Safe Reduced Models for Probabilistic Planning

Sandhya Saisubramanian and **Shlomo Zilberstein**
College of Information and Computer Sciences
University of Massachusetts Amherst
{saisubramanian,shlomo}@cs.umass.edu

Abstract

Reduced models allow autonomous agents to cope with the complexity of planning under uncertainty by reducing the accuracy of the model. However, the solution quality of a reduced model varies as the model fidelity changes. We present *planning using a portfolio of reduced models with cost adjustments*, a framework to increase the safety of a reduced model by selectively improving its fidelity in certain states, without significantly compromising runtime. Our framework provides the flexibility to create reduced models with different levels of detail using a portfolio, and a means to account for the ignored details by adjusting the actions costs in the reduced model. We show the conditions under which cost adjustments achieve optimal action selection and describe how to use cost adjustments as a heuristic for choosing outcome selection principles in a portfolio. Finally, we present empirical results of our approach on three domains that includes an electric vehicle charging problem using real-world data from a university campus.

1 Introduction

Many real-world problems that require sequential decision making under uncertainty are often modeled as Stochastic Shortest Path (SSP) problems [Bertsekas and Tsitsiklis, 1991]. Given the computational complexity of solving large SSPs optimally [Littman, 1997], there has been much interest in developing efficient approximations, such as reduced models, that trade solution quality for computational gains [Yoon *et al.*, 2008]. Reduced models simplify the problem by partially or completely ignoring uncertainty, thereby reducing the set of reachable states a planner needs to consider [Yoon *et al.*, 2010; Keller and Eyerich, 2011]. We consider reduced models in which the number of outcomes per action is reduced relative to the original model.

While this reduction in reachable states accelerates planning, it affects the solution quality, particularly if "risky" states — states that significantly affect the expected cost of reaching a goal — are not preserved in the reduced model. Thus, the action outcomes considered in the reduced model determines the model fidelity and thereby the solution quality. In this paper, we associate the notion of safety with solution quality. We consider a reduced model to be *safe* if it preserves the safety guarantees contained in the original problem by fully accounting for the risky outcomes in the reduced model. Thus, a safe reduced model results in improved plan quality. The existing reduced model techniques have focused on formulating models that reduce planning time, but they do not focus on formulating safe reduced models [Yoon *et al.*, 2007; Keyder and Geffner, 2008]. This limits the applicability of reduced models to many problems that inherently require fast and safe (high-quality) plans. Examples of such domains include wildfire response and various forms of agent interaction with humans such as semi-autonomous driving [Hajian *et al.*, 2016; Wray *et al.*, 2016]. While the model fidelity can be improved by considering the full model in planning, it defeats the purpose of using reduced models. The key question we address in this work is how to formulate a safe reduced model that balances this trade-off.

Intuitively, the trade-off between model simplicity and safety can be optimized by learning when to use a simple model and when to use a more informed model. Consider for example a robot navigating through a building. A plan generated by a simple reduced model might work well when the robot is moving through uncluttered region, but a more informative reduced model or the full model may be required to reliably navigate through a narrow corridor [Styler and Simmons, 2017]. The existing reduced model techniques are incapable of handling such variations in detail, since they employ a uniform approach to determine the number of outcomes and how they are selected for all (s, a) in the reduced model. This limits the scope of the risks they can represent, resulting in sub-optimal solutions. Furthermore, the unaccounted outcomes of an action that are ignored in the reduced model lead to overly optimistic plans. Since the existing techniques do not guarantee bounded-optimal performance, it is hard to predict when they will work well.

This paper formulates safe reduced models by learning to select outcomes for planning. We present two techniques that complement each other in formulating a safe reduced model, without compromising the runtime gains of using a reduced model. First, we introduce *planning* using a *portfolio of reduced models* (PRM), that enables formulating reduced models with different levels of details by using a portfolio of outcome selection principles (Section 3). Secondly, we present

planning using cost adjustments, a technique that improves the solution quality of reduced models by altering the costs of actions to account for the consequences of ignored outcomes in the reduced model (Section 4). Since it is non-trivial to compute the exact cost adjustments, we propose an approximation that *learns* the cost adjustments from samples. Furthermore, the cost adjustments offer a heuristic for choosing the outcome selection principles in a PRM (Section 5). Finally, we empirically demonstrate the benefits of our approach in three different domains including an electric vehicle charging problem using real world data, and two benchmark planning problems (Section 6).

2 Planning Using Reduced Models

We target problems modeled as a Stochastic Shortest Path (SSP) MDP, defined by $M = \langle S, A, T, C, s_0, S_G \rangle$, where S is a finite set of states; A is a finite set of actions; $T(s, a, s') \in [0, 1]$ denotes the probability of reaching a state s' by executing an action a in state s; $C(s, a) \in \{\mathbb{R}^+ \cup \{0\}\}$ is the cost of executing action a in state s; $s_0 \in S$ is the initial state; and $S_G \subseteq S$ is the set of absorbing goal states. The cost of an action is positive in all states except goal states, where it is zero. The objective in an SSP is to minimize the expected cost of reaching a goal state from the start state. The optimal policy, π^*, can be extracted using the value function defined over the states, $V^*(s)$:

$$V^*(s) = \min_a \; Q^*(s, a), \quad \forall s \in S \tag{1}$$

where $Q^*(s, a)$ denotes the optimal Q-value of the action a in state s and is calculated as, $\forall (s, a) \in S \times A$:

$$Q^*(s, a) = C(s, a) + \sum_{s'} T(s, a, s') V^*(s'). \tag{2}$$

While SSPs can be solved in polynomial time in the number of states, many problems have a state-space whose size is exponential in the number of variables describing the problem [Littman, 1997]. This complexity has lead to the use of approximation techniques such as reduced models for planning under uncertainty. Reduced models simplify planning by considering a subset of outcomes. Let $\theta(s, a)$ denote the set of all outcomes of (s, a), $\theta(s, a) = \{s' | T(s, a, s') > 0\}$.

