Minimizing Conditional Cost with Goal-Reachability Guarantee in Stochastic Shortest Paths with Dead-Ends

Matisse Roche[1,2,3], **Yoko Watanabe**[1,2], **Caroline P. C. Chanel**[1,3]
[1]Fédération ONERA ISAE-SUPAERO ENAC, Université de Toulouse, France
[2]ONERA, Toulouse, France
[3]ISAE-SUPAERO, Université de Toulouse, France
matisse.roche@onera.fr

Abstract

In several stochastic shortest path (SSP) planning domains, a planner must find a strategy to reach a goal in an uncertain environment in which some states are irrecoverable dead-ends. When dead-ends cannot always be avoided, the planner may face a tension between being safe and being efficient. Existing policy optimization criteria do not let the human operator control this trade-off directly: S3P imposes a strict priority on success probability before considering path cost, while fSSP relies on a failure penalty parameter whose tuning offers implicit control over the resulting trade-off. We introduce the Safety-constrained Conditional-Cost SSP (SCC-SSP), where the human operator chooses a minimum acceptable success probability and the planner minimizes the expected cost conditioned on reaching the goal. The ratio structure of this objective introduces a coupling between path costs and future goal reachability that places SCC-SSP outside the scope of standard Constrained Markov Decision Processes (C-MDPs). To resolve this coupling, we build an augmented MDP that tracks cumulative cost in the state, and prove that the optimal policy depends not only on the current state but also on the cost accumulated until the current state, with the optimum achieved by a mixture of at most two deterministic cost-dependent policies. This results in a class of policies strictly richer than stationary policies. To compute such policies, our algorithm combines Dinkelbach's method for fractional objectives with a Lagrangian relaxation of the safety constraint. Experiments on classical benchmarks indicate that SCC-SSP can trace the full Pareto front between success probability and conditional cost, whereas C-MDP produces operating points governed by the penalty parameter rather than the safety threshold.

1 Introduction

Goal-oriented Markov Decision Processes (Goal-Oriented MDPs) serve as the fundamental framework for sequential decision-making in stochastic domains. This formalism powers critical applications ranging from autonomous navigation to industrial logistics. Classically, these problems are modeled as Stochastic Shortest Path (SSP) problems (Bertsekas and Tsitsiklis 1991). Standard SSP algorithms rely on a foundational assumption: the existence of at least one proper policy : a policy guaranteed to reach the goal state with probability 1 from any starting configuration. However, this assumption is often unrealistic in practical applications (Kolobov, Mausam, and Weld 2012). Indeed, many environments feature dead-end states, from which the goal becomes strictly unreachable. Consider a fleet of firefighting drones dispatched to contain a wildfire: while each drone must navigate through turbulent winds and dynamic obstacles, a collision results in a catastrophic crash from which recovery is impossible. Similarly, in high-stakes manufacturing, if a robotic arm breaks a unique component, the production goal becomes unattainable. In such environments, dead-ends may be unavoidable: any policy can carry a non-zero probability of failure. Consequently, the standard SSP objective of minimizing the expected total cost becomes ill-posed, as the expected cost of any policy diverges to infinity.

Existing approaches handle this by modifying the MDP to render the expected cost finite. A common device is to introduce an artificial "give-up" action from dead-end states to the goal, effectively terminating failed trajectories. This makes the unconditional expected cost well-defined and amenable to standard optimization.

However, within this modified model, managing the safety-efficiency trade-off remains problematic. Assigning a finite penalty cost to the give-up action and minimizing the resulting unconditional expected cost (Kolobov, Mausam, and Weld 2012) can create a structural failure-seeking bias: as illustrated in Figure 1, the solver exploits cheap failures to deflate the global expectation, even when the cost of successful trajectories is strictly higher. Chance-Constrained SSPs (Hong et al. 2021) subject this minimization to a failure probability constraint, which limits this effect but does not eliminate it, since the objective itself still rewards failure. The second paradigm, represented by S3P (Teichteil-Königsbuch 2012) or i-dual with high penalties (Trevizan et al. 2017), implicitly assigns an infinite cost to failure, thereby strictly prioritizing safety. While theoretically robust, this lexicographic ordering prevents any trade-off: as shown by (Freire and Delgado 2017), such solvers often discard highly efficient policies for negligible safety gains.

A different perspective arises from noting that, in many safety-critical applications, the cost accumulated before a failure carries no operational value. If a firefighting drone crashes after 10 seconds of flight, those 10 seconds contribute nothing to extinguishing the fire: the operator only

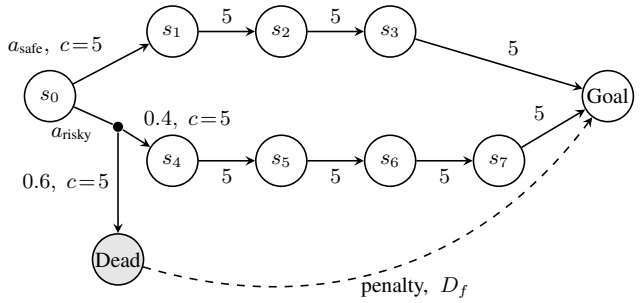

Figure 1: Counter-example. Let π_{safe} and π_{risky} be the policies that select a_{safe} and a_{risky} at s_0, respectively. The policy π_{safe} reaches the goal with certainty and a conditional cost of 20. Yet minimizing the unconditional expected cost prefers π_{risky}, which is both riskier (goal reached with probability 0.4) and more expensive on successful trajectories (conditional cost 25). The failure outcome occurs early and is therefore cheap, which drags down its global expectation: $0.4 \times 25 + 0.6 \times (5 + D_f) = 19 < 20$ for $D_f = 10$.

cares how fast successful drones arrive and how often they fail. This suggests treating safety and efficiency as separate concerns: enforcing a lower bound on the success probability, while measuring efficiency only over successful trajectories. This conditional formulation is moreover inherently well-posed, as every goal-reaching trajectory has finite cost. In this paper, we formalize this idea as the Safety-constrained Conditional-Cost SSP (SCC-SSP). We make four contributions:

- We introduce SCC-SSP, which minimizes the expected cost conditioned on success subject to a user-chosen safety threshold, and show that it generalizes both S3P and standard SSPs (Section 3).
- We prove that optimal SCC-SSP policies are cost-dependent, not stationary. Under an assumption of integer costs and bounded budget, we prove that a mixture of at most two deterministic cost-dependent policies suffices (Sections 3 and 4).
- Under the same assumption, we develop an exact algorithm combining Dinkelbach's transform, Lagrangian relaxation, and backward induction on an augmented MDP (Section 4).
- We show experimentally that SCC-SSP traces the full Pareto front between safety and efficiency, while C-MDP produces isolated operating points and S3P returns the most conservative policy (Section 5).

