve the goal independently of the external agents. For these reasons, it is common to project away all external agents, including friends, in order to simplify a complex multi-agent problem into a single-agent planning problem. In our running example, it is clearly the case that Romeo, or his assistant, can find efficient travel plans without consulting potential collaborators.

Before any decisions have been reached, there are no dependencies between agents at all. A friend can hardly be justified in complaining if one fails to rendezvous, when no rendezvous was arranged. Sometime after the a plan has been constructed, however, it is possible that communication of the plan took place. If so, then the agent might now be committed to some of those decisions. For example, suppose we consider a scenario where Romeo is traveling to Los Angeles to attend a conference. Sometime during the execution, he learns that the conference has been canceled, as the hotel has burned down. If, after making the travel plan, Romeo ended up making arrangements to meet with his friend Bob, who dwells in Los Angeles, then even after the hotel has burned down Romeo does still have a commitment to meet Bob at the prearranged place in Los Angeles.

Suppose we try to capture this altered notion of plan quality with minimal perturbation planning; the new quality metric rewards plans which preserve old structure. In the scenario where the hotel burns down, preserving old plan structure keeps all the traveling actions, thus allowing Romeo to meet with Bob in Los Angeles: preserving the commitment. Of course, even if there had been no commitment, the traveling would still have been performed – without any benefit

to Romeo at all. Moreover, the cost of driving and arranging new lodging may far outweigh the cost of canceling the meeting with Bob. Clearly, arbitrarily preserving syntax is not adequately capturing a measure of plan quality.

Instead, we say that a commitment introduces a new goal in the altered problem instance. Typically, this would be a *soft constraint*, as failing to meet a commitment is not often grounds for completely giving up on any course of action. So, in fact, we are considering representing replanning as alterations to an underlying partial-satisfaction planning problem: where the soft constraints are commitments.

These commitments arise after $\text{AltAlt}^{\text{PDA}}$ automatically synthesizes a plan for Romeo. At that point, Romeo might communicate some aspects of this plan to the many external agents he interacts with. These agents can then request that Romeo *commit* to some or all of those communicated aspects. Given that Romeo agrees to do so, then he would be able to model this commitment as a new goal for $\text{AltAlt}^{\text{PDA}}$ to achieve. Of course, the current plan already achieves any such commitment, so there isn't any need to alter the plan in response to the new goal until after a failure occurs. However, after a failure occurs, the goals introduced by external commitments serve as an adequate measure of plan quality; failing to achieve such a goal fails to maintain the associated commitment (and so fails to accrue the associated reward).

Formal Treatment

To fully ground our presentation of failures and commitments we demonstrate altering instances of metric-temporal partial-satisfaction (van den Briel *et al.* 2004) planning problems to account for execution failures and commitments to external agents. We use metric time to properly motivate the interactions between collaborating agents; weaker forms of planning do not admit a description of some method to synchronize the activities of agents. We presume that an appropriate interface exists (such as a parser, or a GUI) between the top-level user and the automated replanner, specifically, we will assume that the input to the replanner representing failures and commitments has already been formalized in the following manner:

Definition 1 (Failure) A failure, f , is a 4-tuple, (S, A, P, E) , of strings in PDDL syntax. S and P are goal descriptions, A is an operator name followed by a (parenthesized) list of variables and constants, and E is an effect. A failure is activated when the agent attempts to execute some binding of A in some state satisfying S ; the activation alters the preconditions and effects of that action application. Specifically, the action is only executable if it further satisfies P , and the resultant state is obtained by additionally applying the effect E . Conflicts between E and the normal effects of the action are resolved in favor of E : E is applied after the normal effects of the action. However, antecedents of conditional effects in E are still evaluated with respect to the original state, despite the fact that E is applied to an intermediate state.

For example, suppose Romeo notices that $\text{drive}(\text{carA}, \text{PHX}, \text{LA})$ has failed, and further surmises that the fault is due to carA being broken. This failure could

be represented as: $(\text{true}, \text{drive}(\text{carA}, ?x, ?y), \text{true}, \text{no-op})$. That is, in every state of the domain, attempting to drive using carA will activate the failure. The failure manifests itself by causing the drive action to be ineffectual; *no-op* is an abbreviation for the set of conditional effects of the form $(\text{when } p \text{ } p)$ for every literal p . That is, the resultant state must be identical to the original state, so that attempting to drive with carA becomes a no-op.

Definition 2 (Commitment) A commitment to an external agent, $c = (G, r)$, is a 2-tuple consisting of a goal description and an associated reward (which can be a special symbol infinity for a hard goal).

In the context of metric-temporal planning, we assume goals are given as achievement formula within a single window of opportunity, where the right side, the deadline, is typically the more challenging constraint to satisfy. Goal-achievement of a plan is then defined as achieving each formula at some point within each associated window. The reward accrued by achieving soft goals is simply the sum of the associated rewards, and the utility of a plan is its aggregate reward minus its aggregate cost; aggregate cost is simply the sum of the costs of each action.

From here, we can modify the ground instance of a partial-satisfaction metric-temporal planning problem by adding the commitments to the list of goals and altering the ground operators according to the failure descriptions. The last step sometimes involves splitting ground operators, as a ground operator actually corresponds to many transitions in the graph; a failure can theoretically modify just a single transition.

