

Probability-to-Goal and Expected Cost Trade-Off in Stochastic Shortest Path

Isabella Kuo and Valdinei Freire(✉)

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo, São Paulo, Brazil
{15483114.ik,valdinei.freire}@usp.br

Abstract. Markov Decision Processes (MDPs) model problems where a decision-maker makes sequential decisions and the effect of decisions is probabilistic. A particular formulation of MDPs is the Shortest Stochastic Path (SSP), in which the agent seeks to accomplish a goal while reducing the cost of the path to it. Literature introduces some optimality criteria; most of them consider a priority of maximizing probability to accomplish the goal while minimizing some cost measure; such criteria allow a unique trade-off between probability-to-goal and path cost for a decision-maker. Here, we present algorithms to make a trade-off between probability-to-goal and expected cost; based on the Minimum Cost given Maximum Probability (MCMP) criterion, we propose to treat such a trade-off under three different methods: (*i*) additional constraints for probability-to-goal or expected cost; (*ii*) a Pareto's optimality by finding non-dominated policies; and (*iii*) an efficient preference elicitation process based on non-dominated policies. We report experiments on a toy problem, where probability-to-goal and expected cost trade-off can be observed.

Keywords: Shortest stochastic path · Markov Decision Process · Preference elicitation · Non-dominated policies · Goal and cost trade-off

1 Introduction

Markov Decision Process (MDP) models problems where an agent makes sequential decisions and the results of these actions are probabilistic [9]. The agent's objective is to choose a sequence of actions so that the system acts optimally according to previously defined criteria. Such a sequence of actions follows a policy, which maps each observed state into an action.

A special case of MDP is the Stochastic Shortest Path (SSP) [2], where a terminal goal state is considered. The agent is supposed to reach the goal state while minimizing the incurred cost. Two different theoretical scenarios appear whether there exists or there does not exist a proper policy. A proper policy guarantees that the goal state is reached with probability 1.

Usually, when there exists a proper policy, the objective of the agent is to minimize the expected accumulated cost. If a proper policy does not exist, the

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12951, pp. 111–125, 2021.
https://doi.org/10.1007/978-3-030-86970-0_9

agent may get trapped in a dead-end; in this case, infinity cost may be incurred and the problem of minimizing the expected accumulated cost is not well-posed. There exist some approaches in the literature to overcome SSP problems with Unavoidable Dead-End.

The iSSPUDE (SSP with Unavoidable Dead-End) criterion [8] considers two opposite objectives: probability-to-goal and cost-to-goal; and gives priority for the first. Then, an optimal policy maximizes the chance of reaching the goal state, while minimizing the average cost incurred in traces that reaches the goal. Because only traces that reach the goal are considered, the objective cost-to-goal is finite and iSSPUDE is well-posed. Similarly, the MCMP (Minimum Cost given Maximum Probability) criterion [15] considers only policies that maximize probability-to-goal. An additional action of giving up is considered; if a dead-end state is reached, the agent gives up, guaranteeing a finite cost in every trace. Then, different from iSSPUDE, MCMP considers costs of traces that do not reach a goal state and pays the cost incurred until the agent reaches a dead-end.

As pointed out in [6], prioritizing probability-to-goal may incur in undesirable decisions such as trading off a huge cost increase for a tiny probability-to-goal increase. The GUBS criterion [6,7] proposes a semantic of goal prioritization based on the expected utility theory and shows that a more reasonable trade-off can be obtained under a rational approach. Other less refined criteria consist of penalizing traces with dead-ends, for example, by assigning a final penalty cost when giving up [8] or making use of discount factor [13,14].

Although criteria that allow a trade-off between cost-to-goal and probability-to-goal is necessary to avoid undesirable decision in real problems, they require some preference elicitation process to tune parameters in any formalization [1, 3]. Because the GUBS (Goal with UtilityBased Semantics) problem is based on Expected Utility theory, it may be more suitable to preference elicitation processes independent of the SSP dynamics and a trade-off specification can be obtained beforehand. However, setting appropriately the amount of penalty to giving up depends on the SSP dynamics, and requires an on-line preference elicitation based on the SSP problem at hand.

Based on the algorithms developed for MCMP problems and the final penalty cost to giving up, we contribute with an algorithm to obtained non-dominated policies regarding the trade-off of probability-to-goal and cost-to-goal. Moreover, we show how to use such an algorithm to support decisions by eliciting the preference of a decision-maker.

2 Background

2.1 Stochastic Shortest Path

In this article, we consider a special case of Markov Decision Process, the Stochastic Shortest Path (SSP) problem [2]. A SSP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, C, s_0, \mathcal{G} \rangle$ where:

- $s \in \mathcal{S}$ are possible states;
- $a \in \mathcal{A}$ are possible actions;

– $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function;
– $\mathcal{C} : \mathcal{S} \times \mathcal{A} \to R^+$ is the cost function;
– $s_0 \in \mathcal{S}$ is the initial state; and
– $\mathcal{G}$ is the set of goal states.