2 Background and Related Work

Definitions

Markov Decision Processes (MDPs) (Bertsekas and Tsitsiklis 1991; Mausam and Kolobov 2012) provide a fundamental framework for sequential decision-making. A goal-oriented MDP is defined as a tuple $\langle S, A, T, c, \mathcal{G}\rangle$ where S is a finite set of states, A is a finite set of actions, and $T : S \times A \times S \to [0, 1]$ is the transition function. The cost function $c : S \times A \times S \to \mathbb{R}$ assigns a strictly positive cost to every transition originating from non-goal states. $\mathcal{G} \subseteq S$ is the set of goal states, which are cost-free and absorbing ($T(g, a, g) = 1$ and $c(g, a, g) = 0$ for all $g \in \mathcal{G}, a \in A$).

A *trajectory* $\theta = (s_0, a_0, s_1, a_1, \ldots)$ is a sequence of states and actions. We define the cumulative cost of a trajectory as $C(\theta) = \sum_{t=0}^{\infty} c(s_t, a_t, s_{t+1})$ and denote by $\Theta_{\mathcal{G}}^{\pi,s}$ the set of all trajectories generated by policy π starting from s that eventually reach a goal state $g \in \mathcal{G}$.

A *history-dependent* policy maps each history $h_t = (s_0, a_0, \ldots, s_t)$ to a distribution over actions, $\pi_t(a \mid h_t) \in \Delta(A)$. This is the most general class. A policy is *Markovian* if it depends on the history only through the current state, $\pi_t(a \mid s_t)$, and *stationary* if in addition the decision rule does not depend on the time step, yielding a single mapping $\pi(a \mid s) \in \Delta(A)$ applied at every step. Throughout, "stationary" is understood in this sense: Markovian and time-homogeneous. A stationary policy is *deterministic* if $\pi(\cdot \mid s)$ is a Dirac distribution for every state. A *mixture of deterministic policies* is defined by a finite set $\{\pi_1, \ldots, \pi_k\}$ of deterministic policies together with weights $z_1, \ldots, z_k \geq 0$ summing to one. At the beginning of execution, a single policy π_i is drawn with probability z_i and followed for the entire trajectory. This class is strictly richer than deterministic policies but does not require per-step randomization: the only source of stochasticity is the initial draw.

The probability of success of a policy π from state s is:

$$P_{\mathcal{G}}^{\pi}(s) = \mathbb{P}(\theta \in \Theta_{\mathcal{G}}^{\pi,s}) \tag{1}$$

A state $s \in S$ is a *dead-end* if the goal cannot be reached from s under any policy, i.e., $\max_{\pi} P_{\mathcal{G}}^{\pi}(s) = 0$. We denote by $\mathcal{D} \subseteq S$ the set of all dead-end states.

Finally, the conditional expected cost is the expectation of $C(\theta)$ over the set of successful trajectories:

$$E_c^{\pi}(s) = \mathbb{E}[C(\theta) \mid \theta \in \Theta_{\mathcal{G}}^{\pi,s}] \tag{2}$$

Related Work

Safety-Maximizing Approaches. To handle unavoidable dead-ends, a prominent family of approaches strictly prioritizes goal reachability. iSSPUDE (Kolobov, Mausam, and Weld 2012) assigns an infinite penalty to dead-end states, forcing the agent to avoid them whenever possible. Under strictly positive action costs, its optimal policies coincide with those of the **S3P** (Stochastic Safest and Shortest Path) framework (Teichteil-Königsbuch 2012), which lexicographically optimizes the probability of success $P_{\mathcal{G}}^{\pi}(s_0)$ first, and the conditional expected cost $E_c^{\pi}(s_0)$ second:

$$\pi^* \in \arg\min_{\pi \in \Pi_{safe}} E_c^{\pi}(s_0), \text{ where } \Pi_{safe} = \arg\max_{\pi \in \Pi} P_{\mathcal{G}}^{\pi}(s_0). \tag{3}$$

Min-Cost given Max-Prob (MCMP) (Trevizan and Teichteil-Königsbuch 2017) shares this primary objective but minimizes the expected cost over all trajectories, including the cost incurred before entering a dead-end, whereas S3P considers only successful ones. This rigid prioritization ensures maximum safety but frequently yields inefficient policies, unable to balance risk and cost.

Threshold-Based Approaches. Instead of absolute maximization, other approaches seek to satisfy a user-defined safety requirement. The **AtLeastProb** framework (Steinmetz, Hoffmann, and Buffet 2016) focuses on the feasibility problem: it verifies whether there exists a policy π such that $P^\pi_\mathcal{G}(s_0) \geq P_{th}$. However, it terminates as soon as the threshold is met; it does not optimize a cost metric, and gives no guidance on which policy to choose among the safe ones.

Chance-Constrained SSP. The Chance-Constrained SSP framework (Hong et al. 2021) lets users define a set of undesirable states $\mathcal{S}_U$ and constrains the probability of entering them. Formally, the problem minimizes the expected cost over a finite horizon H:

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}^\pi \left[\sum_{t=0}^{H-1} c(s_t, a_t, s_{t+1}) \right] \\ \text{s.t.} \quad & \mathbb{P}^\pi(\exists\, t \leq H,\ s_t \in \mathcal{S}_U) \leq \Delta \end{aligned} \tag{4}$$

Unlike approaches that isolate successful missions, this formulation minimizes the total expected cost over all trajectories, with cost accumulation ceasing once an agent enters an undesirable state. To solve it, Hong et al. (2021) propose an anytime heuristic search that quickly produces a conservative feasible solution and then iteratively improves its efficiency.

Utility-Based Approaches. Freire and Delgado (2017) proposed GUBS, a criterion grounded in expected utility theory that evaluates policies through a non-linear aggregation of cost and goal attainment. A risk parameter controls how aggressively the agent trades safety for efficiency, but failed trajectories still contribute to the objective. As in our setting, optimal GUBS policies are cost-dependent rather than stationary, and require an augmented state space (Freire, Delgado, and Reis 2019; Crispino, Freire, and Delgado 2023).