A **reduced model** of an SSP M is represented by the tuple $M' = \langle S, A, T', C, s_0, S_G \rangle$ and characterized by an altered transition function T' such that $\forall (s, a) \in S \times A, \theta'(s, a) \subseteq \theta(s, a)$, where $\theta'(s, a) = \{s' | T'(s, a, s') > 0\}$ denotes the set of outcomes in the reduced model for action a in state s. We normalize the probabilities of the outcomes included in the reduced model, but more complex ways to redistribute the probabilities of ignored outcomes may be considered. The outcome selection process in a reduced model framework determines the number of outcomes and how the specific outcomes are selected. Depending on these two aspects, a spectrum of reductions exist with varying levels of probabilistic complexity that ranges from the single outcome determinization to the full model [Keller and Eyerich, 2011].

An *outcome selection principle* (OSP) performs the outcome selection process per state-action pair in the reduced model, thus determining the transition function for the state-action pair. The OSP can be some simple function such as always choosing the most likely outcome or a more complex function. Traditionally, a reduced model is characterized by a single OSP. That is, a single principle is used to determine the number of outcomes and how the outcomes are selected across the entire model. A simple example of this is the most-likely outcome determinization.

3 Portfolio of Reduced Models

We define a generalized framework, *planning using a portfolio of reduced models*, that facilitates the creation of safe reduced models by switching between different outcome selection principles, each of which represents a different reduced model. The framework is inspired by the benefits of using portfolios of algorithms to solve complex problems [Petrik and Zilberstein, 2006].

Definition 1. *Given a portfolio of finite outcome selection principles,* $\mathrm{Z} = \{\rho_1, \rho_2, ..., \rho_k\}$, $k > 1$, *a* ***model selector****,* Φ, *generates* T' *for a reduced model by mapping every* (s, a) *to an outcome selection principle,* $\Phi : S \times A \rightarrow \rho_i$, $\rho_i \in \mathrm{Z}$, *such that* $T'(s, a, s') = T_{\Phi(s,a)}(s, a, s')$, *where* $T_{\Phi(s,a)}(s, a, s')$ *denotes the transition probability corresponding to the outcome selection principle selected by the model selector.*

Trivially, the model selector used by the existing reduced models is a special case of the above definition, as Φ always selects the same ρ_i for every state-action pair. Hence, the model selectors of existing reduced models are incapable of adapting to the risks. Typically, in *planning using a portfolio of reduced models* (PRM), the model selector utilizes more than one OSP to determine T'. Each state-action pair may have a different number of outcomes and a different mechanism to select the specific outcomes. We leverage this flexibility in outcome selection to formulate safe reduced models by using more informative outcomes in the risky states and using simple outcome selection principles otherwise. Although the model selector could use multiple ρ_i to generate T' in a PRM, the resulting model is still an SSP.

Definition 2. *A* ***0/1 reduced model*** *(0/1 RM) is a PRM with a model selector that selects either one or all outcomes of an action in a state to be included in the reduced model.*

A 0/1 RM is characterized by a model selector, $\Phi_{0/1}$, that either ignores the stochasticity completely (0) by considering only one outcome of (s, a), or fully accounts for the stochasticity (1) by considering all outcomes of the state-action pair in the reduced model. For example, it may use the full model in states prone to risks or states crucial for goal reachability, and determinization otherwise. Thus, a 0/1 RM that guarantees goal reachability with probability 1 can be devised, if a proper policy exists in the SSP. Our experiments using 0/1 RM show that even this basic instantiation of a PRM works well in practice.

Depending on the model selector and the portfolio, a large spectrum of reduced models exists for an SSP and choosing the right one is non-trivial.

3.1 Model Selector (Φ)

Typically, the model selectors in existing reduced models have been devised to improve the runtime of the reduced models. We aim to devise a model selector whose objective is to account for the risky outcomes in the reduced model without significantly compromising the runtime benefits of using a reduced model. In a 0/1 RM, frequently using the full model may over-complicate the planning process, while always using a single outcome determinization may oversimplify the problem. An efficient model selector selects OSPs for each state-action pair such that the trade-off between solution quality and planning time is optimized.

Devising an efficient model selector automatically can be viewed as a meta-level decision problem that is computationally more complex than solving the reduced model, due to the numerous possible combinations of outcome selection principles. Even in the simple case of 0/1 RM, devising an efficient Φ is non-trivial as it involves deciding when to use the full model and when to use determinization. In the worst case, all the OSPs in Z may have to be evaluated. Let τ_{max} denote the maximum time taken for this evaluation across all states. The OSPs may be redundant in terms of specific outcomes. For example, selecting the most likely outcome and greedily selecting an outcome based on Q-values could result in the same outcome for a (s, a) pair. If every outcome selection principle specifies a unique outcome, then the time taken to devise an efficient Φ could be exponential in the number of states. While this is a trivial fact, it is useful to understand the worst case complexity of devising an efficient model selector as it provides an important link to the need for efficient evaluation metrics. Proposition 1 formally proves this complexity.

Proposition 1. *The worst case time complexity for a model selector, Φ, to generate T' for a PRM is $\mathcal{O}(|A| \cdot 2^{|S|} \cdot \tau_{max})$.*

Proof Sketch. For each (s, a), at most |Z| OSPs are to be evaluated and this takes at most τ_{max} time (as mentioned above). Since this process is repeated for every (s, a), Φ takes $\mathcal{O}(|S||A||\mathrm{Z}|\tau_{max})$ to generate T'. In the worst case, every action may transition to all states, $T(s, a, s') > 0, \forall (s, a, s') \in M$, and the OSPs in Z may be redundant in terms of the number and specific outcomes set produced by them. Hence, the evaluation is restricted to the set of unique outcomes sets denoted by k, $|k| \leq |\mathcal{P}(S)|$, with $\mathcal{P}(S) = 2^{|S|}$. Then, it suffices to evaluate the $|k|$ outcome sets instead of |Z|, reducing the complexity to $\mathcal{O}(|A| \cdot 2^{|S|} \cdot \tau_{max})$. □

This shows that the complexity is independent of $|Z|$, which may be a very large number in the worst case.

