Frontier-Based RTDP: A New Approach to Solving the Robotic Adversarial Coverage Problem

Roi Yehoshua, Noa Agmon and Gal A. Kaminka

Presenters: Chan Min Park, Rachel Prendergass, Olivia Schlepphorst

The Problem

Area coverage problem

- Example: Roomba

Adversarial coverage

- Example: navigating a minefield

Environment: Grid world

P(S)=0.2	P(S)=0.1	P(S)=0.2	P(S)=0.2
P(S)=0.2		P(S)=0.2	P(S)=0.2
start	P(S)=0	P(S)=0.6	P(S)=0.5

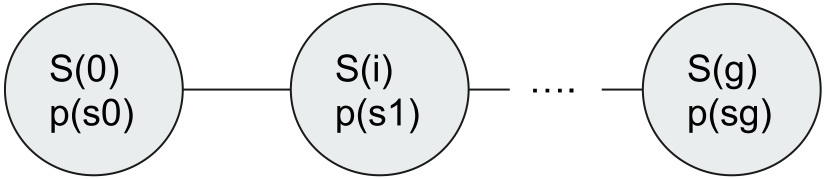
Zhang, Jeremy. “Reinforcement Learning-Implement Grid World From Scratch.” *Medium*, Towards Data Science, 24 Aug. 2019, towardsdatascience.com/reinforcement-learning-implement-grid-world-from-scratch-c5963765ebff.

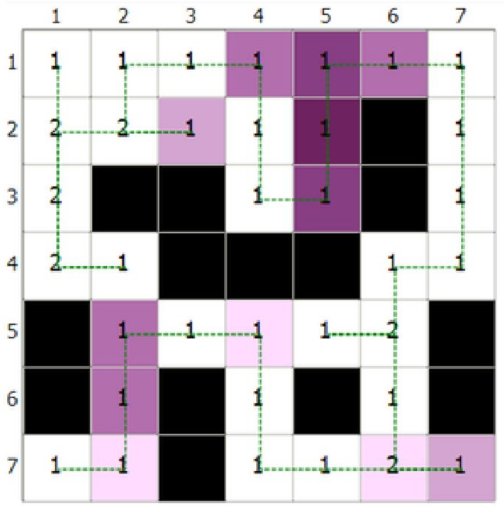
- Grid contains the probability of the robot being stopped by a threat at any point.
- **Offline version:** robot knows the map beforehand and can plan ahead

The Approach: Markov Decision Process (MDP)

- Represent costs and uncertainty of actions
- Stochastic Shortest Path Problem
- Possible state space size requires redefining the search algorithm to solve the MDP
- Introduction of the frontier

Adversarial Coverage





Independance: Each (S,S`) transition is independent

Given Path A:

P(completion):

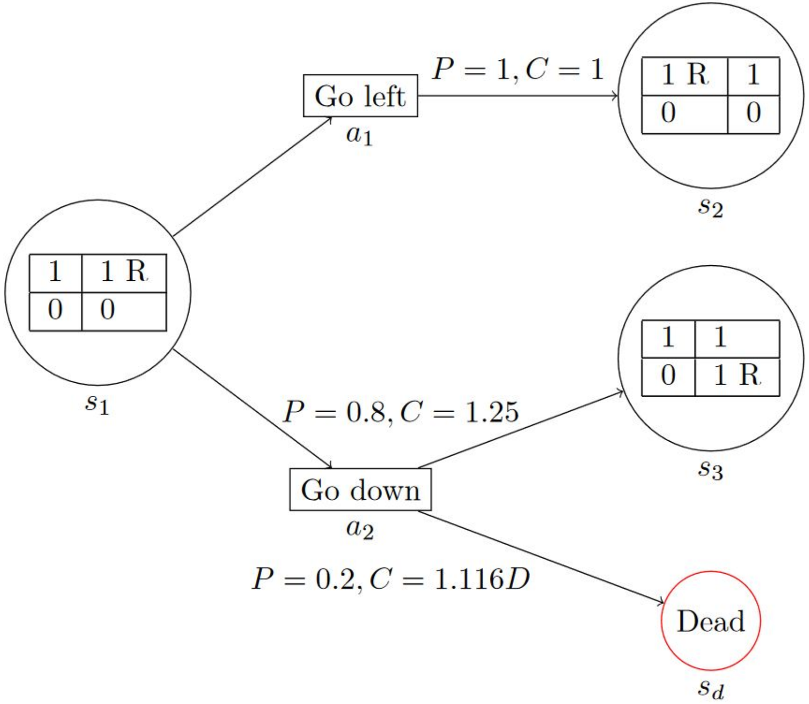
$$P(S_A) = \prod_{i \in (a_1, \ldots, a_m)} (1 - p_i)$$