Example: Suppose Romeo's summer plans not only included the conference in Los Angeles, but also a short educational visit to Boston a month later. Further suppose that Romeo has stipulated a large number of additional places to visit in both Los Angeles and Boston, so that his assistant, $\text{AltAlt}^{\text{PDA}}$, has generated a custom walking tour of both cities, in addition to the rest of the travel plan. Romeo has told Bob of his plans, and they mutually agreed to meet at the California Science Center at noon. Fate is kinder than in the preceding sections: the only failure encountered is that Romeo discovers that he has forgotten to pack his sneakers. As he is hardly willing to engage in the walking tour of Los Angeles in dress shoes, he decides to use $\text{AltAlt}^{\text{PDA}}$ to replan. So, in this case, the tour should be regenerated, favoring public transportation over walking, and preferring to keep the noon appointment with Bob. In uncomfortable shoes, suppose Romeo is willing to walk half as far (500 m.) to a specific destination, and only up to a quarter (5 km.) of the normal maximum distance in any given day. Since Romeo expects to remember to pack comfortable shoes for the trip to Boston, he limits this failure to his current stay in Los Angeles.

The formal representation of the failure is given by: $(\text{at}(\text{Los Angeles}), \text{walk}(?s, ?d), (\text{and}(< \text{distance}(?s, ?d) 500) (< \text{walked_today}(-2500 \text{distance}(?s, ?d))))), \text{nil})$. Likewise, the commitment to meet Bob is just $(\langle \text{at}(\text{California Science Center}), \text{noon} - 5 \text{ min}, \text{noon} + 5 \text{ min} \rangle, r)$, where r is presumably large, since canceling the day of an appointment

with the weak excuse of not having comfortable shoes would probably offend Bob, let alone inconvenience him.

The ground planning instance would consist of all the old goals, along with the commitment to Bob, as well as a split version of the walk operator. The original walk operator $walk_{normal}$, would have an additional precondition of being applicable only outside of Los Angeles. The walking in uncomfortable shoes operator, $walk_{fail}$, would be applicable in Los Angeles, but would additionally require that the destination be half as far, and that the total distance traveled would be less than a quarter of its usual maximum.

Synthesizing an entirely new plan with respect to this planning problem resolves the failure; the Los-Angeles specific goals would be accomplished by alternative means, where necessary, including the explicit commitment to Bob. After leaving Los Angeles the planning instance behaves the same as when $AltAlt^{PDA}$ generated the original plan for Romeo. Therefore, the Boston portion of the plan would remain unaffected, just as in minimal perturbation planning. However, minimal perturbation planning would often miss the commitment to Bob. Suppose Romeo is not visiting any attractions within 500 meters of the California Science Center. Keeping the commitment requires perturbing the prior plan: walking to, from, or past the science center cannot be reused.

Related Work

Proponents of replanning as plan reuse, as exemplified by (van der Krogt & de Weerd 2005), have doubly confused the issue of replanning quality. The first approximation, considered at length in this paper, is in assuming that reducing perturbation caused to other agents can be modeled by minimally altering the structure of plans across iterations: minimal perturbation planning. The second approximation is applying their plan reuse algorithm instead of truly minimal perturbation planning. That is, minimal perturbation planning is just as inadequate in application to plan reuse as it is in application to replanning – though for different reasons (Nebel & Koehler 1995). Specifically, minimal perturbation planning has greater complexity than plan synthesis, which is in direct conflict with the speedup motivation of plan reuse. As noted by Nebel & Koehler, plan reuse systems actually return highly, but not maximally, similar plans.

The robot path planning community (Stentz 1995; Koenig, Likhachev, & Furcy 2004) has long looked at replanning slightly differently from the planning community. In particular they try to ensure that the plan produced by the “re-planner” is as optimal as the one that would have been produced by the from-scratch planner. Due to the nature of robot path planning, this work does not consider the commitments made by the partial execution of the prior plan. One point of similarity is that the robot path planning community does model failures that involve more than initial and goal state changes—action deletion (e.g. certain navigation actions made infeasible by the appearance of new obstacles). This is a kind of systematic failure, which we generalize further based on accounts of real-world replanning systems (Pell et al. 1997; Myers 1999).

Within the planning community, the work by Pell et al. comes closest to recognizing the importance of respecting the commitments generated by a partially executed plan. They however do not give any formal details about how the replanning is realized in their system.

Conclusion

The observation underlying our work is that replanning lacks a formal definition. At a high level, researchers bring up the concepts of commitments, reservations, approximate domain models, mixed-initiative and distributed planning, tightly bounded computational resources, and execution failures. Then a syntactic measure is introduced without any formal connection to these motivations. While approximating replanning using minimal perturbation planning could work well empirically, demonstrating this first requires an idea of non-approximate replanning.

We tackle this problem by considering the intuitively optimal behavior for a set of interesting replanning scenarios. We find that it is easy to construct scenarios where optimal behavior can be explained in terms of systematic execution failures and commitments to external agents. We develop a formal model of replanning that directly captures systematic failures and commitments, and admits as formal solutions only those that properly respond to the additional constraints. Given that those are transcribed correctly, then the formal solution of the model corresponds to the intuitively optimal behavior first described. Our model thus serves as a formal definition of the kind of replanning concerned with resolving execution failures while respecting commitments.

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 Weerd, M. 2005. Plan repair as an extension of planning. In *ICAPS*.