Corollary 1. *The worst case time complexity for $\Phi_{0/1}$ to generate T' for a 0/1 RM is $\mathcal{O}(|A| \cdot |S|^2 \cdot \tau_{max})$.*

Proof Sketch. This proof is along the same lines as that of Proposition 1. To formulate a $0/1$ RM of an SSP, it may be required to evaluate every $\rho_i \in$ Z that corresponds to a determinization or a full model. Hence, in worst case, $\Phi_{0/1}$ takes $\mathcal{O}(|S| \cdot |A| \cdot |\mathrm{Z}| \cdot \tau_{max})$ to generate T'. The set of unique outcomes, k, for a $0/1$ RM is composed of all unique deterministic outcomes, which is every state in the SSP, and the full model, $|k| \leq |S| + 1$. Replacing $|Z|$ with $|k|$, the complexity is reduced to $\mathcal{O}(|A| \cdot |S|^2 \cdot \tau_{max})$. □

Since τ_{max} could significantly reduce the runtime savings of using the reduced model, these results underscore the need for developing faster evaluation techniques to identify relevant outcome selection principles. This would be particularly useful in automated generation of efficient model selectors. In this paper, we focus on creating reduced models that yield high quality results using the existing OSPs from the literature. Therefore, future improvements in OSPs can be leveraged by PRMs.

In the following section, we propose a technique that accounts for the outcomes ignored in the reduced model by adjusting the action costs. We also explain how this acts as a heuristic for selecting OSPs in a PRM, allowing us to reasonably balance the trade-off between solution quality and planning time, as in our experiments.

4 Reduced Models with Cost Adjustments

One of the reasons for existing reduced model techniques producing poor solutions is that some outcomes are completely ignored. In fact, certain ways of accounting for the ignored outcomes could result in optimal action selection for the SSP. Traditionally, only the transition function is altered in a reduced model. To account for the ignored outcomes, we propose a technique that alters the costs of actions in the reduced model. We introduce *planning using cost adjustment*, a technique that accounts for the ignored outcomes by adjusting the costs of actions in the reduced model, thus resulting in optimal action selection in the reduced model.

Definition 3. *A **cost adjusted reduced model** (CARM) of an SSP M is a reduced model represented by the tuple $M'_{ca} = \langle S, A, T', C', s_0, S_G \rangle$ and characterized by an altered cost function C' such that $\forall (s, a)$ in reduced model,*

$$C'(s, a) \leftarrow Q^*(s, a) - \sum_{s' \in \theta'(s,a)} T'(s, a, s') V^*(s').$$

Given an SSP and its reduced model (not necessarily a PRM), the costs are adjusted for every (s, a) in the reduced model to account for the ignored outcomes. Since the costs are adjusted based on the difference in values of states, this may lead to negative cost cycles in an SSP. Therefore, the necessary and sufficient condition for non-negative cost in CARM is that T' satisfies

$$Q^*(s, a) \geq \sum_{s' \in \theta'(s,a)} T'(s, a, s') V^*(s'). \tag{3}$$

This condition may be relaxed as long as there are no negative cost cycles in the reduced model. The optimal state values and action values in M'_{ca} are denoted by $V^*_R(s)$ and $Q^*_R(s, a)$, and its optimal policy by π^*_R. Let X^{π}_R and X^{π} denote the set of states reachable by executing a policy π in M'_{ca} and executing π in M, respectively. Since $\theta'(s, a) \subseteq \theta(s, a)$, we get $X^{\pi}_R \subseteq X^{\pi}$.

Lemma 1. *Given a CARM and policy π, $\forall s \in X^{\pi}_R : V^{\pi}_R(s) = V^{\pi}(s)$, whose goal reachability is preserved in CARM.*

Proof Sketch. We show this using proof by induction on t starting from the goal state and following policy π (assuming proper policy). Trivially, the base case holds as we start from a goal. For readability, let $\theta'_{S_t,\pi} = \theta'(S_t, \pi(S_t))$, $S_{t=1} = s$ and $S_{t-1} = s'$. When $t = 1 : V^\pi(s) = C(s, \pi(s))$, $\forall s \in X^\pi$ and $V^\pi_R(s) = C'(s, \pi(s))$, $\forall s \in X^\pi_R$. Using π and Definition 3, we get $Q^\pi(s, a) = V^\pi(s)$ and $C'(s, a) = V^\pi(s)$. Therefore, $V^\pi_R(s) = V^\pi(s), \forall s \in X^\pi_R$. Thus, this holds true for $t = 1$.

Inductive Step: Assume true for $t - 1$ (induction hypothesis), must show that for t, $V^\pi_R(S_t) = V^\pi(S_t)$. Then,

$$V^\pi_R(S_t) = C'(S_t, \pi(S_t)) + \textstyle\sum_{s' \in \theta'_{S_t,\pi}} T'(S_t, \pi(S_t), s') V^\pi_R(s').$$

Using Definition 3 in the above,

$$V^\pi_R(S_t) = Q^\pi(S_t, \pi(S_t)) + \sum_{s' \in \theta'_{S_t,\pi}} T'(S_t, \pi(S_t), s') \\ (V^\pi_R(s') - V^\pi(s')).$$

By induction hypothesis, $V^\pi_R(s') = V^\pi(s')$, and for a fixed policy, π, $Q^\pi(S_t, \pi(S_t)) = V^\pi(S_t)$. Using these in the above equation, we get $V^\pi_R(S_t) = V^\pi(S_t)$. Thus, by induction, this holds true for all t, $V^\pi_R(s) = V^\pi(s), \forall s \in X^\pi_R$. □

If T' does not preserve the goal reachability (introduces dead ends by ignoring certain outcomes) for a state, then the expected cost of reaching the goal will be different in the original problem and CARM.