A SSP defines a process of an agent interacting with an environment and at all time step t: (i) the agent observes a state s_t, (ii) the agent chooses an action a_t, (iii) the agent pays a cost c_t; and (iv) the process moves to a new state s_{t+1}. The process ends when a goal state $s \in \mathcal{G}$ are reached. Transitions and costs present Markov's property, i.e., they both depend only on the current state s_t and chosen action a_t.

The solution for a SSP consists of policies that describe which actions should be taken in each situation. Here we consider deterministic stationary policies $\pi : \mathcal{S} \to \mathcal{A}$ that maps each state for a unique action. A policy is proper if it guarantees that the goal is reached with probability 1 when the process starts in the state s_0.

Classically, only problems with proper policies are considered. Given a proper policy, the expected accumulated cost to any state $s \in \mathcal{S}$ is defined as:

$$V^{\pi}(s) = \mathrm{E}\left[\sum_{t=0}^{\infty} c_t \middle| s_0 = s, \pi\right] \tag{1}$$

and the optimal policy is defined by $\pi^*(s) = \arg\min_{\pi \in \Pi} V^{\pi}(s)$, where Π is the set of deterministic stationary policies.

When a proper policy does not exist, dead-ends occur, which are states from which a goal state cannot be reached. In the presence of dead-ends, the expected accumulated cost is infinite and $V^{\pi}(s)$ is uninformative. Therefore, criteria different from the classic one should be considered.

An alternative criterion consists in maximizing the probability of reaching a goal state in $\mathcal{G}$ from the initial state s_0 (probability-to-goal). While probability-to-goal can differentiate policies, such a criterion is not well-posed, since policies with completely different cost profile can present the same probability-to-goal. MCMP and iSSPUDE consider probability-to-goal as a first criterion and present some cost-dependent informative criterion as a tiebreaker [8,15].

2.2 Minimum Cost Given Maximum Probability (MCMP) Criterion

The MCMP criterion [15] considers policies that maximize probability-to-goal, while minimizing expected cost. To avoid infinite expected costs, the agent is allowed to giving up the process when he/she finds himself/herself in a dead-end.

Formalization. When following a policy, the process described by a SSP generates a random state-trace T, i.e., a sequence of states. Traces can be finite, when the agent reached a goal state, or infinite, otherwise. Consider the following concepts:

- T^i: the i-th state of the trace T,
- $\mathcal{T}_{\pi,s}$: the set of all traces obtained from s following a policy π, and
- $\mathcal{T}^{\mathcal{G}}_{\pi,s} \subseteq \mathcal{T}_{\pi,s}$: the subset of traces that reach the goal.

Note that, if a trace $T \in \mathcal{T}_{\pi,s}$ is finite, then T reaches a goal state. The occurrence probability for any finite trace $T \in \mathcal{T}^{\mathcal{G}}_{\pi,s}$ conditioned on $s_0 = s$ and policy π is given by:

$$P_{T,\pi} = \prod_{i=0}^{|T|-1} \mathcal{P}(T^{i+1}|T^i, \pi(T^i)). \tag{2}$$

The probability that a policy π will reach a goal state in $\mathcal{G}$ when starting in state $s \in \mathcal{S}$ is:

$$P^{\mathcal{G}}_{\pi,s} = \sum_{T \in \mathcal{T}^{\mathcal{G}}_{\pi,s}} P_{T,\pi},$$

and the accumulated cost of a finite trace is

$$C_{T,\pi} = \sum_{i=0}^{|T|-1} \mathcal{C}(T^i, \pi(T^i)).$$

To deal with infinite traces, the MCMP criterion allows a give up action, that is, when it is faced with a dead-end, it can give up and pay the cost spent up to that moment. To ensure that the probability of reaching the goal is maximized, the give up action is used only when a dead-end is found.

Consider the following auxiliary function that makes all traces of a policy finite:

$$\psi(sT) = \begin{cases} s & \text{, if } |T| = 0 \text{ or } P^{\mathcal{G}}_{\pi,s} = 0 \\ s\psi(T) & \text{, otherwise} \end{cases}. \tag{3}$$

If T is an infinite trace, then $\psi(T) = T'$ is a truncated version of T that ends up in the first dead-end in trace T when following π. If T is a finite trace, then $\psi(T) = T$ and T ends up in a goal state.

Consider $p_{max} = \max_{\pi \in \Pi} P^{\mathcal{G}}_{\pi,s_0}$, that is, the maximum probability of reaching the goal from the initial state s_0, an optimal policy under the MCMP criterion is defined by:

$$\pi^* = \arg \min_{\{\pi | P^{\mathcal{G}}_{\pi,s_0} = p_{max}\}} \sum_{T \in \mathcal{T}_{\pi,s_0}} C_{\psi(T),\pi} P_{T,\pi}. \tag{4}$$

Here, the expected cost is calculated over truncate versions of traces occurred when starting in the initial state s_0 and following policy π.