Cost Function:

$$f(A) = -\alpha \cdot P(S_A) + \beta \cdot |A|$$

α: Risk , β: Coverage time

Coverage expressed as an MDP



State: Gridworld coverage and Robot Position

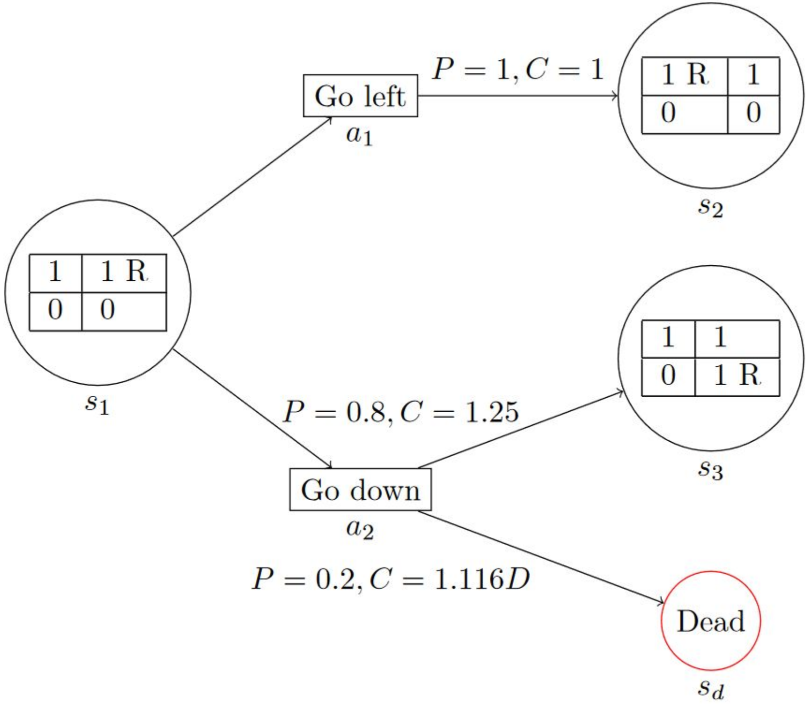
Expansion: State Actions:

- State expansion shows possible transitions
- Action expansion shows possible results

Goal State: All cells in coverage state are 1

Termination State: Goal State or Dead

Transition Function



Transition probability success:

$$P_a(s'|s) = 1$$

Transition probability fail:

$$P_a(s'|s) = 1 - p_j$$

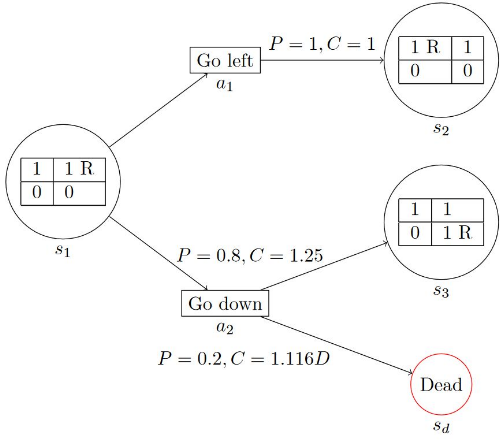
Cost Success:

$$C_a(s, s') = \frac{1}{1-p_j}$$

Cost Fail:

$$C_a(s, s_d) = -D \cdot \frac{\log(1-p_j)}{p_j}$$

Transition Function



Discount & Risk Ratio:

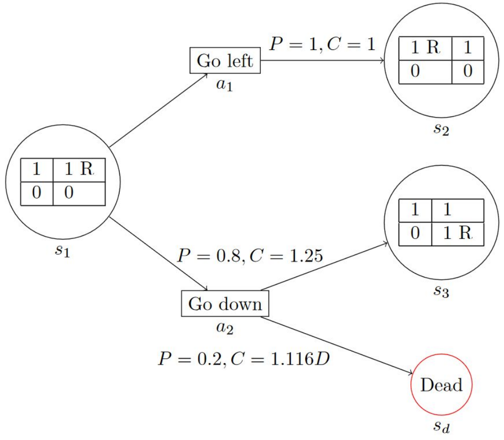
$$D = -\frac{\alpha}{\beta} \cdot \frac{1}{\log(1 - p_{min})}$$