Proposition 2. *A CARM that preserves goal reachability yields optimal action selection for the SSP, if there exists a proper policy in the SSP.*

Proof. We prove this by showing that $\forall (s, a) \in M'_{ca}$, the optimal Q-values of the SSP and its cost adjusted reduced model are equal, $Q^*_R(s, a) = Q^*(s, a)$. However, if the reduced model introduces dead ends by ignoring certain outcomes (does not preserve goal reachability and has improper policy), then the Q-values will be different. Therefore, we restrict the proof to a CARM that preserves goal reachability. By definition, $\forall (s, a) \in S \times A$:

$$Q^*_R(s, a) = C'(s, a) + \sum_{s' \in \theta'(s,a)} T'(s, a, s') V^*_R(s').$$

Using Definition 4 in the above equation, we get

$$Q^*_R(s, a) = Q^*(s, a) + \sum_{s' \in \theta'(s,a)} T'(s, a, s') \Big(V^*_R(s') - V^*(s') \Big).$$

Since we assume a proper policy and for all states whose goal reachability is preserved in CARM, using Lemma 1 in the above equation yields $Q^*_R(s, a) = Q^*(s, a)$. □

Thus, a CARM that preserves goal reachability, produces optimal action selection for the SSP.

4.1 Approximate Cost Adjustments

Generating a CARM may involve solving the SSP to estimate the optimal values of the outcomes, which defeats the purpose of using reduced models. Hence, we propose an approximation for cost estimation, and the resultant reduced model with approximate costs is referred to as *approximately cost adjusted reduced model* (ACARM).

Learning feature-based costs In a factored state representation, the cost of an action can depend on a subset of state features [Boutilier *et al.*, 1999]. Along these lines, we propose estimating the costs based on features of the states.

Definition 4. *A **feature-based cost function** estimates the cost of an action in a state using the features of the state, $C'(s, a) = g(\vec{f}(s), a)$, where $g : \vec{f} \times A \to \mathcal{R}$.*

Let $\vec{f}(s) = \langle f_1(s), ..., f_n(s) \rangle$ denote features in a state s that significantly affect the costs of actions. Identifying such important features has been actively studied over the years in the context of state abstraction and value function approximation [Kolobov *et al.*, 2009; Mahadevan, 2009; Parr *et al.*, 2007], and machine learning techniques such as regression and decision stump [Shah *et al.*, 2012]. These techniques along with using domain knowledge offer a suite of methods to identify features that significantly affect the cost.

Given such features, the feature-based approximate costs are estimated by generating and solving sample problems. The samples are either known small problem instances in the target domain or generated automatically by sampling states from the target problem. In this paper, smaller problems are created by multiple trials of depth limited random walk on the target problems and solved using LAO* [Hansen and Zilberstein, 2001]. The feature-based costs are learned by computing the cost adjustments in hindsight for these samples using their exact solutions and the given features. The learned values are projected onto the target problem using the feature-based cost function. Trivially, as the number of samples and the depth of the random walk are increased, the estimates converge to their true values [Bonet and Geffner, 2003]. For problems with unavoidable dead ends, sampling states may not be a good representative of the target problem; smaller problem instances from the domain may be used instead.

State Independent Costs We also consider an extreme case, where the feature set characterizing each state is empty.

Definition 5. *A **state independent cost adjustment** assigns a constant cost per action, regardless of the state, resulting in a constant cost $C'(s, a) = g(a)$, where $g : A \to \mathcal{R}$.*

This simple form of generalization of the cost adjustment ignores the state altogether. In particular, PPDDL description of problems in a domain [Younes and Littman, 2004] have a shared action schema and hence having constant cost adjustment for actions in a problem instance can be extended to various problem instances in the domain. If the cost of an action, $C(s, a)$, and the relative discrepancy between the values of the outcomes of a are the same in every state, then the cost adjustment can be trivially generalized with a state independent cost adjustment.

Example 1. *Consider an SSP in which an action achieves a successful outcome with probability $1-p$ or fails with probability p, leaving the state unchanged. Let s denote a state for which a successful execution of action a with cost $C(s, a)$ results in outcome state s' [Keyder and Geffner, 2008].*

This example describes a class of problems for which state independent cost produces optimal action selection.

Proposition 3. *State independent cost adjustment results in optimal action selection for the class of problems identified in Example 1, for a fixed policy.*

Proof Sketch. Since the policy is fixed and a is stochastic, for a problem identified in Example 1,

$$Q^*(s,a) \;=\; \frac{C(s,a)}{1-p} + V^*(s').$$

To satisfy Equation 3, the failure outcome would be ignored in the reduced model. Substituting these in Definition 3,

$$C'(s,a) \;=\; \frac{C(s,a)}{1-p}. \tag{4}$$

This illustrates a class of problems for which state independent cost is accurate with optimal action selection. □

An example of Proposition 3 is the Blocksworld domain [Little and Thiebaux, 2007]. In this domain, given an initial configuration, the blocks need to be rearranged to satisfy some goal conditions. Since the actions are stochastic, an action, for example, "pick block" may be unsuccessful. If unsuccessful, the block slips and is dropped on the table. Since the relative discrepancy in the values of the outcomes is constant, a constant state independent cost exists. Consider the setting with unit cost actions that fail with a probability of 0.25. Empirically, regardless of the specific block, the state independent cost for this action is constant and our experimental results match the value of 1.33 obtained using Equation 4. Identifying domains and actions that have this property would alleviate pre-processing and help exploit the hidden structure in the given domain.

Thus, a good approximation can considerably improve the solution quality of an ACARM without affecting the planning time, as learning the costs is a pre-processing step.

5 Complementary benefits of the approaches

In this section, we discuss the complementary benefits of using a portfolio of reduced models and cost adjustments in formulating a safe reduced model. Specifically, we focus on two key aspects: (i) how the cost adjustments act as a heuristic for model selector in a PRM; and (ii) the benefits of using cost adjusted actions in a PRM. For the sake of clarity and simplicity, we discuss these in the context of 0/1 RM with a portfolio consisting of the most likely outcome determinization and the full model. However, the extension to a richer portfolio is straightforward.