Linear Programming Solution. The MCMP criterion can be solved using two linear programming problems, LP1 and LP2 [15]. LP2 finds the maximum probability p_{max} to reach a goal state. LP1 finds the minimum expected cost policy that reaches a goal state with probability p_{max}.

Both problems consider variables $x_{s,a}$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ that indicates an expected accumulated occurrence frequency for every state-action pair. The SSP

dynamics restricts the solutions by specifying an $in(s)$ and $out(s)$ flow model for every state s. Every state but initial state and goal state must equalize $in(s)$ and $out(s)$. Initial state s_0 presents a unity in-out difference, while goal states in $\mathcal{G}$ has no output. For every state $s \in \mathcal{S}$, we have:

$$in(s) = \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} x_{s',a} \mathcal{P}(s|s', a) \quad \text{and} \quad out(s) = \sum_{a \in \mathcal{A}_{(s)}} x_{s,a}. \tag{5}$$

The LP2 finds the p_{max} value by maximizing the accumulated input for all the goal states in $\mathcal{G}$ and by not considering at all accumulated cost. LP2 is described as follows:

$$\begin{array}{lll} \text{LP2} & \max_{x,a} & \sum_{s_g \in \mathcal{G}} in(s_g) \\ & \text{s.t.} & x_{s,a} \geq 0 \qquad\qquad\quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ & & out(s) - in(s) = 0 \quad \forall s \in \mathcal{S} \backslash (\mathcal{G} \cup s_0) \\ & & out(s_0) - in(s_0) = 1 \end{array}. \tag{6}$$

The LP1 finds the minimum cost given maximum probability p_{max}. LP1 differs from LP2 regarding four aspects:

1. the flow that enters an state is allowed to be bigger than the flow that runs out, this characterize the give up action;
2. previous aspect is also applied to initial state s_0, i.e., give up action can be applied since the initial state;
3. the flow that enters goal state must be p_{max} and guarantees that the optimal policy maximize probability-to-goal; and
4. the objective minimizes the expected cost.

Implicitly, previous conditions guarantee Eq. 4. The linear programming LP1 is described as follows:

$$\begin{array}{lll} \text{LP1} & \min_{x_{s,a}} & \sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s, a) \\ & \text{s.t.} & x_{s,a} \geq 0 \qquad\qquad\quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ & & out(s) - in(s) \leq 0 \quad \forall s \in \mathcal{S} \backslash (\mathcal{G} \cup s_0) \\ & & out(s_0) - in(s_0) \leq 1 \\ & & \sum_{s_g \in \mathcal{G}} in(s_g) = p_{max} \end{array}. \tag{7}$$

The optimal policy π^* to the MCMP criterion can be obtained from expected accumulated occurrence frequency $x_{s,a}$ in LP1 by:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} x_{s,a} \quad \forall s \in \mathcal{S},$$

and π^* incurs in expected cost $c_{min} = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s, a)$.

3 Probability-to-Goal and Average Cost Trade-Off

The MCMP and iSSPUDE criteria establish a preference system that prioritizes probability-to-goal, disregarding the associated cost. The great advantage of

MCMP and iSSPUDE is that both criteria have no parameter to be tuned. However, as highlighted by [6], a small reduction in the probability to reach the goal can bring a considerable reduction in the expected cost and decision-makers should be able to choose a better trade-off between probability-to-goal and expected cost. In the next subsections, we discuss and propose different strategy to obtain better trade-offs between Probability-to-Goal and Average Cost.

3.1 Trade-Offs Under Additional Constraint

Trade-offs can be easily founded by constraining one of the variable: probability-to-goal or expected cost. First, make use of LP2 and LP1 to find p_{max} and c_{min}. Then, ask for a decision-maker an additional equality constrain on probability-to-goal or expected cost with range on $[0, p_{max}]$ and $[0, c_{min}]$, respectively.

If a constraint in probability-to-goal $p \leq p_{max}$ is desired, minimize the expected cost under the constraint that $\sum_{s_g \in \mathcal{G}} in(s_g) = p$, i.e.,

$$\begin{aligned} \text{LP3} \qquad \min_{x_{s,a}} \ & \textstyle\sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s,a) & & \\ \text{s.t.} \ & x_{s,a} \geq 0 & & \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ & out(s) - in(s) \leq 0 & & \forall s \in \mathcal{S} \backslash (\mathcal{G} \cup s_0)\,, \\ & out(s_0) - in(s_0) \leq 1 & & \\ & \textstyle\sum_{s_g \in \mathcal{G}} in(s_g) = p & & \end{aligned} \tag{8}$$

and return expected cost $c = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s,a)$.