Constrained SSPs (C-SSPs) and i-dual. The C-SSP framework (Altman 1999) generalizes standard SSPs to multi-objective settings. It seeks a policy π that minimizes a primary expected cost C_0 subject to upper bounds $\hat{c}_j$ on K secondary expected costs $C_j, j \in \{1...K\}$:

$$\min_{\pi} \mathbb{E}^\pi[C_0(\theta)] \quad \text{s.t.} \quad \mathbb{E}^\pi[C_j(\theta)] \leq \hat{c}_j, \quad \forall j \in \{1...K\}$$

This is typically solved by transforming it into a Linear Program (LP) over the convex set of occupation measures (the expected number of visits to each state-action pair). In contrast to standard SSPs, whose optimal policy is deterministic, optimal solutions to this constrained formulation are generally stochastic (Chatterjee, Majumdar, and Henzinger 2006). To solve it efficiently, Trevizan et al. (2017) introduced **i-dual**, which uses heuristic search in the dual space and avoids enumerating all states, offering significant speedups over standard LP solvers. To remain applicable with dead-ends, i-dual relies on a penalty-based modification: an artificial action allows transitions from dead-ends to the goal at a high penalty, which numerically keeps the primary expected cost finite.

3 Safety-constrained Conditional-Cost SSP Problem

In this section, the Safety-constrained Conditional-Cost Stochastic Shortest Path problem (SCC-SSP) is introduced generalizing existing frameworks.

The Safety-constrained Conditional SSP Problem

For a given probability threshold $P_{th} \in (0, 1]$, we define the Safety-constrained Conditional-Cost Stochastic Shortest Path problem (SCC-SSP) as:

$$\min_{\pi} \ E^\pi_c(s_0) \quad \text{subject to} \quad P^\pi_\mathcal{G}(s_0) \geq P_{th} \tag{5}$$

This formulation isolates operational efficiency from failure scenarios and offers a flexible safety-efficiency trade-off controlled by P_{th}. Furthermore, it strictly generalizes two well-known frameworks:

- **Generalization of S3P:** Setting $P_{th} = P^*(s_0)$ (the maximum achievable success probability) recovers the secondary objective of the S3P framework (Teichteil-Königsbuch 2012), which minimizes the conditional cost within the set of safest policies.
- **Generalization of SSP:** In environments without dead-ends, $P^*(s_0) = 1$. Setting $P_{th} = 1$ makes the conditional cost equal to the unconditional cost, reducing the problem to a standard SSP.

Why Standard C-MDP Algorithms Fall Short

At first glance, problem (5) might appear solvable within the Constrained MDP (C-MDP) framework (Altman 1999): one would minimize an expected cost subject to the linear constraint $P^\pi_\mathcal{G}(s_0) \geq P_{th}$. However, there are two fundamental obstacles.

The objective is non-linear. Standard C-MDP algorithms require the objective to be linear in the state-action occupation measures. In our formulation, the conditional expected cost can be rewritten as:

$$\mathbb{E}^\pi_c(s_0) = \frac{\mathbb{E}^\pi[C \cdot \mathbf{1}_{\text{succ}} \mid s_0]}{P^\pi_\mathcal{G}(s_0)} \tag{6}$$

This fractional structure places the problem outside the scope of standard linear programming methods used to solve C-MDPs. To the best of our knowledge, no efficient algorithm exists in the literature to solve problem (5) exactly for goal-oriented MDPs with dead-ends.

Enforcing the constraint does not prevent the penalty bias. To work within the C-MDP framework, one must replace the conditional objective by an unconditional expected cost, which is linear. However, as discussed in the introduction, the unconditional cost is ill-defined when dead-ends are unavoidable. The standard workaround is to introduce an artificial action from dead-end states to the goal at a finite penalty cost D_f, yielding a valid C-MDP solvable by algorithms such as i-dual (Trevizan et al. 2017). The safety constraint $P^\pi_\mathcal{G}(s_0) \geq P_{th}$ remains useful in this reformulation: it guarantees that the returned policy achieves the requested

success probability. However, it does not prevent the failure-seeking bias. If D_f is zero or small, the solver can exploit cheap failures to deflate the global expectation, as long as the constraint is satisfied, as shown in Section 5 and Figure 1. A natural remedy is to increase D_f to penalize failures more heavily. But if D_f is too large, the penalty dominates the unconditional cost and the solver maximizes safety regardless of P_{th}, making the constraint redundant. In summary, the C-MDP formulation with a penalty can enforce the constraint, but cannot optimize the conditional cost: the trade-off nature remains governed by the arbitrary choice of D_f.

Structure of the Optimal Policy

The ratio structure of the conditional cost (6) has direct consequences on the class of policies needed to solve SCC-SSP. We show that the optimal action at a state may depend on the cost already accumulated, and that the accumulated cost is the *only* piece of past information that matters.

Definition 1 (Cost-dependent policy)**.** *A* cost-dependent policy *is a mapping* $\pi\colon S \times \mathbb{R}_+ \to \Delta(A)$*, where the action at time* t *depends on the current state* s_t *and the cumulative cost* $c_t = \sum_{k=0}^{t-1} c(s_k, a_k, s_{k+1})$.

Proposition 1 (Sufficiency)**.** *For any history-dependent policy* π_H*, there exists a cost-dependent policy* π_{CD} *with the same success probability and conditional expected cost.*

Proof. Write $h_t = (s_0, a_0, \ldots, s_t)$ for the history at time t and $c_t = \sum_{k=0}^{t-1} c(s_k, a_k, s_{k+1})$ for the accumulated cost. Consider a non-terminal state $s_t \notin \mathcal{G} \cup \mathcal{D}$. Since $C(\theta) = c_t + C(\theta_{\text{fut}})$ and $\mathbf{1}_{\text{succ}}$ then depends only on the future,

$$\begin{aligned}\mathbb{E}^{\pi_H}\big[C \cdot \mathbf{1}_{\text{succ}} \mid h_t\big] &= c_t \cdot P_{\mathcal{G}}^{\pi_H}(s_t, h_t) \\ &\quad + \mathbb{E}^{\pi_H}\big[C(\theta_{\text{fut}}) \cdot \mathbf{1}_{\text{succ}} \mid h_t\big].\end{aligned} \tag{7}$$

(If $s_t \in \mathcal{G} \cup \mathcal{D}$, the trajectory has already terminated and $\mathbf{1}_{\text{succ}}$ is determined by s_t alone.) The transitions depend only on (s_t, a_t), so the right-hand side depends on the past only through (s_t, c_t): the accumulated cost is the only piece of history that matters.