5.1 Model selection guided by cost adjustments

Typically, ignoring states with higher expected costs of reaching the goal in the reduced model results in higher cost adjustment value. Ignoring such outcomes in the reduced model results in an optimistic view of the problem. Since the cost adjustment value reflects the criticality of a state for goal reachability, it can be used as a heuristic for the model selector in a portfolio of reduced models. For example, a $\Phi_{0/1}$ can be designed such that it employs the full model in the states with high cost adjustment values, and determinization in other states.

By altering the cost adjustment threshold at which the full model is triggered, reduced models with different levels of sensitivity to risks may be produced. This also produces reduced models with possibly different levels of computational gains and solution quality, due to the difference in fraction of full model usage in the reduced model.

5.2 Cost Adjusted Actions in a PRM

To understand the need for combining a PRM with cost adjusted actions, we discuss the drawbacks in formulating a safe reduced model with each approach independently.

In a 0/1 RM, the model selector would aim to minimize the use of full model to reduce the planning time, by employing a full model at critical states, and determinization otherwise. The states using the most likely outcome determinization may affect the solution quality in the following ways. First, it is possible that the most likely outcome determinization in some states could prevent the planner from reaching or expanding these critical states in the search phase. Second, the optimal policy in the states with the full model cannot compensate for the poor solutions produced by states using the most likely outcome determinization. Because of these two reasons, a 0/1 RM may still result in poor solution quality despite using the full model sparingly in critical states.

The primary motivation for using approximate costs is that calculating the exact cost adjustments without solving the problem is non-trivial. Since the feature-based approximate costs do not guarantee bounded performance, using a cost adjusted determinization alone does not guarantee optimal or near-optimal solutions. However, future advancements in techniques that compute the cost adjustment without solving the problem or compute approximate cost adjustments with bounded errors may be leveraged to produce safe reduced models without using a portfolio. With the current machinery, a cost adjusted determinization alone may be insufficient to formulate a safe reduced model.

These illustrate the need for augmenting a 0/1 RM with cost adjusted actions. Our experiments show that using a 0/1 RM with cost adjustment both as a heuristic for a model selector and to adjust the costs of the actions in states using determinization produces safe reduced models that yield almost optimal results.

6 Experimental Results

We experiment with the approximately cost adjusted 0/1 RM (ACARM-0/1 RM) on three domains including an electric vehicle charging problem using real world data from a university campus, and two benchmark planning problems: racetrack domain and sailing domain. The aim of these experiments is to demonstrate that planning using a portfolio of reduced models with cost adjustments improves the solution quality without compromising the runtime gains. Therefore, we experiment with a 0/1 RM and a simple portfolio $Z = \{$most likely outcome determinization (MLOD), full model$\}$. We compare the results of ACARM-0/1 RM with the results obtained by solving:

- a 0/1 RM of the problem using the cost adjustment values as a heuristic for model selector;
- the models formed by using each OSP in the portfolio independently, that is MLOD and full model only, with and without cost adjustment; and
- the original problem using FLARES, a state-of-the-art domain-independent algorithm, with horizon 0 and 1 [Pineda *et al.*, 2017].

We compare our results with that of FLARES as it is short-sighted labeling based algorithm, which is another popular approach to solve large SSPs apart from reduced models. We evaluate the results in terms of plan quality, which is the expected cost of reaching the goal and planning time. In the domains used in our experiments, the most likely outcome is also the most desirable outcome, thus providing an optimistic baseline for comparison.

The approximate costs are estimated using a feature-based cost function that uses simple and intuitive state features, identified by us. Estimating feature-based costs is required only once per domain and the scalability is preserved as we limit the size of the sampled problems. These costs are also used as a heuristic for the model selector in the 0/1 RM. Note that the 0/1 RM uses the approximate costs only for the model selector and the costs are unaltered, while an ACARM-0/1 RM uses the feature-based costs for the model selector and to alter the action costs.

All results are averaged over 100 trials of planning and execution simulations and the average times include re-planning time. Standard errors are reported for expected cost. The deterministic problems are solved using the A* algorithm [Hart *et al.*, 1968], and other problems using LAO*, and complemented by re-planning. All algorithms were implemented with $\epsilon{=}10^{-3}$ and using h_{min} heuristic computed using a labeled version of LRTA* [Korf, 1990].

6.1 EV Charging Problem

We experimented with the electric vehicle (EV) charging domain, operating in a vehicle-to-grid setting [Saisubramanian *et al.*, 2017], where the EV can charge and discharge energy from a smart grid. By planning when to buy or sell electricity, an EV can devise a robust policy for charging and discharging that is consistent with the owner's preferences, while minimizing the long-term operational cost of the vehicle.

We modified the problem to increase the difficulty such that parking duration of the EV is uncertain and is denoted by a distribution, indicating that certain states could become a terminal state with some probability. Therefore, the maximum parking duration is the horizon, H. Each state is represented by $\langle l, t, d, p, e\rangle$, where l is the current charge level, $t \leq H$ is the time step, d and p denote the current demand level and price distribution for electricity respectively, and $0 \leq e \leq 3$ denotes the departure communication from the owner. If the owner has not communicated, then $e{=}3$ and the agent plans for H. Otherwise, e denotes the time steps remaining for departure. The process terminates when $t{=}H$ or if $e{=}0$.

We experimented with four demand levels, and two price distributions. Each t is equivalent to 30 minutes in real time. We assume that the owner is most likely to depart between four to six hours of parking with communication probability as 0.2. For all other t, the owner communicates with probability 0.05. The charging costs and the peak hours are based on real data [Eversource, 2017]. The battery capacity and the charge speeds for the EV are based on Nissan Leaf configuration. We assume the charge and discharge speeds to be equal. The battery inefficiency is accounted for by adding a 15% penalty on the costs. The feature-based costs are estimated using state features and one-step lookahead. The features include the time remaining for departure, if the current time is peak or not, and if the current charge level is sufficient to discharge. For all states with highest feature-based costs in this domain, the model selector uses a full model. In all other states, MLOD is used. In our experiments, we observe that this results in using MLOD until one hour from departure, and then a full model is triggered.