If a constraint in expected cost $c \leq c_{min}$ is desired, maximize the probability-to-goal under the constraint that $\sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s,a) = c$, i.e.,

$$\begin{aligned} \text{LP4} \qquad \max_{x,a} \ & \textstyle\sum_{s_g \in \mathcal{G}} in(s_g) & & \\ \text{s.t.} \ & x_{s,a} \geq 0 & & \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ & out(s) - in(s) \leq 0 & & \forall s \in \mathcal{S} \backslash (\mathcal{G} \cup s_0)\,. \\ & out(s_0) - in(s_0) \leq 1 & & \\ & \textstyle\sum_{s \in \mathcal{S}, a \in \mathcal{A}} x_{s,a} \mathcal{C}(s,a) = c & & \end{aligned} \tag{9}$$

and return probability-to-goal:

$$p = \sum_{s_g \in \mathcal{G}} in(s_g) = \sum_{s_g \in \mathcal{G}} \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} x_{s',a} \mathcal{P}(s_g | s', a).$$

3.2 Non-dominated Policies

Although with additional constraints it is possible to obtain any trade-off for a SSP problem, it demands of decision-makers additional information which may not be at hand. Another alternative is to present Pareto optimal policies, i.e., non-dominated policies regarding probability-to-goal and expected cost. Then, decision-makers may choose which one is better regarding their preference. In this section we make use of a geometric approach to find non-dominated policies for SSPs [11].

Consider a policy π^{α} found by minimizing a linear combination between probability-to-goal and expected cost, i.e.,

$$\begin{array}{lll} \text{LP5} & \min_{x,a} \ \sum_{s\in\mathcal{S},a\in\mathcal{A}} x_{s,a}\mathcal{C}(s,a) - \alpha \sum_{s_g\in\mathcal{G}} in(s_g) & \\ & \text{s.t. } x_{s,a} \geq 0 & \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ & out(s) - in(s) \leq 0 & \forall s \in \mathcal{S}\backslash(\mathcal{G} \cup s_0) \\ & out(s_0) - in(s_0) \leq 1 & \end{array} . \tag{10}$$

Note that policy π^{α} is non-dominated. For $\alpha = 0$, LP5 returns probability-to-goal equal 0 and expected cost equal 0; and for $\alpha \to \infty$, LP5 returns probability-to-goal equal p_{max} and expected cost equal c_{min}.

Consider two non-dominated policies π' and π'' and their respective probabilities-to-goal and expected costs, (p', c') and (p'', c''). Then, it is possible to design a non-stationary probabilistic policy π^{β} that returns any interpolation between (p', c') and (p'', c''). At time step 0, policy π^{β} chooses policy π' with probability β and follows it thereafter, and chooses policy π'' with probability $1 - \beta$ and follows it thereafter.

In a finite SSP, i.e., when $|\mathcal{S}| < \infty$, we have a finite amount of stationary deterministic policies. In this case, the set of non-dominated policies are formed by: (i) a finite set of stationary deterministic policies ordered by their probability-to-goal returns; and (ii) interpolation between two consecutive stationary deterministic non-dominated policies. We introduce NDGC algorithm (Non-Dominated Probability-to-Goal and Expected Cost Trade-off), an anytime algorithm to find the set of stationary deterministic non-dominated policies Π_{ND}.

NDGC algorithm is based on the idea of non-dominated witness [10]. Given any two stationary deterministic non-dominated policies π' and π'', a non-dominated witness is a stationary deterministic policy π^w with respective return (p^w, c^w) that dominates some interpolation π^{β} of π' and π''.

Figure 1 shows the NDGC algorithm working. NDGC algorithm starts with two non-dominates policies: policy π^0, the policy that gives up at time step 0, and policy π^*, the optimal policy for MCMP criterion; and all interpolation between them. Call both policies generically π' and π'' with respective performance (p', c') and (p'', c''), NDGC algorithm finds a witness that is the farthest performance from any interpolation (see Fig. 1.a)); this is done by choosing $\alpha > 0$ such that $c' - \alpha p' = c'' - \alpha p''$. Then, the original interpolations are substituted by two new ones based on π' and π^w; and π^w and π''. NDGC algorithm continues by choosing one of the interpolations and finding a new witness (Fig. 1). NDGC algorithm stops when no witness is founded for any interpolation.

