Reliable Plan Selection with Quantified Risk-Sensitivity*

Tobias John[1], Mahya Mohammadi Kashani[2], Jeremy Paul Coffelt[3], Einar Broch Johnsen[1], and Andrzej Wąsowski[2]

[1] University of Oslo, Oslo, Norway, tobiajoh@uio.no, einarj@uio.no
[2] IT-University of Copenhagen, Copenhagen, Denmark, mahmo@itu.dk, wasowski@itu.dk
[3] ROSEN Technology and Research Center GmbH, Bremen, Germany
jcoffelt@rosen-group.com

Robots in many domains need to plan and make decisions under uncertainty [1, 4]; for example, autonomous underwater vehicles (AUVs) gathering data in environments inaccessible to humans, need to perform automated task planning [3]. Planning problems are typically solved by risk-neutral optimization maximizing a single objective, such as limited time or energy consumption [2].

A typical probabilistic planner synthesizes a plan to reach the desired goals with a maximum expected reward, given the possible initial states and actions of the world. In this work, we additionally consider risk metrics for selecting solutions to such planning problems. Consider a marine robotics mission scenario where the task is to survey pipeline segments safely based on various risk measurements. During the mission, the AUV needs to choose between two paths P_1 and P_2. A typical probabilistic planner here finds the expected accumulated reward R_2 for P_2 to be greater than the accumulated reward R_1 of P_1, and selects P_2 (see Fig. 1). However, P_1 has a *higher variance*, which means that it is in fact less likely to achieve the expected reward than P_2. The red dashed line depicts a minimum success reward value where the AUV's mission will not fail. Note that the probability of failing (the shaded area) is much higher for P_1 than for P_2, even though P_2 exhibits higher expected reward.

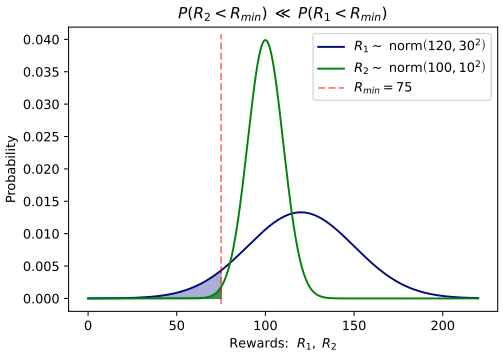

Figure 1: Reward comparison between two plans using risk metric, e.g. variance

We consider planning problems that additionally capture the certainty (or, alternatively, the risk) associated with the candidate solutions and rewards in a semantic form. We use *Probabilistic Programming Domain Definition Language* (PPDDL), and translate its models into *Markov decision processes* (MDPs) [7]. In other words, we translate a planning problem to a new risk-sensitive planning problem, which enables the selection of a plan switch with different risk levels. For example, to produce a risk-sensitive plan for the scenario above, we transform the PPDDL planning problem with a waypoint-following action in which there is a 90% chance that an AUV can move from point A to point B and obtain two reward points, as shown in Fig. 2.

Constructing Risk-Sensitive Plans We model probabilistic systems as MDPs with rewards, i.e., 5-tuples $\mathcal{M} = (S, A, \mathbf{P}, s_0, \mathrm{R})$ where S is a finite set of states, A a finite set of actions, $\mathbf{P} : S \times A \times S \to [0, 1]$ the transition function, $s_0 \in S$ the initial state and $\mathrm{R} : S \mapsto \mathbb{R}_{\geq 0}$ the reward.

Given a plan $\pi : S \mapsto A$, nondeterminism in the MDP can be resolved to obtain the induced *Markov chain (MC)* $\mathcal{M}_\pi = (S, \mathbf{P}_\pi, s_0, \mathrm{R})$ where for each pair of states s, s' the transition

*This work is part of the project REMARO that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956200.

```
(:action waypoint-following
  :parameters (?from ?to)
  :precondition (position ?from)
  :effect (probabilistic 0.9 (and
     (position ?to)
     (not (position ?from))
     (increase (reward) 2)
  ))
)
```

(a) before transformation

```
(:action waypoint-following
  :parameters (?from ?to)
  :precondition (position ?from)
  :effect (probabilistic -0.225 (and
     (position ?to)
     (not (position ?from))
  ))
)
```

(b) after transformation

Figure 2: Transformation at the level of a PPDDL action where $\gamma = 0.5$. Probability of effect of taking action from (a) transformed to *pseudo-probability* in (b).

probability is defined by $\mathbf{P}_\pi(s, s') = \mathbf{P}(s, \pi(s), s')$. A *history* h is a finite sequence of states $h = (s_0, s_1, \ldots, s_n)$. Define a history's *probability* by $\mathbf{P}(h) = \prod_{i=0}^{n-1} \mathbf{P}_\pi(s_i, s_{i+1})$ and its *reward* by $\mathrm{R}(h) = \sum_{i=0}^{n-1} \mathrm{R}(s_i)$. For goal states $G \subseteq S$, the corresponding reward of the Markov Chain, $\mathrm{R}_\mathcal{M}(G)$, is a discrete random variable with:

$$\Pr\Big(\mathrm{R}_\mathcal{M}(G) = x\Big) = \sum_{\substack{h=(s_0,\ldots,s_n) \\ s_n \in G; s_0,\ldots,s_{n-1} \notin G \\ \mathrm{R}(h)=x}} \mathbf{P}(h)$$

To find a plan π that best balances expected reward and associated risk for a given MDP $\mathcal{M}$ with goal states G, we first generate a set of different *candidate plans*. These candidates trade some of the expected reward for a lower risk. Our framework is agnostic to the algorithm to compute the different candidates.