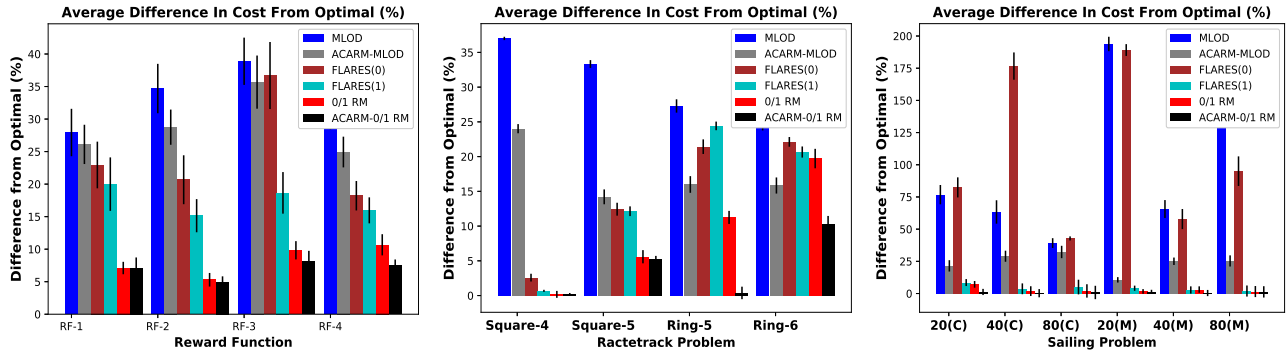
EV Dataset The data used in our experiments consist of charging schedules of electric cars over a four month duration in 2017 from the UMass Amherst campus. The data is clustered based on the entry and exit charges, and we selected 25 problem instances across all clusters for our experiments, based on frequency of occurrence in the dataset. The data is from a typical charging station, where the EV is unplugged once the desired charge level is reached. Since we are considering an extended parking scenario (e.g., parking at work), we assume a parking duration of up to eight hours. Therefore, for each problem instance, we only alter the parking duration and retain the charge levels and entry time from the dataset.

6.2 Racetrack Domain

We experimented with four problem instances from the racetrack domain [Barto *et al.*, 1995], with modifications to increase the difficulty of the problem. We modified the problem such that, in addition to a 0.10 probability of slipping, there is a 0.20 probability of randomly changing the intended acceleration by one unit. The feature-based costs use one-step lookahead and state features such as: whether the successor is a wall or pothole or goal, and if the successor is moving away from goal, which can be estimated using heuristic value. The feature-based costs serve as a heuristic for the model selector. For states with highest cost adjustments, a full model is used. Otherwise, determinization is used.

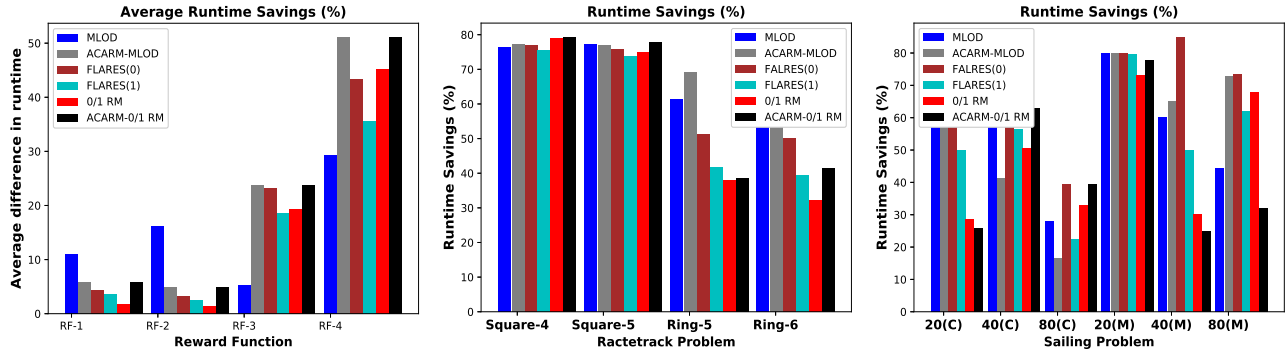
6.3 Sailing Domain

Finally, we present results on six instances of the sailing domain [Kocsis and Szepesvári, 2006]. The problems vary in terms of grid size and the goal position (opposite corner (C) or middle (M) of the grid). In this domain, the actions are deterministic and uncertainty in the domain is caused by stochastic changes in the direction of the wind. Each action's cost depends on the direction of movement and direction of the wind. The feature-based costs are estimated using one-step lookahead and based on state features such as: the difference between the action's intended direction of movement and the wind's direction, and if the successor is moving away from goal, which can be estimated using heuristic value. The model selector uses the full model in all states with the highest cost adjustment in this domain. In all other states, MLOD is used.



(a) EV Domain (b) Racetrack Domain (c) Sailing Domain

Figure 1: Average (%) Cost Difference



(a) EV Domain (b) Racetrack Domain (c) Sailing Domain

Figure 2: Average (%) Runtime Savings

6.4 Discussion

Figures 1(a), 1(b), and 1(c) show the average differences in cost (%) of the six techniques on the three domains. For the EV domain, the results are aggregated over 25 problem instances for each reward function. The *lower values* indicate that the performance of the technique is closer to the optimal value. In many problems, 0/1 RM and ACARM-0/1 RM yield almost optimal results.

Table 1 shows the full model usage (%) in the 0/1 RM using cost adjustment as heuristic for model selector. For most problem instances, we achieve near-optimal solutions by sparingly using the full model. However, for the sailing domain, many states have a high cost adjustment value since the costs of actions depend on the direction of the wind. Since we use the full model in states with highest cost adjustments in the domain, the fraction of full model usage is relatively high in this domain. By altering the cost adjustment threshold at which the full model is triggered, the full model usage may be reduced, although it affected the plan quality in our initial experiments.