Algorithm 1 details the NDGC algorithm. Initial interpolation is found by solving MCMP problem (lines 1–3); then, the initial interpolation is added to a queue (lines 4–5). For each pair of policies in the queue, a witness policy is found if it does exist (lines 7–9). The witness policy is evaluated regarding the difference in cost between witness solution and interpolation (line 10). If the π^w improves on cost more than ϵ (line 11), π^w is added to *NonDominatedPolicies* (line 12) and both of the new interpolations are added to *queue* (lines 13–14). Note that

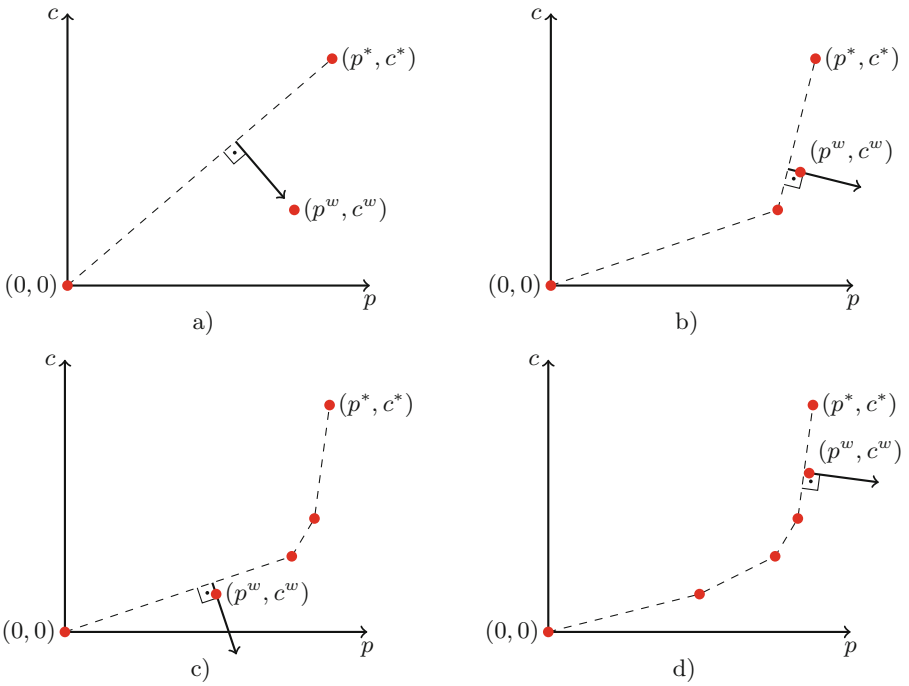

Fig. 1. Graphic demonstration of NDGC algorithm working.

after any iteration, we always have an approximation to the non-dominate set of policies.

3.3 Preference Elicitation

The NDGC algorithm finds out every non-dominated policy within an arbitrary error ϵ. Although it may inform a decision-maker a limited set of policies, it may generate too many ones. This can be a burden regarding computational complexity and also decision-maker cognition effort.

Based in [12], we adapt NDGC algorithm to a preference elicitation algorithm, PEGC algorithm, which elicits decision-maker preference. First, it decreases cognition effort on decision-makers by asking dichotomy questions, i.e., preference between two policies, both summarized in probability-to-goal and expected cost. Second, it decreases computational complexity by generating just a few policies to infer decision-maker preference, NDGC algorithm manages to do so by a search procedure that restricts the number of LP problems to be solved.

Algorithm 2 details the PEGC algorithm. PEGC algorithm considers that decision-maker evaluates policies linearly on probability-to-goal and expected cost, i.e., the value of a policy π^w is $V(\pi^w) = \alpha^* p^w - c^w$, where α^* is the unknown decision-maker trade-off.

Algorithm 1: NDGC algorithm (Non-Dominated Probability-to-Goal and Expected Cost Trade-off)

Input: SSP $\mathcal{M}$, threshold ϵ
1 $p_{max} \leftarrow \text{LP2}(\mathcal{M})$
2 $(\pi^*, p^*, c^*) \leftarrow \text{LP1}(\mathcal{M}, p_{max})$
3 $NonDominatedPolicies \leftarrow \{\pi^0, \pi^*\}$
4 $queue \leftarrow \emptyset$
5 $queue.Add((0,0),(p^*,c^*))$
6 **while** *priorityQueue is not empty* **do**
7 $\quad (p',c'),(p'',c'') \leftarrow priorityQueue.Remove()$
8 $\quad \alpha \leftarrow \dfrac{c''-c'}{p''-p'}$
9 $\quad (\pi^w, p^w, c^w) \leftarrow \text{LP5}(\mathcal{M}, \alpha)$
10 $\quad d \leftarrow c' + \alpha(p^w - p')$
11 $\quad$ **if** $d > \epsilon$ **then**
12 $\quad\quad NonDominatedPolicies \leftarrow NonDominatedPolicies \cup \{\pi^w\}$
13 $\quad\quad queue.Add((p',c'),(p^w,c^w))$
14 $\quad\quad queue.Add((p^w,c^w),(p'',c''))$
15 $\quad$ **end**
16 **end**
Output: $NonDominatedPolicies$

Suppose that policy π' is preferred to policy π'', and remember that $\alpha^* > 0$, then:

$$\begin{aligned} V(\pi') &> V(\pi'') \\ \alpha^* p' - c' &> \alpha^* p'' - c'' \\ \alpha^* &< \frac{c''-c'}{p''-p'} \end{aligned}$$

and, by choosing $\alpha = \frac{c''-c'}{p''-p'}$ for LP5, PEGC algorithm discovers a new non-dominated policy π^w that is better than π''. The same analysis can be made when policy π'' is preferred to policy π', in this case, we have that:

$$\alpha^* > \frac{c''-c'}{p''-p'}.$$

In PEGC algorithm, a pair of extreme policies is found by solving MCMP problem (lines 1–3). Then, the following steps are repeated:

1. a witness policy is founded (lines 5–6);
2. an observation of decision-maker preference is made (line 5–6); and
3. conditioned on decision-maker preference, the witness policy substitute π' or π'' (line 9–15).