Define π_{CD} by averaging π_H over all histories consistent with a given (s, c) pair:

$$\pi_{CD}(a \mid s, c) = \mathbb{E}^{\pi_H}\Big[\pi_H(a \mid h_t) \;\Big|\; s_t = s,\ c_t = c\Big]. \tag{8}$$

We show by induction that the marginal of (s_t, c_t) is the same under both policies. At $t = 0$, both start at $(s_0, 0)$. If the marginals agree at t, then (8) gives the same action distribution given (s_t, c_t), and since transitions depend only on (s_t, a_t), agreement extends to $t+1$. The distribution over trajectories is thus preserved, so $P_{\mathcal{G}}^{\pi_{CD}}(s_0) = P_{\mathcal{G}}^{\pi_H}(s_0)$ and $\mathbb{E}^{\pi_{CD}}[C \cdot \mathbf{1}_{\text{succ}}] = \mathbb{E}^{\pi_H}[C \cdot \mathbf{1}_{\text{succ}}]$. □

Proposition 2 (Necessity)**.** *There exist SCC-SSP instances where no stationary policy, whether deterministic, stochastic, or any mixture of deterministic stationary policies, achieves the optimal conditional cost.*

Proof sketch. Consider the MDP $\{s_0, g, d\}$ where s_0 has two actions: a_1 loops back to s_0 with probability $1 - p_1$ and reaches g with probability p_1 (cost c_1); a_2 reaches g with probability p_2 or d with probability $1 - p_2$ (cost c_2), with $p_1 < p_2$ and $c_1 > c_2$. A cost-dependent policy can play a_1 for exactly k rounds, attempting the goal without any risk of hitting the dead-end, and then switch to a_2. A stationary policy must use the same distribution at every visit of s_0, so it cannot replicate this switch. For $(p_1, p_2, c_1, c_2) = (0.1, 0.6, 5, 1)$, direct computation confirms that the cost-dependent policy strictly dominates every stationary policy for a given threshold. □

Propositions 1 and 2 together characterize the search space for SCC-SSP: cost-dependent policies are sufficient, and stationary policies are not. In the classical SSP setting, an optimal deterministic stationary policy is guaranteed to exist (Bertsekas and Tsitsiklis 1991); in Constrained MDPs, optimality is achieved within the class of stationary (possibly randomized) policies (Altman 1999). The ratio structure of the conditional cost breaks both results and forces us to work with a strictly richer policy class. How to compute an optimal policy in this larger space is the subject of the next section.

4 Policy Optimization

Propositions 1 and 2 established that the optimal SCC-SSP policy is cost-dependent. In general, the space of such policies is infinite-dimensional, since the accumulated cost is real-valued. Under an assumption of integer costs and bounded budget (Assumption 1), we construct an augmented MDP that makes this space finite and reformulate SCC-SSP as a fractional program over a polytope (Section 4.1). We then solve this program through three nested layers: Dinkelbach's method converts the ratio objective into a sequence of linear subproblems (Section 4.2), Lagrangian relaxation handles the safety constraint in each subproblem (Section 4.3), and backward induction on the augmented MDP solves the resulting unconstrained minimization (Section 4.4).

Augmented MDP and Problem Reformulation

Assumption 1 (Integer costs and finite budget)**.** *All transition costs are positive integers. There exists a known constant* $B \in \mathbb{N}$ *such that every trajectory of interest reaches* $\mathcal{G} \cup \mathcal{D}$ *before accumulating a total cost exceeding* B.

The first condition is standard in planning domains where costs represent discrete quantities such as time steps or fuel units. The budget B models a natural operational constraint: a battery capacity, a mission deadline, or any hard limit on the resources available before the mission must end. It is a parameter of the problem, chosen by the operator.

Definition 2 (Augmented MDP M_B^+)**.** *The augmented MDP* M_B^+ *has state space* $\hat{S} = \{(s, r) : s \in S \setminus (\mathcal{G} \cup \mathcal{D}),\ r \in \{0, \ldots, B\}\} \cup \{\hat{g}, \hat{d}\}$*, where* $\hat{g}$ *and* $\hat{d}$ *are absorbing terminal states. From state* (s, r) *under action* a*, the successor is* $(s', r + c(s, a, s'))$ *if* $s' \notin \mathcal{G} \cup \mathcal{D}$ *and* $r + c(s, a, s') \le B$*;* $\hat{g}$ *if* $s' \in \mathcal{G}$ *and* $r + c(s, a, s') \le B$*; and* $\hat{d}$ *otherwise (i.e.,* $s' \in \mathcal{D}$ *or the budget is exceeded).*

Since costs are positive integers, the cost component increases strictly at every transition: any transition from (s, r) leads to a state (s', r') with $r' > r$, or to a terminal state. No cycle is possible. The augmented MDP M_B^+ is therefore a directed acyclic graph (DAG) with at most $|S \setminus (\mathcal{G} \cup \mathcal{D})| \cdot (B+1) + 2$ states.

From cost-dependent policies to stationary policies in M_B^+. By construction, the augmented state (s, r) encodes both the current state s and the accumulated cost r. A cost-dependent policy in M, which selects a distribution over actions based on the pair (s, r), corresponds to a stationary policy in M_B^+: one that maps each augmented state to a distribution over actions. Proposition 1 showed that cost-dependent policies are sufficient for SCC-SSP. We can therefore restrict the search to stationary policies in M_B^+ without loss of optimality.

Stochastic policies reduce to mixtures of deterministic ones. Because M_B^+ is a DAG, every state is visited at most once along any trajectory. A stochastic policy π therefore induces a distribution over trajectories in which each randomization occurs at a distinct state; every realized trajectory coincides with the trajectory of some deterministic policy. The distribution over trajectories under π is thus a mixture over deterministic policies. Since $V^\pi(s_0)$ and $P_\mathcal{G}^\pi(s_0)$ are both expectations over trajectories, they decompose linearly over this mixture, so the performance pair $(P_\mathcal{G}^\pi(s_0), V^\pi(s_0))$ lies in the convex hull of the deterministic pairs. This argument relies on the DAG structure: in an MDP with cycles, a single trajectory may visit the same state repeatedly, and local randomizations no longer map to a single deterministic policy.

Reformulation as optimization over a polytope. Let $\Pi_{\text{det}} = \{\pi_1, \dots, \pi_N\}$ denote the finite set of all the deterministic stationary policies in M_B^+. For each π_i, write $P_i = P_\mathcal{G}^{\pi_i}(s_0)$ for its success probability and $V_i = \mathbb{E}^{\pi_i}[C(\theta) \cdot \mathbf{1}_{\text{succ}} \mid s_0]$ for its success-weighted cost, where in general $V^\pi(s_0) = \mathbb{E}^\pi[C(\theta) \cdot \mathbf{1}_{\text{succ}} \mid s_0]$ denotes the success-weighted cost of a policy π. Let $\mathcal{F} = \{(P_i, V_i)\}_{i=1}^N$ and let $\text{conv}(\mathcal{F})$ denote its convex hull. By the decomposition above, the set of all achievable performance pairs over stationary policies in M_B^+ (stochastic included) is exactly $\text{conv}(\mathcal{F})$.