Figures 2(a), 2(b), and 2(c) show the average runtime savings (%) of the techniques in each domain. The *higher values* indicate improved runtime gains by using the model. Note that these estimates include the time taken for re-planning. The runtime of ACARM-0/1 RM is at least 20% faster than solving the original problem, except in EV RF-1 and RF-2. In some problem instances, the runtime of ACARM-0/1 RM is comparable to that of MLOD and FLARES. This is primarily due to better solution quality that requires fewer re-planning. Again, the objective of our approach is not to im-

Problem	% Full Model
EV-RF-1	2.685
EV-RF-2	2.750
EV-RF-3	3.764
EV-RF-4	3.505
Racetrack-Square-4	0.071
Racetrack-Square-5	0.034
Racetrack-Ring-5	1.859
Racetrack-Ring-6	0.327
Sailing-20(C)	37.414
Sailing-40(C)	37.478
Sailing-80(C)	37.495
Sailing-20(M)	37.414
Sailing-40(M)	37.478
Sailing-80(M)	37.495

Table 1: % Full model usage in 0/1 RM using cost adjustment as heuristic for model selector.

prove runtime, but to improve the solution quality without compromising the runtime gains of using a reduced model. Our results indicate that ACARM-0/1 RM with a good model selector and cost estimation can achieve near-optimal performance without significantly affecting the planning time.

We solve 0/1 RM and ACARM-0/1 RM using an optimal algorithm, LAO*, to demonstrate the effectiveness of our framework by comparing the optimal solutions of the models. Since the 0/1 RM and ACARM-0/1 RM are still SSPs, in practice, any SSP solver (optimal or not) may be used. In our experiments, we use a simple model selector that is intuitive, and uses the cost adjustments as heuristic. We use the full model in states with the highest cost adjustments in each domain, since it denotes states which could significantly affect the expected cost of reaching the goal. Automating the model selector would benefit the approach, and this requires faster techniques to identify and evaluate relevant outcome selection principles for the domain, which are currently open challenges. The aim of this paper is to demonstrate the potential of out frameworks in improving solution quality and to identify important open questions.

7 Related Work

The Stochastic Shortest Path (SSP) [Bertsekas and Tsitsiklis, 1991] is a widely-used model for sequential decision making in stochastic environments, for which numerous planning algorithms have been developed. Among the different reduced model techniques for SSPs, determinization has attracted significant interest because it greatly simplifies the problem and can solve large MDPs much faster. Interest in determinization increased after the success of *FF-Replan* [Yoon *et al.*, 2007], which won the 2004 IPPC using the *Fast Forward* (FF) technique to generate deterministic plans [Hoffmann, 2001]. Following the success of FF-Replan, researchers have proposed various methods to improve determinization [Kolobov *et al.*, 2009; Yoon *et al.*, 2010; Keller and Eyerich, 2011; Keyder and Geffner, 2008; Issakkimuthu *et al.*, 2015]. However, determinization-based algorithms may produce plans that are arbitrarily worse than the optimal plan because they consider each outcome of an action in isolation. The M_l^k reduced model generalizes the single outcome determinization by considering a set of primary outcomes (l) and a set of exceptions (k) per action that are fully accounted for in the reduced model [Pineda and Zilberstein, 2014]. It has shown to accelerate planning time considerably compared to solving the problem optimally, while improving solution quality compared to determinization. However, it is hard to identify a priori which M_l^k reduction is best for a problem.

Despite the success of existing reduced model techniques in improving the runtime, they cannot be applied to many large real-world problems in which ignoring probabilistic outcomes can introduce considerable risks, such as wildfire response and semi-autonomous driving [Hajian *et al.*, 2016; Wray *et al.*, 2016]. A major drawback of the existing techniques is the lack of a mechanism to identify risky outcomes in the original problem and account for them in the reduced model, which is required to produce high-quality plans.

A further benefit of our approach is the use of a portfolio of reduced models that offers additional flexibility. Since model fidelity affects both runtime and solution quality, it makes it possible to design *contract anytime algorithms* [Zilberstein, 1996] for SSPs, which allow solution quality to degrade gracefully with runtime. The approach could therefore provide multiple methods for solving a given SSP that can be used within the *progressive processing* framework [Mouaddib and Zilberstein, 1998].

The importance of accounting for risks in AI systems is attracting growing attention [Cserna *et al.*, 2018; Zilberstein, 2015; Kulić and Croft, 2005]. However, safety in reduced model formulations has not been explored. We propose techniques to formulate a safe reduced model for stochastic planning. We achieve this by switching between different outcome selection principles and adjusting the costs of actions.

8 Conclusion and Future Work

Reduced models have become a popular approach to quickly solve large SSPs. However, the existing techniques are oblivious to the risky outcomes in the original problem when formulating a reduced model. We propose two general methods that help create safe reduced models of large SSPs. First, we propose planning using a portfolio of reduced models that provides flexibility in outcome selection. Secondly, we introduce reduced models with cost adjustments, a technique that accounts for ignored outcomes in the reduced model. Since computing the exact cost adjustment requires the optimal values of the states, we propose approximate techniques for cost estimation and also provide conditions under which state independent costs result in optimal action selection. We then describe how cost adjustment can be used as a heuristic for model selector in PRMs. Our empirical results demonstrate the promise of this framework as cost adjustments in a basic instantiation of a PRM offer improvements; ACARM-0/1 RM yields near-optimal solutions in most problem instances. Our results contribute to a better understanding of how disparate reduce model techniques relate to each other and could be used together to leverage their complementary benefits.

The 0/1 RM represents an initial exploration of a broad spectrum of PRMs. There are a number of improvements that could add value to our approach. First, we aim to devise online learning mechanisms for the cost estimation to avoid the preprocessing phase. Secondly, we aim to identify other notions of safety in reduced models. Finally, we are working on practical methods for automatically devising good model selectors beyond the cost adjustment heuristic. This involves developing improved metrics and techniques for evaluating outcome selection principles.

Acknowledgments

We thank Prashant Shenoy for his help formulating and accessing data for the EV charging problem. We thank Luis Pineda and Kyle Wray for the helpful discussions and providing us with the code for FLARES algorithm. Support for this work was provided in part by the National Science Foundation under grant IIS-1524797.

References

[Barto *et al.*, 1995] Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138, 1995.

[Bertsekas and Tsitsiklis, 1991] Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16:580–595, 1991.