The PEGC algorithm stops if the witness policy is too similar to the interpolation of π' and π''.

Algorithm 2: PEGC algorithm (Preference Elicitation Probability-to-Goal and Expected Cost Trade-off)

Input: SSP $\mathcal{M}$, threshold ϵ
$p_{max} \leftarrow \text{LP2}(\mathcal{M})$
$(\pi', p', c') \leftarrow (\pi^0, 0, 0)$
$(\pi'', p'', c'') \leftarrow \text{LP1}(\mathcal{M}, p_{max})$
while *true* **do**
  $\alpha \leftarrow \dfrac{c'' - c'}{p'' - p'}$
  $(\pi^w, p^w, c^w) \leftarrow \text{LP5}(\mathcal{M}, \alpha)$
  $d \leftarrow c' + \alpha(p^w - p')$
  ask decision maker between (p', c') and (p'', c'')
  if (p', c') *is preferred to* (p'', c'') **then**
    $(\pi'', p'', c'') \leftarrow (\pi^w, p^w, c^w)$
    $\pi^* \leftarrow \pi'$
  else
    $(\pi', p', c') \leftarrow (\pi^w, p^w, c^w)$
    $\pi^* \leftarrow \pi''$
  end
  if $d < \epsilon$ **then**
    break
  end
end
Output: π^*

4 Experiments

In this section we report experiments of NDGC and PEGC algorithms. We designed experiments on the River Domain [7] where trade-offs between probability-to-goal and expected cost can be easily observed.

4.1 The River Problem

The River problem, as described in [7], considers a grid world where an agent starts in one side of the river and must reach a goal state in the other side (Fig. 2).

The safest path consists in walking along the riverside until a bridge, walk by the bridge, and go back by the opposite riverside. On the other hand, the shortest path consists in crossing the river; however, actions in the river are influenced by the flow of the river and the agent may be dragged to the waterfall, which is a dead-end.

When walking on the riverside, there is a small probability of falling at the river; therefore, the river problem does not present proper policies for the initial state. When inside the river, instead of behaving deterministically, the river flow compels with a influence chance the agent to the waterfall. Depending on

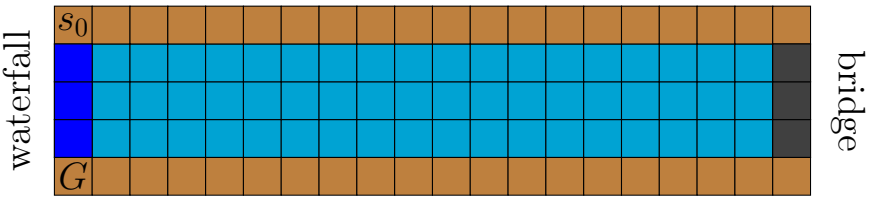

Fig. 2. 5 × 20 River problem.

whether the agent avoids the river, or where the agent chooses to enter the river incurs in different probability-to-goal and expected cost trade-offs.

We experiment on two sizes of River problems: 5×40 and 10×60. The chance of falling into the river was set to 0.1 and the influence chance of the river flow was set to 0.5 (see [7] for a complete description of the river problem).

4.2 Non-Dominated Policies

Figures 3 and 4 show the trade-off between probability-to-goal and expected cost for 5×40 and 10×60 river problems, respectively. We run NDGC algorithm with $\epsilon = 0.1$.

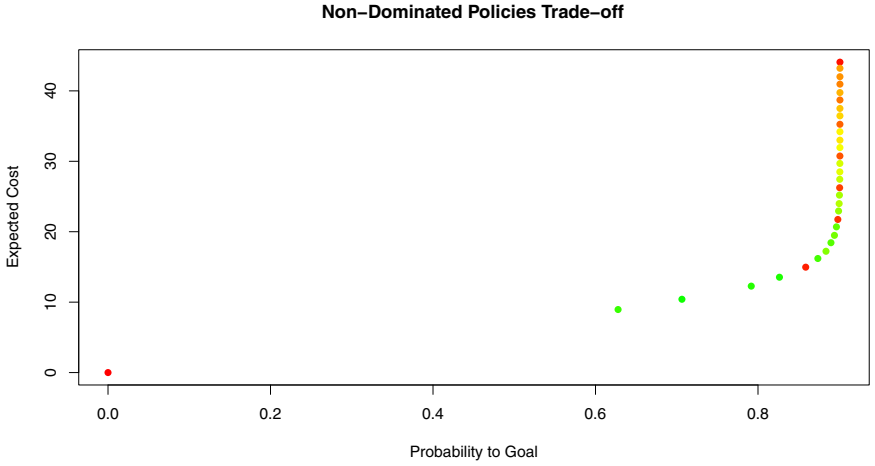

Fig. 3. Non-dominated policies for the 5 × 40 river problem. (Color figure online)

