Computing High-Quality Solutions to Probabilistic Planning Problems

Alberto Camacho [1] Christian Muise [2] Sheila A. McIlraith [1]

[1]Department of Computer Science
University of Toronto, Canada
{acamacho,sheila}@cs.toronto.edu

[2]Department of Computing and Information Systems
University of Melbourne, Australia
christian.muise@unimelb.edu.au

Planning in Artificial Intelligence is the problem of finding action strategies, typically executed by intelligent software agents, autonomous robots, or unmanned vehicles. Given a description of the world, a set of actions that an agent is capable of performing, and some desired goal or objective, the solution to a planning problem is a policy – a mapping from states to actions – that the agent can execute to achieve the goal. There is often uncertainty in the task either because the world is not modeled precisely, or because action outcomes are intrinsically uncertain or outside the control of the agent. For example, a slippery road or strong wind can affect the movement of a vehicle. In this work we address the class of Probabilistic Planning problems, where the outcomes of the actions are non-deterministic and follow a probabilistic transition model. In particular, we focus on finding policies that maximize the probability of reaching a prescribed goal. Our algorithm, Prob-PRP, outperforms the state of the art, computing substantially more robust policies orders of magnitude faster than the state of the art.

Motivation

Objective: To synthesize controllers for software agents and devices that can handle uncertainty in the world, provide compact solutions, and are able to provide guarantees with respect to their behaviour.

Existing solutions to Probabilistic Planning problems:

- Offline planners cannot handle large problems
- Online planners:
  - no guarantees of optimality
  - poor mechanism to avoid deadends
  - large solutions

Approach: Exploit (our) state-of-the-art techniques for *F*ully *O*bservable *N*on-*D*eterministic (FOND) planning:

- better scalability
- compact solutions

The Problem

Given: a description of the **initial state** of the world, a set of **actions**, and a **goal** or objective

Compute: a **policy** — a mapping from states to actions that the agent can execute to achieve the goal.

A sequence of executable actions, π, is called a **plan**.

FOND planning: non-deterministic action outcomes

Probabilistic planning: stochastic action outcomes

Quality of Solution

1. goal achievement criterion
   - FOND planning: strong or strong cyclic plan
   - Probabilistic planning: maximize probability of reaching the goal (MAXPROB)
2. compact policy
3. short plans

PRP: Core FOND Algorithm

```
Policy ← ∅
while Policy changes do
    Open ← {Init}
    Seen ← {}
    while Open ≠ ∅ do
        s ← Open.pop()
        if s ⊭ Goal ∧ s ∉ Seen then
            Seen.add(s)
            if Policy(s) is undefined then
                [a_1 ··· a_n] ← GenerateWeakPlan(s, Goal)
                φ ← Goal
                for i = n, ..., 1 do
                    φ = Regress(φ, a_i)
                    Policy ← Policy ∪ {⟨φ, a_i⟩}
            if s not a deadend then
                for s′ ∈ Prog(s, Policy(s)) do
                    Open.add(s′)
    ProcessDeadends()
return Policy
```

✓ Forbidden State-Action pairs
✓ Compact state representation
✓ Compact solutions

PRP → Prob-PRP

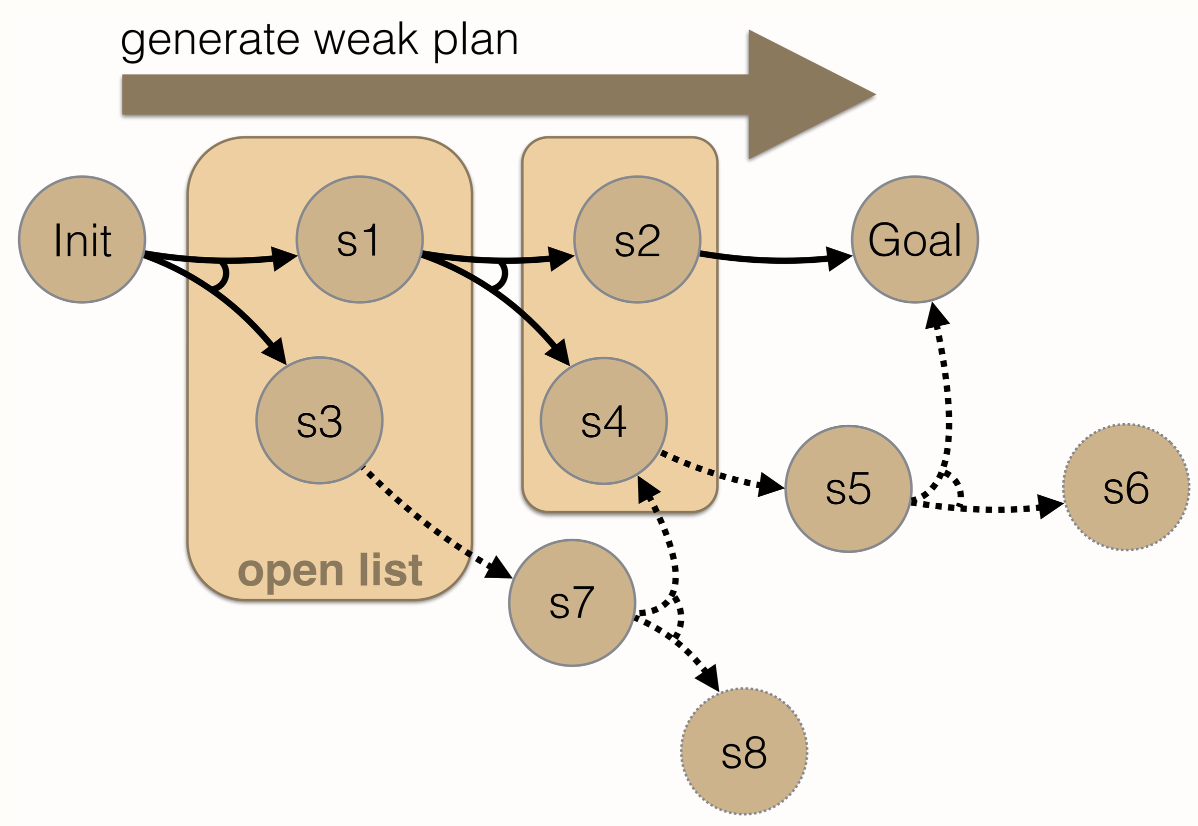
Prob-PRP extends PRP to obtain high-quality MAXPROB solutions to probabilistic problems.