[Bonet and Geffner, 2003] Blai Bonet and Hector Geffner. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *International Conference on Automated Planning and Scheduling*, 2003.

[Boutilier *et al.*, 1999] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:94, 1999.

[Cserna *et al.*, 2018] Bence Cserna, William J. Doyle, Jordan S. Ramsdell, and Wheeler Ruml. Avoiding dead ends in real-time heuristic search. In *32nd Conference on Artificial Intelligence*, 2018.

[Eversource, 2017] Eversource. Eversource energy - time of use rates. https://www.eversource.com/clp/vpp/vpp.aspx, 2017.

[Hajian *et al.*, 2016] Mohammad Hajian, Emanuel Melachrinoudis, and Peter Kubat. Modeling wildfire propagation with the stochastic shortest path: A fast simulation approach. *Journal of Environmental Modelling & Software*, 82:73–88, 2016.

[Hansen and Zilberstein, 2001] Eric A. Hansen and Shlomo Zilberstein. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129:35–62, 2001.

[Hart *et al.*, 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.

[Hoffmann, 2001] Jörg Hoffmann. FF: The fast-forward planning system. *AI Magazine*, 22:57, 2001.

[Issakkimuthu *et al.*, 2015] Murugeswari Issakkimuthu, Alan Fern, Roni Khardon, Prasad Tadepalli, and Shan Xue. Hindsight optimization for probabilistic planning with factored actions. In *25th International Conference on Automated Planning and Scheduling*, 2015.

[Keller and Eyerich, 2011] Thomas Keller and Patrick Eyerich. A polynomial all outcome determinization for probabilistic planning. In *21st International Conference on Automated Planning and Scheduling*, 2011.

[Keyder and Geffner, 2008] Emil Keyder and Hector Geffner. The HMDP planner for planning with probabilities. In *International Planning Competition (IPC 2008)*, 2008.

[Kocsis and Szepesvári, 2006] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, volume 6, pages 282–293. Springer, 2006.

[Kolobov *et al.*, 2009] Andrey Kolobov, Mausam, and Daniel S Weld. ReTrASE: Integrating paradigms for approximate probabilistic planning. In *21st International Joint Conference on Artificial Intelligence*, 2009.

[Korf, 1990] Richard E. Korf. Real-time heuristic search. *Artificial intelligence*, 42(2-3):189–211, 1990.

[Kulić and Croft, 2005] Dana Kulić and Elizabeth A Croft. Safe planning for human-robot interaction. *Journal of Field Robotics*, 22(7):383–396, 2005.

[Little and Thiebaux, 2007] Iain Little and Sylvie Thiebaux. Probabilistic planning vs. replanning. In *ICAPS Workshop on the International Planning Competition: Past, Present and Future*, 2007.

[Littman, 1997] Michael L. Littman. Probabilistic propositional planning: Representations and complexity. In *14th International Conference on Artificial Intelligence*, 1997.

[Mahadevan, 2009] Sridhar Mahadevan. Learning representation and control in markov decision processes: New frontiers. *Foundations and Trends in Machine Learning*, 1:403–565, 2009.

[Mouaddib and Zilberstein, 1998] Abdel-Illah Mouaddib and Shlomo Zilberstein. Optimal scheduling of dynamic progressive processing. In *13th European Conference on Artificial Intelligence*, 1998.

[Parr *et al.*, 2007] Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing feature generation for value-function approximation. In *24th International Conference on Machine learning*, 2007.

[Petrik and Zilberstein, 2006] Marek Petrik and Shlomo Zilberstein. Learning parallel portfolios of algorithms. *Annals of Mathematics and Artificial Intelligence*, 48(1-2):85–106, 2006.

[Pineda and Zilberstein, 2014] Luis Pineda and Shlomo Zilberstein. Planning under uncertainty using reduced models: Revisiting determinization. In *24th International Conference on Automated Planning and Scheduling*, 2014.

[Pineda *et al.*, 2017] Luis Pineda, Kyle Wray, and Shlomo Zilberstein. Fast SSP solvers using short-sighted labeling. In *31st International Conference on Artificial Intelligence*, 2017.

[Saisubramanian *et al.*, 2017] Sandhya Saisubramanian, Shlomo Zilberstein, and Prashant Shenoy. Optimizing electric vehicle charging through determinization. In *ICAPS Workshop on Scheduling and Planning Applications*, 2017.

[Shah *et al.*, 2012] Mohak Shah, Mario Marchand, and Jacques Corbeil. Feature selection with conjunctions of decision stumps and learning from microarray data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:174–186, 2012.

[Styler and Simmons, 2017] Breelyn Styler and Reid Simmons. Plan-time multi-model switching for motion planning. In *27th International Conference on Automated Planning and Scheduling*, 2017.

[Wray *et al.*, 2016] Kyle Hollins Wray, Luis Pineda, and Shlomo Zilberstein. Hierarchical approach to transfer of control in semi-autonomous systems. In *25th International Joint Conference on Artificial Intelligence*, 2016.

[Yoon *et al.*, 2007] Sungwook Yoon, Alan Fern, and Robert Givan. FF-Replan: A baseline for probabilistic planning. In *17th International Conference on Automated Planning and Scheduling*, 2007.

[Yoon *et al.*, 2008] Sungwook Yoon, Alan Fern, Robert Givan, and Subbarao Kambhampati. Probabilistic planning via determinization in hindsight. In *23rd Conference on Artificial Intelligence*, 2008.

[Yoon *et al.*, 2010] Sungwook Yoon, Wheeler Ruml, J. Benton, and Minh B. Do. Improving determinization in hindsight for on-line probabilistic planning. In *20th International Conference on Automated Planning and Scheduling*, 2010.

[Younes and Littman, 2004] Hakan L. S. Younes and Michael L. Littman. PPDDL1.0: The language for the probabilistic part of IPC-4. In *International Planning Competition*, 2004.

[Zilberstein, 1996] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.

[Zilberstein, 2015] Shlomo Zilberstein. Building strong semi-autonomous systems. In *29th Conference on Artificial Intelligence*, 2015.