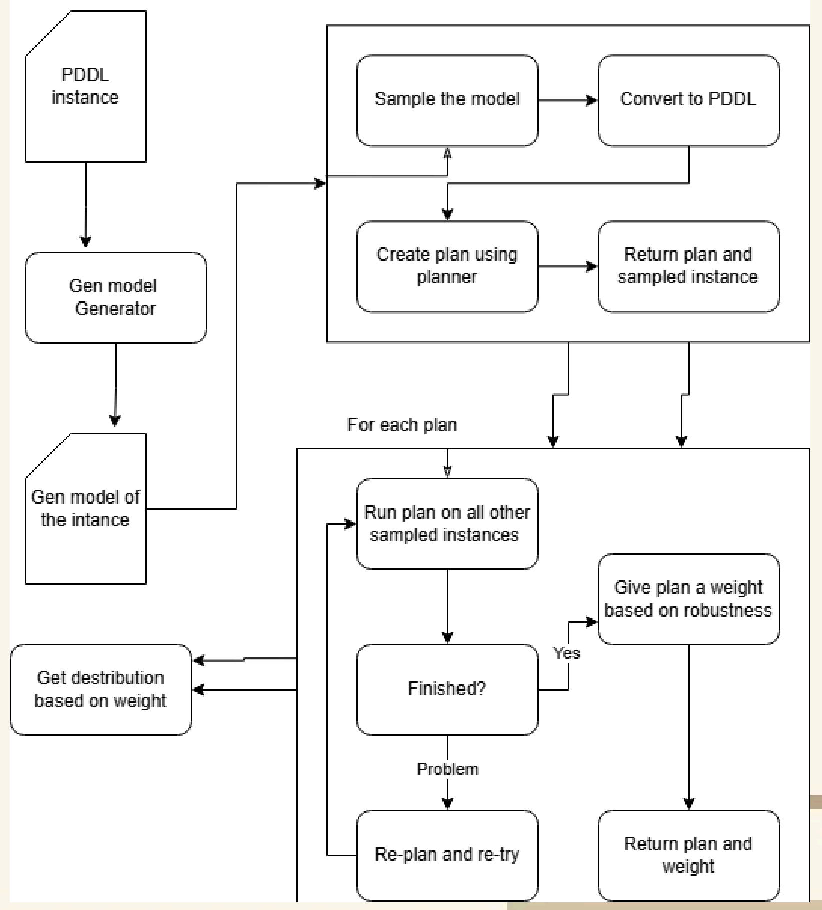
Robust Planning as Probabilistic Inference

Author: Matthijs Bonke (M.S.Bonke@student.tudelft.nl
Supervisor: Issa Hanou, Reuben Gardos Reid

I. Introduction

- In today's busy world, effective planning is crucial - but real-life plans often fail due to unexpected events. Unlike idealized, deterministic environments, the real world introduces uncertainty, where missing a single step can cause an entire plan to break down.
- Many tasks depend on completing earlier steps - for example, you can't make planks without collecting wood. As more elements are added, planning becomes exponentially harder. Uncertainties like missing resources or changing environments make robust planning essential.
- Previous research has addressed uncertainty by modifying the planning algorithm itself. However, these tailored solutions often don't transfer well to other problems.
- This research explores how to build robust plans on top of existing planners, without altering their core logic. We investigate this using the Minecraft domain from the PDDL Gym, focusing on resilience in probabilistic settings.



II. Research question

How can we obtain robust plans for a Planning Domain Definition Language (PDDL) with probabilistic inference without changing the underlying scheduling algorithm?

III. Possible solutions

To get accurate plans for a domain with interference we need a accurate model including these uncertainties.
One possible way to create a distribution of plans is by sampling the model and using the these slightly different domains to create plans for each scenario. With each plan it is possible to run it on the different domains to evaluate on how robust a certain plan is compared to other plans.
There are many ways a distribution of plans can be created; Rejection sampling, Importance sampling, Markov chain Monte Carlo....

IV. The experiment

- Setup

For the experiment a detailed model of the domain from the (PDDL) gym library, In this case a model of the Minecraft planner, is required. For the modelling of this domain with inference, the Gen modelling software is used. For the real-life uncertainty blocks missing has been choosen.

