Inverse Reward Design

Dylan Hadfield-Menell **Smitha Milli** **Pieter Abbeel*** **Stuart Russell** **Anca Dragan**
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, CA 94709
{dhm, smilli, pabbeel, russell, anca}@cs.berkeley.edu

Inverse Reward Design

- Learning and Exploration

Zilun Peng
Huiwen You

Introduction

- When human designs a reward function, (known as **proxy reward function**), they try to capture as much about the world as possible.
- Inevitably, agent may encounter new states, leading to unexpected behaviour (known as **negative side effect**).
- To avoid failure, agent should act in a **risk-averse** manner when it recognizes new scenario.
- It should take **proxy reward function as an guess** at what the **true reward function** is and assign uncertainty estimates to the rewards generated by our reward function.
- **IRD**: A problem of **finding the distribution of true reward functions** given the designed reward function and the designed environment.

Approach

- compute IRD Posterior - posterior distribution over the optimal reward function (i.e. computing the robot's uncertainty about reward evaluations.)

$$P(w = w^* | \tilde{w}, \tilde{M}) \propto \frac{\exp\left(\beta w^\top \tilde{\phi}\right)}{\tilde{Z}(w)} P(w), \tilde{Z}(w) = \int_{\tilde{w}} \exp\left(\beta w^\top \tilde{\phi}\right) d\tilde{w}.$$

$$\tilde{\phi} \;=\; \mathbb{E}[\phi(\xi) | \xi \sim \pi(\xi | \tilde{w}, \tilde{M})] \qquad \hat{Z}(w) = w^\top \phi_w + \sum_{i=0}^{N-1} \exp\left(\beta w^\top \phi_i\right)$$

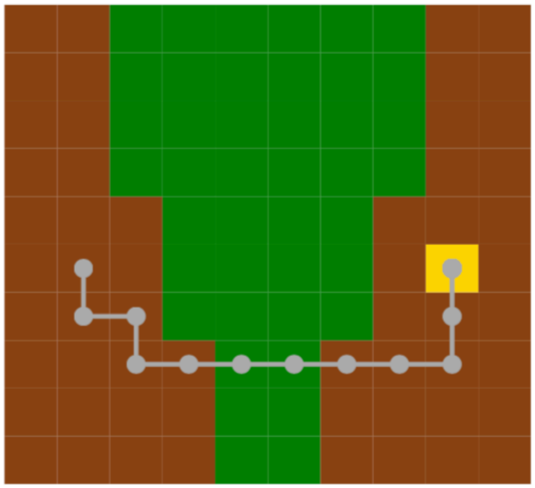
Sample-Z: sample to approximate the normalizing constant
inspired by methods in approximate Bayesian computation (Sunnåker et al., 2013)

- risk-averse planning - make use of the uncertainty to compute optimal trajectory

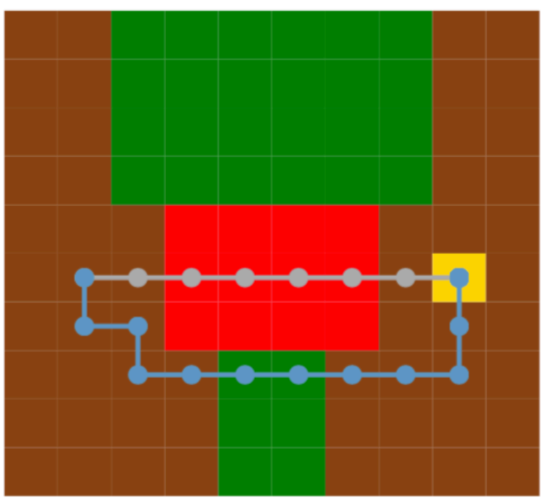
$$\xi^* = \underset{\xi}{\operatorname{argmax}} \min_{w \in \{w_i\}} w^\top \phi(\xi)$$

Trajectory optimization via linear programming approach (Syed et al. (2008).)