Exploring most likely plans

- **Likelihood** of a plan π: $L(\pi) = \Pi_{i=0}^{n} \Pr(s_i, a_i, s_{i+1})$
- GenerateWeakPlan modified to give **preference** to exploring the **most likely plans**

Maximizing Reachability

- **Best quality** policy P found so far is selected
- All states reachable by P are **fully explored**



Problem	RFF %	RFF L	RFF S	RFF T	Prob-PRP %	Prob-PRP L	Prob-PRP S	Prob-PRP T
blocksworld-p01	**100**	23	18	0,02	**100**	**19**	**17**	**0,00**
blocksworld-p03	**100**	23	18	0,02	**100**	**19**	**17**	**0,00**
blocksworld-p05	**100**	65	61	0,72	**100**	**47**	**43**	**0,16**
blocksworld-p07	**100**	64	61	0,69	**100**	**47**	**43**	**0,16**
blocksworld-p09	**100**	**41**	**38**	0,67	**100**	65	61	**0,46**
blocksworld-p11	**100**	**42**	**39**	0,66	**100**	65	61	**0,46**
blocksworld-p13	0	0	117	17	**100**	**115**	**107**	**1,38**
blocksworld-p15	0	0	117	17	**100**	**115**	**107**	**1,38**
boxworld-p01	**100**	**29**	**50**	0,43	**100**	32	57	**0,06**
boxworld-p03	**100**	**29**	**48**	0,38	**100**	32	57	**0,06**
boxworld-p05	**100**	**39**	**81**	1,77	**100**	**39**	105	**0,24**
boxworld-p07	**100**	**65**	**160**	13,0	**100**	69	266	**2,32**
boxworld-p09	**100**	65	**132**	7,56	**100**	**63**	207	**1,84**
boxworld-p11	**100**	**73**	**183**	22,2	**100**	102	415	**17,9**
boxworld-p13	0	0	344	36	**100**	**178**	906	130
boxworld-p15	0	0	347	35	**100**	**178**	906	160
ex-blocksworld-p02	28	12	37	0,11	**54**	**10**	**15**	**0,02**
ex-blocksworld-p04	52	14	49	0,09	**59**	21	**18**	**0,06**
ex-blocksworld-p06	90	13	62	0,10	**96**	22	28	0,34
ex-blocksworld-p08	7	24	69	0,64	**36**	**18**	**32**	**0,38**
ex-blocksworld-p10	2	36	77	0,97	**3,1**	**26**	105	14,3
ex-blocksworld-p12	1	38	97	2,15	**2,1**	**17**	**78**	6,28
schedule-p02	**100**	59	**5**	**0,01**	**100**	**48**	7	0,04
schedule-p03	**100**	100	**5**	**0,01**	**100**	**87**	7	0,12
schedule-p04	96	58	**14**	**0,02**	**100**	**46**	21	0,14
schedule-p05	89	116	**14**	**0,03**	**100**	**95**	16	0,18
schedule-p06	**45**	**364**	**141**	**1,42**	0	–	–	–
triangle-tire-p02	**100**	13	81	0,17	**100**	**12**	**23**	**0,00**
triangle-tire-p04	**100**	30	248	1,76	**100**	**25**	**55**	**0,06**
triangle-tire-p06	**100**	46	490	7,98	**100**	**39**	**95**	**0,22**
triangle-tire-p08	**100**	62	958	36,5	**100**	**52**	**143**	**0,72**
triangle-tire-p10	**100**	78	1595	111	**100**	**65**	**199**	**2,38**

Table: Successful runs (%), expected plan length (L), policy size (S), and computation time (T) for previous state-of-the-art MAXPROB planner, RobustFF (Teichteil-Konigsbuch 2010), and Prob-PRP. Bold numbers indicate superior performance. Dash (–) indicates the planner exceeded the 2GB memory limit during computation.