A mixture policy draws π_i with probability z_i at the start and follows it until termination. By the law of total expectation:

$$P_\mathcal{G}^z(s_0) = \sum_{i=1}^N z_i P_i, \qquad V^z(s_0) = \sum_{i=1}^N z_i V_i. \tag{9}$$

The SCC-SSP problem in M_B^+ becomes:

$$\lambda^* = \min_{z \in \Delta_N} \frac{\sum_i z_i V_i}{\sum_i z_i P_i} \qquad \text{subject to} \qquad \sum_i z_i P_i \geq P_{th} \tag{10}$$

where $\Delta_N = \{z \in \mathbb{R}^N : z_i \geq 0,\ \sum_i z_i = 1\}$ is the probability simplex. This is a fractional program over a polytope. Since only the two quantities $P^z = \sum z_i P_i$ and $V^z = \sum z_i V_i$ matter, the geometry of the problem lives in $\mathbb{R}^2$.

Dinkelbach's Transform

The optimization problem (10) minimizes a ratio, which is not directly amenable to linear programming. Dinkelbach's method (Dinkelbach 1967) converts this fractional program into a sequence of linear subproblems.

Instead of minimizing V^z/P^z directly, we introduce a parameter $\lambda \geq 0$ and define:

$$F(\lambda) = \min_{z \in Z_f} \sum_{i=1}^N z_i (V_i - \lambda P_i) \tag{11}$$

where $Z_f = \{z \in \Delta_N : \sum_i z_i P_i \geq P_{th}\}$ is the set of feasible mixtures. The function F has three key properties. First, $F(\lambda^*) = 0$: if z^* solves SCC-SSP with $\lambda^* = V^{z^*}/P^{z^*}$, then $\sum_i z_i^* (V_i - \lambda^* P_i) = 0$, and for any other feasible z we have $\sum_i z_i (V_i - \lambda^* P_i) \geq 0$. Second, F is strictly decreasing in λ. Third, F has a unique zero. These properties are established in the supplementary material.

These properties ground Dinkelbach's iteration: starting from $\lambda \geq \lambda^*$, we evaluate $F(\lambda)$, read off the optimal mixture z, and update $\lambda \leftarrow V^z/P^z$. The sequence of iterates decreases monotonically and converges superlinearly to the unique root λ^* (Dinkelbach 1967). Each evaluation of $F(\lambda)$ requires solving the subproblem (11): minimizing a linear function of z over the set of feasible mixtures in M_B^+, subject to the constraint $\sum_i z_i P_i \geq P_{th}$. We handle this next.

Lagrangian Relaxation

By fixing λ and writing $w_i = V_i - \lambda P_i$, the subproblem (11) is the linear program:

$$\min_{z \in \mathbb{R}^N} \sum_{i=1}^N z_i\, w_i \ \text{ s.t. } \ \sum_i z_i P_i \geq P_{th}, \ \ \sum_i z_i = 1, \ \ z_i \geq 0. \tag{12}$$

The number of variables N is exponential, so we never solve this LP directly. Since both the LP and its dual are feasible, strong duality holds, and the Lagrangian relaxation below is exact.

Relaxing both constraints of (12) with multipliers $\mu \geq 0$ and $\nu \in \mathbb{R}$ gives

$$L(z, \mu, \nu) = \sum_i z_i \big[w_i - \mu P_i - \nu\big] + (\mu\, P_{th} + \nu). \tag{13}$$

Minimizing over $z \geq 0$, this is $-\infty$ unless $w_i - \mu P_i - \nu \geq 0$ for all i, so the dual is

$$\max_{\mu \geq 0,\ \nu} \ (\mu\, P_{th} + \nu) \qquad \text{s.t.} \qquad \nu \leq w_i - \mu\, P_i \ \ \forall\, i. \tag{14}$$

Since we maximize ν, the optimum sets $\nu^* = \min_i [w_i - \mu P_i]$, which reduces the dual to:

$$\max_{\mu \geq 0} \ \varphi(\mu), \qquad \varphi(\mu) = \min_{i \in \{1, \dots, N\}} \big[w_i + \mu\,(P_{th} - P_i)\big]. \tag{15}$$

The function φ is concave and piecewise linear. We maximize it by bisection on μ: each evaluation requires solving

$$\min_{\pi \in \Pi_{\text{det}}(M_B^+)} \big[V^\pi(s_0) - \Lambda\, P^\pi(s_0)\big] \qquad \text{with } \Lambda = \lambda + \mu, \tag{16}$$

which we handle in the next subsection. The inner problem is solved to a tolerance As μ increases, the policy returned has higher success probability; the bisection adjusts μ until the constraint is met. At convergence, if the bisection finds a deterministic policy π^* with $P^{\pi^*} = P_{th}$, it is optimal on its own. In the general case, there exist two policies π_L and π_R that bracket the threshold ($P^{\pi_L} < P_{th} < P^{\pi_R}$), and their mixture $\pi_{\text{mix}} = \alpha\,\pi_L + (1-\alpha)\,\pi_R$ with weight

$$\alpha = \frac{P^{\pi_R} - P_{th}}{P^{\pi_R} - P^{\pi_L}} \tag{17}$$

is optimal over all (possibly stochastic) policies in M_B^+ for the given λ, by strong duality. Since this holds at every iteration of the outer Dinkelbach loop, the final output inherits the same structure.

Corollary 1. *Under Assumption 1, the SCC-SSP problem on the original MDP M admits an optimal policy that is a mixture of at most two deterministic cost-dependent policies, selected once before execution.*

Backward Induction on the Augmented MDP

For a given Λ, we solve problem (16) by equipping M_B^+ with the following cost structure. All intermediate costs are zero, and the only nonzero costs arise at transitions to the goal:

$$c^+\big(\hat{s},\, a,\, \hat{s}'\big) = \begin{cases} r + c(s,a,s') - \Lambda & \text{if } \hat{s}' = \hat{g}, \\ 0 & \text{otherwise.} \end{cases}$$

where $\hat{s} = (s, r)$. The structure of M_B^+ (states, transitions) does not change; only the terminal costs depend on Λ.