We plot trade-off points in order of occurrence in NDGC algorithm, from red (first) to green (last). In both scenarios, we can see that there exist many don-dominated policies with very close probability-to-goal. Although the number of non-dominated policies can be large (more than 30 policies for small problems), we can see it is possible to have a good approximation with just a few policies.

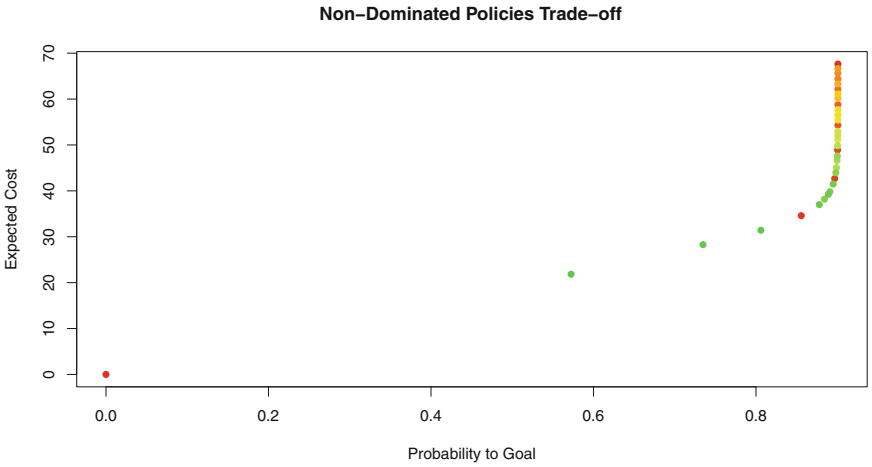

Fig. 4. Non-dominated policies for the 10×60 river problem. (Color figure online)

In the case of size 5×40 we have that $p_{max} = 0.9009$ and $c_{min} = 44.08$. If we allow a probability-to-goal $p = 0.8982$, we have expected cost $c = 21.74$, half of c_{min}. In the case of size 10×60 we have that $p_{max} = 0.9009$ and $c_{min} = 67.64$. If we allow a probability-to-goal $p = 0.8983$, we have expected cost $c = 43.96$, two thirds of c_{min}.

As already noted in [6], a tiny decrease in maximum probability-to-goal, in general, brings a large decrease in expected cost. This experiments corroborate the hypotheses that giving priority to probability-to-goal, as in iSSPUDE and MCMP, may incur on unnecessary spent in cost.

Note that a probability-to-goal and expected cost trade-off can be easily observed after the non-dominated policies are plotted in a graph. Knowing such graph, a decision-maker can make a informed choice over policies.

4.3 Preference Elicitation

To evaluate PEGC algorithm, we simulate a decision-maker whose preference equals MCMP criterion and test it in the 10×60 river problem. In this case, the decision-maker prefers policy π^* to any other. Note that this decision-maker represents the worst scenario for PEGC algorithm, since its preference is located in the most dense region in Fig. 4.

Figure 5 shows the trade-off between probability-to-goal and expected cost for all policies found during PEGC algorithm execution; 10 policies were necessary.

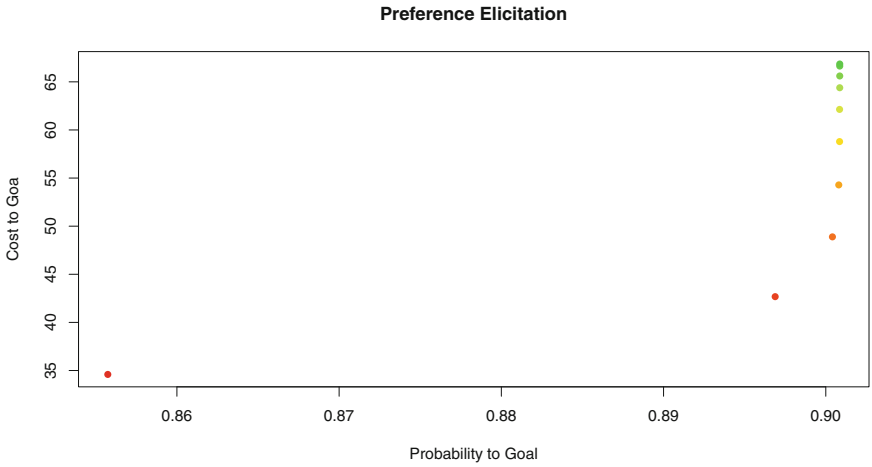

Fig. 5. Subset of non-dominated policies for the 10×60 river problem when executing PEGC algorithm.