In this paper, we generate candidate plans using Koenig and Simmons' approach [5] to generate risk-averse plans with a non-linear utility function for cumulative rewards. This function values a difference between high rewards less than the same difference between smaller rewards; i.e., the cumulative reward of a few "best-case" paths has smaller impact on plan selection than many paths with a decent cumulative reward. The utility function depends on a parameter γ, which balances expected reward and involved risk. The main advantage of this construction is that we obtain a utility function for the cumulative reward for risk-sensitive planning by only making local changes; i.e., the probabilities of the MDP's transitions are replaced by values that take the probability, immediate reward and the risk sensitivity into account.

Definition 1 (Transformed Transition System)**.** *Given an MDP $\mathcal{M} = (S, A, \mathbf{P}, s_0, R)$ and a parameter γ with $0 < \gamma < 1$, we define the transformed transition system $\mathcal{M}^\gamma = (S, A, \mathbf{P}^\gamma, s_0)$ where $\mathbf{P}^\gamma : S \times A \times S \to [-1, 0]$ with*

$$\mathbf{P}^\gamma(s, a, s') = \mathbf{P}(s, a, s') \cdot \left(-\gamma^{R(s)}\right)$$

Although this transition system is not an MDP, standard algorithms can be used to find the plan that maximizes the "pseudo-probability" to reach a goal state [5]. This construction is well-suited for our setting, as we consider PPDDL domains to describe the MDP and the transformation described in Def. 1 can be done at the level of the actions in these domains.

We can now iterate over different values of γ and call a generic PPDDL planner to generate the optimal plan for each value. This "optimal plan" is simply the plan that maximizes the probability of reaching one of the goal states. For different values of γ, we get different plans, which form our set of candidates.

A crucial point of our approach is that plans are always generated w.r.t. a limited number of risk metrics, as planners can only handle a limited number of objectives at a time. In our specific case, only the parameter γ is considered. This risk measurement might—but need not—correlate to other risk metrics, e.g., variance or entropy of the cumulative reward. Therefore, we propose different methods to asses candidate plans across *all* risk measurements of interest.

Plan Selection To select the best plan, we use metrics based on the probability distribution of the reward $\mathrm{R}_{\mathcal{M}_\pi}(G)$ to evaluate each of the candidate plans. One approach for this is to utilize Monte Carlo simulation in the MDP $\mathcal{M}$. However, this is insufficient for autonomous underwater robots since formalizing an MDP model $\mathcal{M}$ for robotic behaviour always involves simplifying assumptions of their physical environment. For AUVs, this would require neglecting critical considerations such as unpredictable water currents, GPS-denied vehicle localization or noisy acoustic sensors. In order to overcome such limitations, we use an *underwater physics simulator* that also provides increased realism. The following are metrics we consider:

- The *expected reward* $\mathbb{E}[D']$ is typically a primary concern when selecting a plan. Even in risk-averse settings, a decent expected reward is required.
- The *variance* of the reward $\mathbb{E}[D' - \mathbb{E}[D']^2]$ is an often used measurement for risk [6]. The higher the variance, the more risk is involved.
- The *entropy* of the reward $-\sum_{x \in R} \Pr(D' = x) \log_2(\Pr(D' = x))$ is another common metric used to measure the uncertainty or "surprise" or a reward. In general, a generalization of entropy to continues variables might be needed.
- The *reward-bounded probability* $\Pr(D' \leq b)$ is the probability that the reward falls below a given bound b. This metric allows us to estimate how often bad runs occur.

Depending on the specific application, these metrics might differ on importance. By considering a variety of risk metrics, practitioners can obtain a more balanced and informed assessment of the risk involved in the selected plan.

References

[1] G. Canal, M. Cashmore, S. Krivic, G. Alenyà, D. Magazzeni, and C. Torras. Probabilistic planning for robotics with ROSPlan. In *Proc. 20th Conf. Towards Autonomous Robotic Systems (TAROS 2019)*, volume 11649 of *LNCS*, pages 236–250. Springer, 2019.

[2] M. Cashmore, M. Fox, D. Long, D. Magazzeni, and B. Ridder. Opportunistic planning in autonomous underwater missions. *IEEE Trans. Autom. Science and Engineering*, 15(2), 2017.

[3] X. Chen, N. Bose, M. Brito, F. Khan, B. Thanyamanta, and T. Zou. A review of risk analysis research for the operations of autonomous underwater vehicles. *Reliability Engineering & System Safety*, 216:108011, 2021.

[4] M. Hinostroza and A. M. Lekkas. A rudimentary mission planning system for marine autonomous surface ships. *IFAC-PapersOnLine*, 55(31):196–203, 2022.

[5] S. Koenig and R. G. Simmons. How to make probabilistic planners risk-sensitive (without altering anything). In *Proc. 2nd Intl. Conf. on Artificial Intelligence Planning Systems*. AAAI, 1994.

[6] M. Sato, H. Kimura, and S. Kobayashi. TD algorithm for the variance of return and mean-variance reinforcement learning. *Trans. Jap. Soc. for Artificial Intelligence*, 16(3):353–362, 2001.

[7] H. L. Younes and M. L. Littman. PPDDL 1.0: An extension to PDDL for expressing planning domains with probabilistic effects. 2004.