Proposition 3 (Equivalence)**.** *For any policy π in M_B^+, the expected cost $W^\pi(s_0, 0)$ under c^+ satisfies $W^\pi(s_0, 0) = V^\pi(s_0) - \Lambda\, P^\pi(s_0)$.*

Proof. A trajectory reaching $\hat{g}$ after accumulating cost r_τ pays $r_\tau + c(s_\tau, a_\tau, g) - \Lambda = C(\theta) - \Lambda$; a trajectory reaching $\hat{d}$ pays 0. Taking the expectation:

$$\begin{aligned} W^\pi(s_0, 0) &= \mathbb{E}^\pi\big[C(\theta)\cdot \mathbf{1}_{\text{succ}}\big] - \Lambda\, \mathbb{E}^\pi[\mathbf{1}_{\text{succ}}] \\ &= V^\pi(s_0) - \Lambda\, P^\pi(s_0). \;\square \end{aligned}$$

Since M_B^+ is a DAG, the optimal value function satisfies the standard Bellman equation:

$$W^*(\hat{s}) = \min_{a \in A(s)} \sum_{s' \in S} T(s,a,s') \,\Big[\, c^+(\hat{s},\, a,\, \hat{s}') + W^*(\hat{s}') \,\Big], \tag{18}$$

where $\hat{s} = (s, r)$, the successor augmented state is $\hat{s}' = \big(s',\, r + c(s,a,s')\big)$, and the boundary condition is

$$W^*(\hat{g}) = 0, \qquad W^*(\hat{d}) = 0.$$

A backward sweep by decreasing r computes W^* for all augmented states in $O(|S| \cdot B \cdot |A|)$ time, since each $\hat{s}' = (s', r')$ with $r' > r$ has already been evaluated.

Complete Algorithm

Algorithm 1 assembles both layers. The outer bisection on λ converges to the optimal conditional cost λ^*. Each iteration calls LAGRANGIANDUAL, which bisects on μ using backward induction on M_B^+ to solve (16), and returns the policy pair (π_L, π_R) together with $F(\lambda)$. When the outer loop terminates, the mixing weight α is computed from (17).

Algorithm 1: Dinkelbach-Lagrangian solver for SCC-SSP

Require: Augmented MDP M_B^+, threshold $P_{th} < P^*(s_0)$, tolerances ε_λ, ε_μ, upper bound M on the Lagrange multiplier μ
Ensure: Policies π_L, π_R and mixing weight α

$\lambda \leftarrow B$
repeat
  $\pi_L,\, \pi_R,\, \alpha \leftarrow$ LAGRANGIANDUAL(λ)
  $V_z \leftarrow \alpha V^{\pi_R} + (1-\alpha) V^{\pi_L}$
  $P_z \leftarrow \alpha P^{\pi_R} + (1-\alpha) P^{\pi_L}$
  $\lambda' \leftarrow V_z / P_z$
until $|\lambda' - \lambda| \le \varepsilon_\lambda$
$\lambda \leftarrow \lambda'$
return π_L, π_R, α

function LAGRANGIANDUAL(λ)
  $\mu_{\min} \leftarrow 0;\ \mu_{\max} \leftarrow M$
  $\pi_L \leftarrow$ nil; $\pi_R \leftarrow$ nil
  while $\mu_{\max} - \mu_{\min} > \varepsilon_\mu$ **do**
    $\mu \leftarrow (\mu_{\min} + \mu_{\max})/2$
    $\pi \leftarrow$ BACKWARDINDUCTION($M_B^+,\ \lambda + \mu$)
    if $P^\pi < P_{th}$ **then**
      $\mu_{\min} \leftarrow \mu;\ \pi_L \leftarrow \pi$
    else
      $\mu_{\max} \leftarrow \mu;\ \pi_R \leftarrow \pi$
    end if
  end while
  $\alpha \leftarrow (P^{\pi_R} - P_{th})/(P^{\pi_R} - P^{\pi_L})$
  return π_L, π_R, α

Convergence. The outer Dinkelbach iteration converges because F is strictly decreasing with a unique zero. The inner bisection converges because $\varphi(\mu)$ is concave with a finite number of breakpoints. Since the LP (12) and its dual are both feasible, strong duality of linear programming guarantees that the Lagrangian relaxation incurs no duality gap, so the returned mixture is a global optimum of SCC-SSP in M_B^+.

Complexity. Each backward induction on M_B^+ costs $O(|S| \cdot B \cdot |A|)$, and the inner bisection on μ requires $O(\log(M/\varepsilon_\mu))$ of them. Writing N_D for the number of outer Dinkelbach iterations (small in practice, as the method converges superlinearly (Dinkelbach 1967)), the total cost is $O\big(N_D \cdot \log(M/\varepsilon_\mu) \cdot |S| \cdot B \cdot |A|\big)$.

5 Experiments

We evaluate SCC-SSP against two baselines on classical planning benchmarks with unavoidable dead-ends. Our goal is to answer two questions: (1) does SCC-SSP produce a richer set of trade-offs between success probability and conditional cost than existing methods? and (2) how large are the efficiency gains over the conservative S3P baseline?

Domains

Triangle Tireworld. Introduced by (Little and Thiébaux 2007) and used as a benchmark at IPPC 2008 (Bryce and

Buffet 2008), this domain models a vehicle navigating a triangular grid of n rows from a start vertex to a goal vertex. After each move, the tire may go flat with probability p_f. Spare tires are available along a safe but longer border path; the interior of the grid has no spares by default. A flat tire with no spare is an irrecoverable dead-end. The operator faces a clear risk-cost trade-off: the shortest path cuts through the unprotected interior, while the safest path follows the border. We fix $p_{\text{change}} = 1$ (tire change always succeeds when a spare is available) and vary grid size n and flat probability p_f. In some instances, we also distribute additional spares uniformly at random over a fraction σ of the interior locations, creating intermediate risk-reward paths.

Instance names follow the convention `TW-`n, with optional suffixes `-pf`x when $p_f \neq 0.3$ (the default) and `-sd`x when $\sigma > 0$. For example, `TW-25-pf0.2` denotes a grid with $n = 25$ rows and flat probability $p_f = 0.2$, while `TW-25-sd0.3` has the default $p_f = 0.3$ with 30% of interior locations receiving spare tires.

River Problem. Proposed by (Freire and Delgado 2017), this domain places an agent on an $N_x \times N_y$ grid crossed by a river. On the banks and the bridge, movements are deterministic. In the river, each action may push the agent one row south with probability p_r, toward a waterfall (dead-end) at the bottom. The goal is at the bottom-right corner.

We select 16 instances with at least 1000 states: 9 Tireworld and 7 River, varying size, risk, and spare density.