High-Quality Solutions

Prob-PRP finds high-quality solutions that outperform the state of the art.

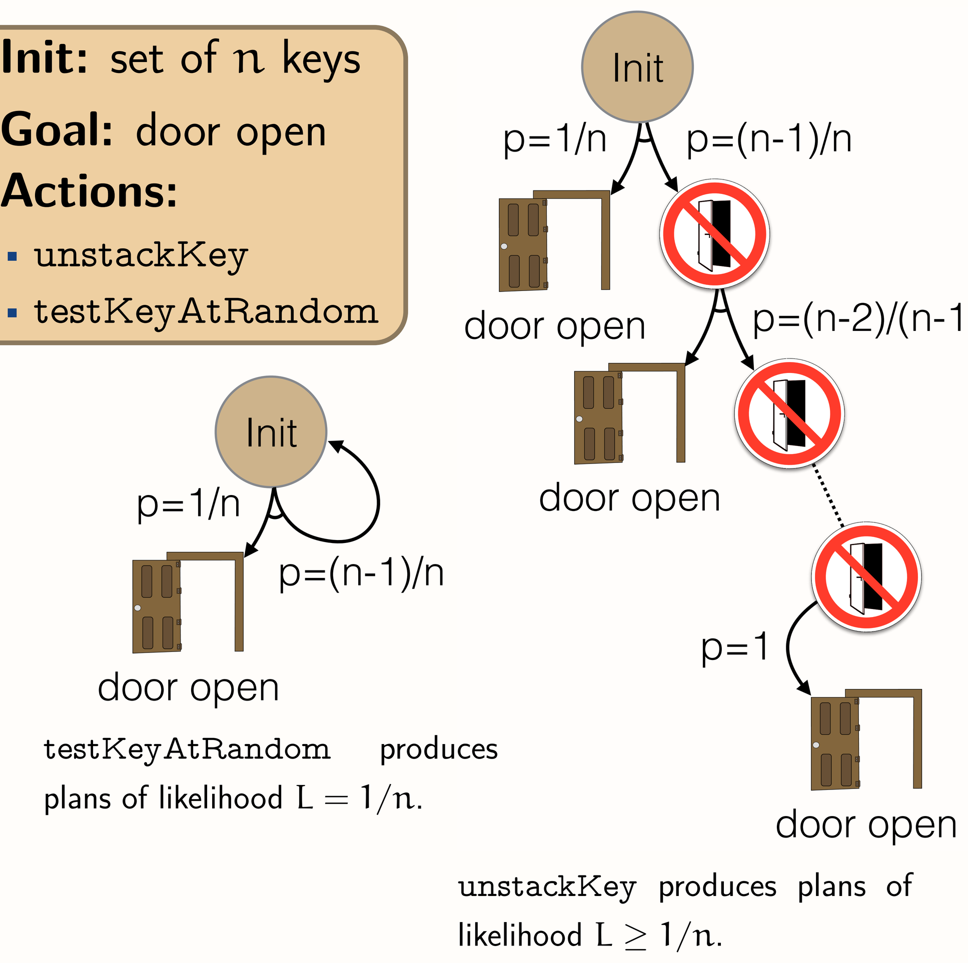
MAXPROB Prob-PRP is **guaranteed** to find optimal MAXPROB solutions when deadends are avoidable.

Compact Policy Prob-PRP inherits the **compact representation** of the policies from PRP, leading to **small policies**.

Short Plans Prob-PRP solutions do not rely on highly improbable events to reach the goal.

Example

- **Init:** set of n keys
- **Goal:** door open
- **Actions:**
  - unstackKey
  - testKeyAtRandom



Potential Applications

Summary

- Introduced Prob-PRP: a planner capable of finding high-quality MAXPROB solutions offline.
- Identified properties of high-quality solutions.
- Prob-PRP improves the previous state of the art in Probabilistic Planning in terms of performance and quality of the solutions.

References

[1] **From FOND to Probabilistic Planning: Computing policies that sidestep avoidable deadends** Camacho, A.; Muise, C.; and McIlraith, S. A. Submission under review. Preliminary version appearing in the Workshop on Heuristics and Search for Domain-independent Planning (HSDIP). ICAPS, 2015.

[2] **Improved Non-deterministic Planning by Exploiting State Relevance** Muise, C.; McIlraith, S. A.; and Beck, Christopher, J. In the 22nd International Conference on Automated Planning and Scheduling, 2012.