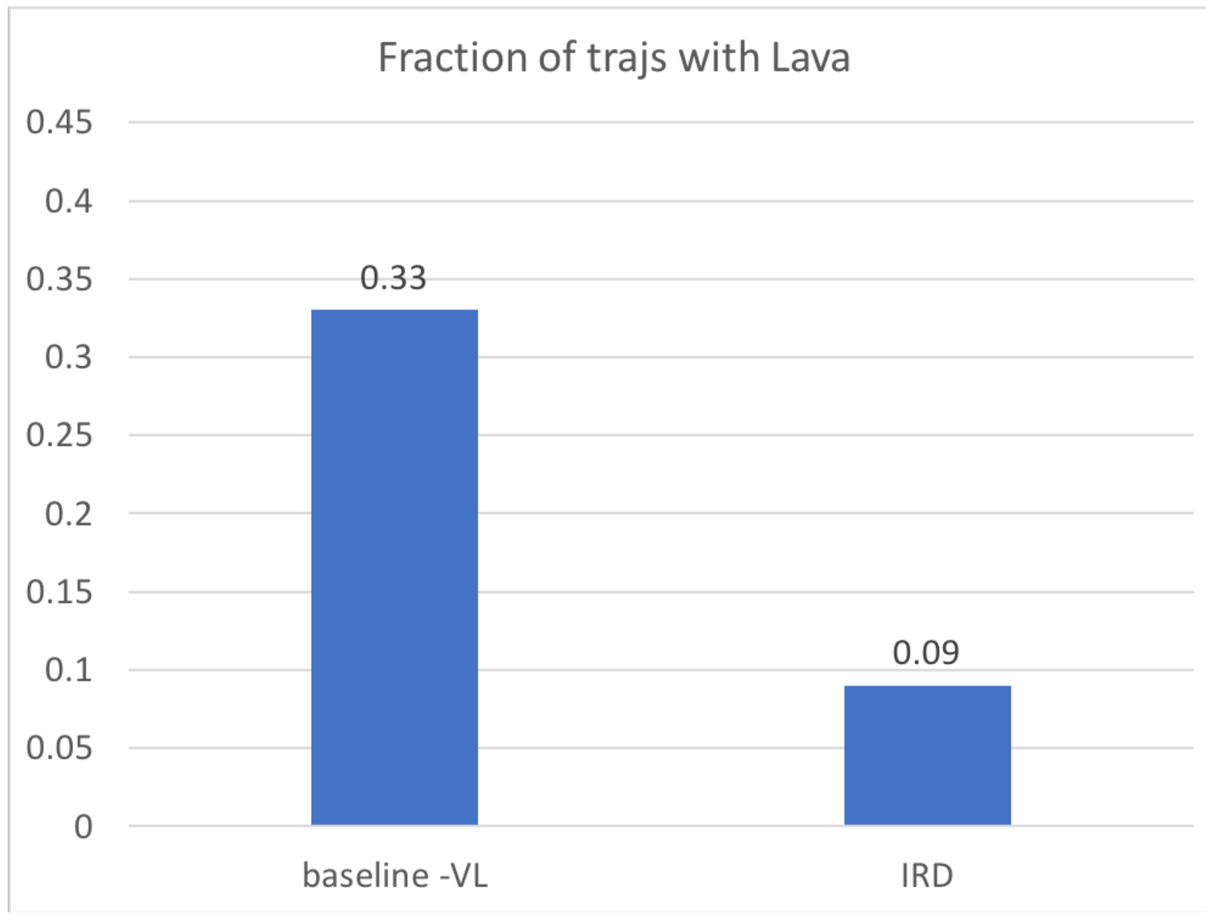
Result



training (expected) env.



testing (un-observed) env.



Conclusion

- Correctness?
- Generalizability?
  - Risk-averse planning avoids good risk.
  - Problem scope does not scale. Impossible to solve the planning problem with complicate mdp and reward function.
  - In the paper, the proposed solution is based on the assumption that reward function is linear. (e.g. what if the feature space is not discrete terrain value but ground colour.)
  - Proxy reward function has to be “good” to make IRD work.