Baselines

We compare three methods. **S3P** is solved with the GPCI algorithm (Teichteil-Königsbuch 2012): it maximizes the success probability first, then minimizes the conditional cost among safest policies. **C-MDP** follows the constrained MDP approach (Altman 1999): we solve an LP over occupation measures that minimizes the unconditional expected cost with a finite penalty D_f for dead-end transitions, subject to the constraint $P^{\pi}_{\mathcal{G}}(s_0) \geq P_{th}$. We sweep $D_f \in \{1, 10, 100, 1000\}$ and keep the best feasible policy for each P_{th}. **SCC-SSP** uses our Dinkelbach-Lagrangian algorithm on the augmented MDP $\mathcal{M}^+_B$, with 20 uniformly spaced values of P_{th} and cost budget $B = 4n$ (Tireworld) or $B = 4(N_x + N_y)$ (River). All policies are evaluated on the original MDP to obtain the pair $(P^{\pi}_{\mathcal{G}}, E^{\pi}_c)$.

Results

Aggregate comparison. Table 1 reports, for each instance, the mean and maximum reduction in conditional cost E^{π}_c relative to S3P, computed over 20 values of P_{th}. For the C-MDP baseline, we solve the LP that minimizes the total expected cost with penalty D_f for dead-end transitions, subject to the linear constraint $P^{\pi}_{\mathcal{G}}(s_0) \geq P_{th}$, for each $D_f \in \{1, 10, 100, 1000\}$. At each threshold, we select the policy with the lowest E^{π}_c.

The results show a clear and consistent advantage for SCC-SSP. On Tireworld instances, the C-MDP barely improves over S3P: its average gain stays below 7% across all instances, and often sits near 1%. SCC-SSP, by contrast, achieves average gains between 10% and 23%, with peak reductions reaching 60%. The gap widens as the flat probability increases. At $p_f = 0.5$, for example, the C-MDP gains at most 4% while SCC-SSP still reaches 28%. This happens because higher risk leaves fewer viable stationary policies, which limits what the C-MDP can find. SCC-SSP, working with cost-dependent policies, retains access to a richer set of strategies.

				Gain (%)			
			S3P	C-MDP		SCC-SSP	
Instance	$\lvert S\rvert$	$\lvert S^+\rvert$	E_c	Avg	Max	Avg	Max
Triangle Tireworld							
22	1014	7.14×10^4	62.8	1	10	**17**	**54**
25	1302	1.03×10^5	72.4	1	7	**15**	**47**
30	1862	1.76×10^5	88.4	1	6	**13**	**39**
35	2522	2.76×10^5	104.4	1	5	**11**	**33**
40	3282	4.09×10^5	120.4	1	4	**10**	**29**
25-pf0.2	1302	1.03×10^5	63.2	4	22	**23**	**60**
25-pf0.4	1302	1.03×10^5	81.6	1	5	**11**	**35**
25-pf0.5	1302	1.03×10^5	91.0	0	4	**10**	**28**
25-sd0.3	1302	1.12×10^5	47.1	7	15	**17**	**40**
River							
30x35	1050	2.66×10^5	96.0	33	33	**45**	**60**
35x40	1400	4.11×10^5	111.0	32	32	**44**	**58**
40x50	2000	7.08×10^5	136.0	36	37	**47**	**60**
55x60	3300	1.50×10^6	171.0	30	30	**42**	**54**
35x35-P0.1	1225	3.35×10^5	79.0	40	41	**47**	**56**
35x35-P0.2	1225	3.35×10^5	101.0	39	40	**50**	**61**
35x35-P0.4	1225	3.35×10^5	101.0	0	0	**23**	**45**

Table 1: Comparison across benchmark instances. $|S|$ and $|S^+|$ are the sizes of the original and augmented MDP M^+_B; E_c is the conditional cost of S3P. The four rightmost columns report the *gain*, i.e. the mean/maximum reduction (in %) in E_c relative to S3P over 20 values of P_{th} (larger is better). The best gain on each row is shown in bold. SCC-SSP solves all instances in < 5 s. C-MDP is generally slower (up to 51 s).

On River instances, the C-MDP performs better than on Tireworld instances, with average gains of 30% to 40%. The River domain has a simpler structure that gives stationary policies more room to maneuver. Yet SCC-SSP still outperforms it on every instance, with average gains of 42% to 50%. The most striking case is RV-35×35 with $p_r = 0.4$: the C-MDP finds no improvement at all over S3P, while SCC-SSP reduces the conditional cost by 23% on average and up to 45%.

A closer look at a single instance. To understand what drives these differences, we examine one Tireworld instance in detail ($n = 25$, $p_f = 0.3$, $\sigma = 0.3$, 1302 states). We use a finer grid of penalty values $D_f \in \{5, 20, 40, 50, 100\}$ to better illustrate how the C-MDP behavior varies with D_f. Figure 2 plots the achieved success probability (top) and the conditional cost (bottom) as functions of P_{th}.

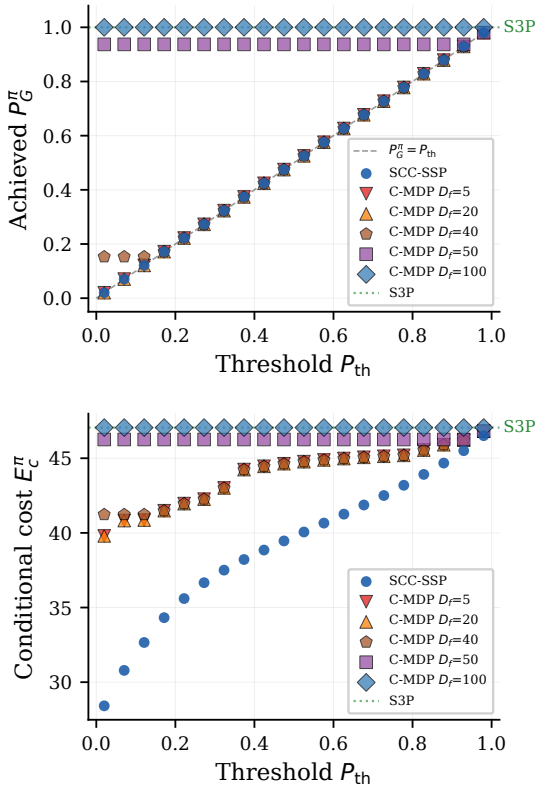

Figure 2: Triangle Tireworld ($n=25, p_f=0.3, \sigma=0.3$, 1302 states). Top: achieved success probability vs. threshold P_{th}. Bottom: conditional cost E_c^π vs. P_{th}. C-MDP is solved for $D_f \in \{5, 20, 40, 50, 100\}$, each under $P_{\mathcal{G}}^\pi \geq P_{th}$.