5 Conclusion

In this work, were presented three methods to help a decision-maker to find a personal trade-off between probability-to-goal and the expected cost. In the first method, additional constraints, the decision-maker proposes an additional constraint to the SSP problem; to help the decision-maker, extremes trade-offs $(0, 0)$ and (p^*, c^*) are informed beforehand. In the second method, non-dominated policies, the decision-maker is informed with a set of non-dominated policies, from which he/she can choose the one that best pleases him/her. In the third method, preference elicitation, dichotomy questions are formulated and asked decision-maker; in the end, the policies that best pleases him/her is returned.

Although we experiment on small toy problems, the conclusion is powerful. First, by using some criteria from the literature, such as MCMP or iSSPUDE, may generate a disastrous decision in real problems; a huge expected cost may be paid for a small increase in probability-to-goal. Second, the decision-maker must be in the loop if we want to avoid such a disastrous decision, we have offered three types of interaction between decision-maker and stochastic planning algorithms. Third, there is an inverse proportional relationship between the cognitive effort required from the decision-maker and computational effort.

We only considered the trade-off between probability-to-goal and the expected cost. Here, we considered the expected cost as defined in the MCMP criterion, which also considers the cost spent on traces that do not reach goal states. On the other side, iSSPUDE criterion considers only cost on traces that reach the goal. Presenting MCMP and iSSPUDE expected cost to decision-makers may reach a better-informed trade-off. Instead of considering expectation, we can also

consider a distribution over cost, for example, by considering percentile, such as the one considered in CVaR MDP [4,5]. How to efficiently summarize policies into different aspects of costs and design an efficient preference elicitation process are the next steps of our research.

References

1. Benabbou, N., Leroy, C., Lust, T.: Regret-based elicitation for solving multi-objective knapsack problems with rank-dependent aggregators. In: The 24th European Conference on Artificial Intelligence (ECAI 2020). Saint Jacques de Compostelle, Spain (June 2020). https://hal.sorbonne-universite.fr/hal-02493998
2. Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. Math. Oper. Res. **16**(3), 580–595 (1991)
3. Branke, J., Corrente, S., Greco, S., Gutjahr, W.: Efficient pairwise preference elicitation allowing for indifference. Comput. Oper. Res. **88**, 175–186 (2017)
4. Carpin, S., Chow, Y.L., Pavone, M.: Risk aversion in finite Markov decision processes using total cost criteria and average value at risk. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 335–342. IEEE (2016)
5. Chow, Y., Tamar, A., Mannor, S., Pavone, M.: Risk-sensitive and robust decision-making: a CVaR optimization approach. In: Advances in Neural Information Systems, pp. 1522–1530 (2015)
6. Freire, V., Delgado, K.V.: GUBS: a utility-based semantic for goal-directed Markov decision processes. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pp. 741–749 (2017)
7. Freire, V., Delgado, K.V., Reis, W.A.S.: An exact algorithm to make a trade-off between cost and probability in SSPs. In: Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, pp. 146–154 (2019)
8. Kolobov, A., Mausam, Weld, D.S.: A theory of goal-oriented MDPs with dead ends. In: de Freitas, N., Murphy, K.P. (eds.) Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, pp. 438–447. AUAI Press (2012)
9. Mausam, A.K.: Planning with Markov decision processes: an AI perspective. Synth. Lect. Artif. Intell. Mach. Learn. **6**(1), 1–210 (2012)
10. Regan, K., Boutilier, C.: Robust policy computation in reward-uncertain MDPs using nondominated policies. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, pp. 1127–1133. AAAI Press (2010)
11. Freire da Silva, V., Reali Costa, A.H.: A geometric approach to find nondominated policies to imprecise reward MDPs. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part I. LNCS (LNAI), vol. 6911, pp. 439–454. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23780-5_38
12. Silva, V.F.D., Costa, A.H.R., Lima, P.: Inverse reinforcement learning with evaluation. In: IEEE International Conference on Robotics and Automation (ICRA 2006), pp. 4246–4251. IEEE, Orlando (May 2006)
13. Teichteil-Königsbuch, F.: Stochastic safest and shortest path problems. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2012), pp. 1825–1831 (2012)

14. Teichteil-Königsbuch, F., Vidal, V., Infantes, G.: Extending classical planning heuristics to probabilistic planning with dead-ends. In: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7–11, 2011 (2011)
15. Trevizan, F., Teichteil-Königsbuch, F., Thiébaux, S.: Efficient solutions for stochastic shortest path problems with dead ends. In: Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI) (2017)