For this experiment, the settings of creating plans has been altered between experimental runs. The following configurations where used:

1. No re-planning. 2. Re-planning

- Methods

Data has been created by running the algorithm on all the domains predifined in the PDDL gym library. Creating a distrubution of plans for each domain seperately.

For each domain a model was created with Gen, from this model plenty of domains are sampled. For each sampled domain a plan is created. We run these plans against all other domains.

- Observation

Each Distrubution of plans will be evaluated based on the following criteria: 1. Completeness percentage. 2. Efficient path. 3. re-planning count. Based on this each plan will be given a weight.

V. The results

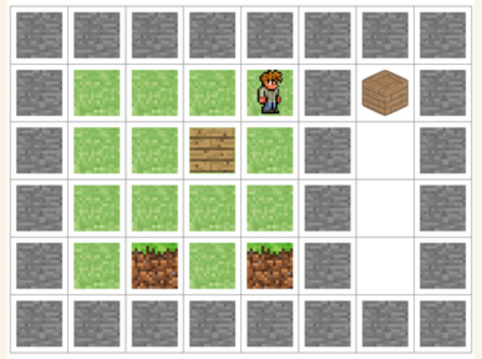
- Robust plan selection improves success
  - Achieved 92% success on unseen test worlds
  - Outperformed efficiency-only (70.1%) and random plans (65%)
- Efficiency-only plans are fragile
  - Shorter but prone to failure when world structure changes
- Replanning boosts performance further
  - Success rate increased to 94.2%
  - Failure rate dropped to 5.8%

Evaluating plans across sampled environments + enabling dynamic replanning = greater reliability in uncertain, partially observable domains...

Strategy	Success Rate (%)	Failure Rate (%)	Avg. Plan Length
Random Selection	65.0	35.0	12.3
Efficiency-Only (weight 0.9)	70.1	29.9	**10.1**
Robust Selection	**92.0**	8.0	13.5
Robust + Replanning	**94.2**	**5.8**	13.2

VI. Conclusion

- Can we select robust plans from a distribution of sampled plans?
  Yes. Weighted selection consistently chose plans that succeeded in 92% of test cases, outperforming random (65%) and efficiency-based plans.
- Does replanning improve robustness and efficiency?
  Yes. Replanning reduced failure rates from 28% to 8%, with slight efficiency gains.
- Does emphasizing efficiency reduce robustness?
  Yes. Plans optimized heavily for efficiency (weight ≥ 0.7) were shorter but more fragile, often failing when conditions changed.
- Do robust plans work in completely new environments?
  Partially. Robust plans performed better than baselines, but still showed a ~5% drop in success on unseen worlds.



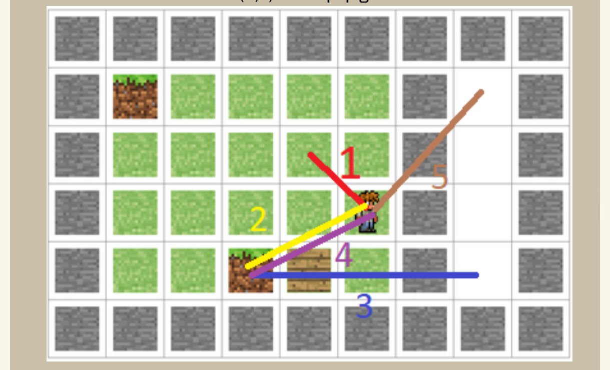


figure 1: path of domain 2

VII. Discussion

- Robust plans outperform baselines
  Plans selected for durability succeed more often in uncertain environments than those chosen randomly or purely for efficiency.
- Efficiency vs. robustness is a trade-off
  Heavily weighting efficiency can lead to shorter, but fragile plans. These fail more often when unexpected changes occur.
- Limited by non-adaptive planners
  The robust selection process can only choose from a fixed set of plans, reducing flexibility and limiting performance in diverse environments.
- Room for future improvement
  - Use adaptive or generative planners to increase plan diversity and robustness.
- Incorporate probabilistic inference methods (e.g., MCMC) to guide planning under uncertainty more effectively.