The top panel shows how each method responds to the safety requirement. SCC-SSP tracks the diagonal: the success probability of the returned policy is close to P_{th} at every threshold, delivering the requested safety without overshooting. The C-MDP tells a different story. For $D_f = 100$, the policy always reaches the goal with probability 1: avoiding dead-ends dominates the LP objective, the safety constraint plays no role, and the solver behaves exactly like S3P. For $D_f = 50$, the success probability jumps to about 0.94 even when P_{th} is as low as 0.15. Only for small penalties ($D_f = 5$ or $D_f = 20$) does the C-MDP produce policies whose success probability varies with the threshold. The behavior of the C-MDP is thus governed by D_f, not by P_{th}: when D_f is large, the constraint is useless; when small, the penalty still shapes the policy. The bottom panel reveals a subtle point: even when C-MDP matches the exact success probability of SCC-SSP, its conditional cost remains systematically higher. At $P_{th} = 0.02$, both policies achieve $P_{\mathcal{G}}^\pi = 0.02$, yet SCC-SSP yields $E_c^\pi = 28.4$ compared to 39.8 for C-MDP. This is not due to conservatism but to the linear C-MDP objective itself: minimizing a combination of overall trajectory costs and dead-end penalties based on occupation measures, the LP finds the globally cheapest way to satisfy the constraint, but lacks any incentive to explicitly minimize the conditional cost of the successful trajectories.

SCC-SSP behaves differently. Because it optimizes $V^\pi / P_{\mathcal{G}}^\pi$ directly, lowering P_{th} leads the solver to select genuinely different routes: shorter and riskier paths when the threshold is low, longer and safer paths when it is high. The result is a smooth curve on the bottom panel, where the conditional cost decreases steadily as the operator accepts more risk. The C-MDP, in contrast, keeps roughly the same underlying strategy across thresholds and adjusts only the fraction of trajectories that succeed.

Discussion. The comparison above may seem unfair in one respect: neither S3P nor the C-MDP optimizes the conditional cost directly, so they are not designed to perform well on our metric. This is precisely the point: to the best of our knowledge, no existing planner targets the conditional objective (5). The closest criterion is GUBS (Freire and Delgado 2017), recently revisited in (Roche, Watanabe, and Chanel 2026), which also produces cost-dependent policies on an augmented state space, but differs from SCC-SSP in two respects.

First, GUBS aggregates successful and failed trajectories into a single non-linear utility, whereas SCC-SSP measures efficiency exclusively over goal-reaching paths. Second, the GUBS risk parameter K_g offers only qualitative control over safety: increasing K_g produces more conservative policies, but there is no direct mapping between K_g and the success probability that the resulting policy achieves. SCC-SSP takes P_{th} as a direct input. More broadly, in several domains the conditional cost is not just an alternative criterion but the operationally correct one, as in the firefighting drone example introduced in the introduction: if a drone crashes before reaching the fire, the time it spent in the air contributes nothing to the mission; what matters is how many drones arrive and how fast they do.

6 Conclusion and Future Work

We introduced SCC-SSP, a formulation that minimizes the expected cost conditioned on reaching the goal subject to a user-specified safety threshold. The ratio structure of this objective places the problem outside standard C-MDP methods. Under integer costs and bounded budget, we showed that optimal policies are cost-dependent and that a mixture of at most two deterministic cost-dependent policies suffices. Our exact three-layer algorithm combines Dinkelbach's method, Lagrangian relaxation, and backward induction on an augmented MDP; in benchmarks it traces the full safety-cost Pareto front, improving on S3P by up to 60%.

The budget assumption is both a modeling choice and a computational lever: any policy optimal in M_B^+ is feasible in the original MDP, but optimality is guaranteed only among policies that stay within budget. The augmented state space scales linearly in B, which can become prohibitive for large instances. Developing forward search methods such as LRTDP or LAO* on the augmented space would avoid full enumeration and is a natural next step. Relaxing the integer-cost assumption to handle continuous costs is another open direction, as the DAG construction relies on discreteness.

References

Altman, E. 1999. *Constrained Markov Decision Processes*. London: CRC Press.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1991. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3): 580–595.

Bryce, D.; and Buffet, O. 2008. International Planning Competition: Uncertainty Part: Benchmarks and Results. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Chatterjee, K.; Majumdar, R.; and Henzinger, T. A. 2006. Markov decision processes with multiple objectives. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3884 of *LNCS*, 325–336. Springer.

Crispino, G. N.; Freire, V.; and Delgado, K. V. 2023. GUBS criterion: Arbitrary trade-offs between cost and probability-to-goal in stochastic planning based on expected utility theory. *Artificial Intelligence*, 316: 103848.

Dinkelbach, W. 1967. On Nonlinear Fractional Programming. *Management Science*, 13(7): 492–498.

Freire, V.; and Delgado, K. V. 2017. GUBS: A Utility-Based Semantic for Goal-Directed Markov Decision Processes. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 741–749.

Freire, V.; Delgado, K. V.; and Reis, W. A. S. 2019. An Exact Algorithm to Make a Trade-Off between Cost and Probability in SSPs. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS)*, 157–165. AAAI Press.

Hong, S.; Lee, S. U.; Huang, X.; Khonji, M.; Alyassi, R.; and Williams, B. C. 2021. An Anytime Algorithm for Chance Constrained Stochastic Shortest Path Problems and Its Application to Aircraft Routing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 475–481. IEEE.

Kolobov, A.; Mausam; and Weld, D. S. 2012. A Theory of Goal-Oriented MDPs with Dead Ends. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, 438–447. AUAI Press.

Little, I.; and Thiébaux, S. 2007. Probabilistic Planning vs Replanning. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 224–231. AAAI Press.

Mausam; and Kolobov, A. 2012. *Planning with Markov Decision Processes: An AI Perspective*. Morgan & Claypool Publishers.

Roche, M.; Watanabe, Y.; and Chanel, C. P. C. 2026. Planning with Formal Reachability Guarantees in Goal-Oriented MDPs with Dead-Ends. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI)*. To appear.

Steinmetz, M.; Hoffmann, J.; and Buffet, O. 2016. Goal Probability Analysis in Probabilistic Planning: Exploring and enhancing the state of the art. *Journal of Artificial Intelligence Research (JAIR)*, 57: 229–271.

Teichteil-Königsbuch, F. 2012. Stochastic Safest and Shortest Path Problems. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 1825–1831. AAAI Press.

Trevizan, F.; and Teichteil-Königsbuch, F. 2017. Efficient Solutions for Stochastic Shortest Path Problems with Dead Ends. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 1–10. AUAI Press.

Trevizan, F.; Thiébaux, S.; Santana, P.; and Williams, B. 2017. I-dual: Solving Constrained SSPs via Heuristic Search in the Dual Space. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 4399–4405.