α: Risk , β: Coverage time

Expected cost of state:

$$E[C(c_j)] = (1 - p_j) \cdot \frac{1}{1 - p_j} + p_j \cdot \left[-D \cdot \frac{\log(1 - p_j)}{p_j} \right]$$
$$= 1 - D \log(1 - p_j)$$

Minimum Path Cost and Limitation



Minimum Path Cost:

$$\sum_{t=0}^{n-1} E[C_{a_t}(s_t, s_{t+1})] = n - D\sum_{j=1}^{n} \log(1 - p_j)$$

Limitation:

- Searching through possible states is a NP-complete problem!
- Processing and memory limitations

Real-Time Dynamic Programming

- Heuristic-search DP algorithm

- Benefits
  - Is focused
  - Has a good anytime behavior

- Repeated trials
  - Initial state s0
  - Ends in goal state or dead-end state
  - Action selection → greedy
  - Outcome selection → randomly determined
  - Coverage path is built

Frontier Based Real-Time Dynamic Programming

- RTDP wastes time
- Frontier states = separate covered regions from uncovered
- Path with minimum expected cost
- Composite action/outcomes
- Dijkstra's algorithm → shortest path

Algorithm - FBRDTP

Algorithm 2 Frontier_Based_RTDP

Data structures: *frontier* - set of frontier states
visited - set of states already visited by the current trial

function FBRTDP(s_0) // *s_0 is the initial state*
  while $\max_{s \in visited}$ RESIDUAL(s) $> \epsilon$ **do**
    $visited \leftarrow \{s_0\}$
    $frontier \leftarrow$ all successors of s_0
    FBRTDPTRIAL(s_0)

function FBRTDPTRIAL(s) // *Execute one trial*
  while not GOAL(s) and $s \neq s_d$ **do**
    // *Pick best composite action and update hash*
    $\hat{a} \leftarrow$ GREEDYCOMPOSITEACTION(s)
    UPDATE($s, \hat{a}$)
    // *Stochastically simulate next state*
    $s \leftarrow$ CHOOSENEXTSTATE($\hat{a}$)
    if $s \notin visited$ **then**
      Add s to *visited*
      UPDATEFRONTIER(s)

function GREEDYCOMPOSITEACTION(s)
  Build a graph G that consists of the states in $visited \cup frontier$ and its edge weights defined as the expected costs of the state transitions
  Run Dijkstra on the graph G starting from s
  Find a frontier f with minimum cost path from s
  Let $\hat{a} = (a_1, ..., a_n)$ be the sequence of actions leading from s to f on the minimum cost path
  return $\hat{a}$

function QVALUE($s, \hat{a}$)
  return $\sum_{s' \in S} P(s'|s, \hat{a})[C_{\hat{a}}(s, s') + s'.V]$

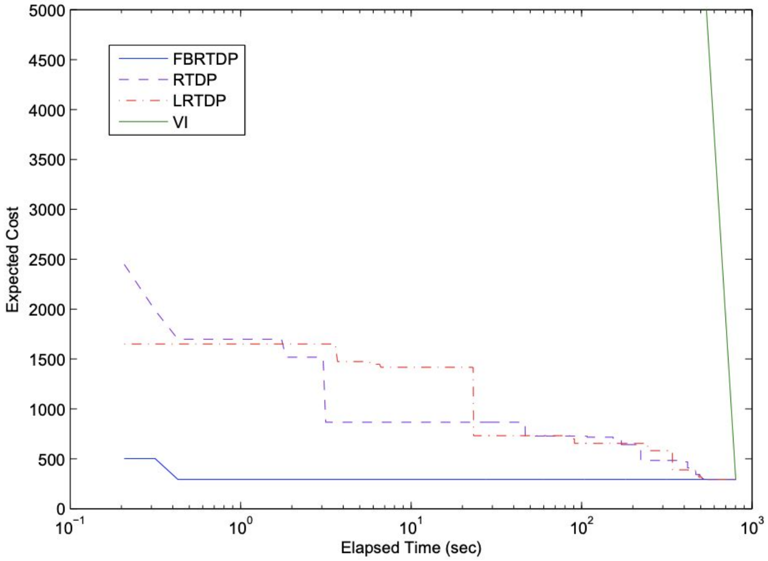
function UPDATE($s, \hat{a}$)
  $s.V \leftarrow$ QVALUE($s, \hat{a}$)

function CHOOSENEXTSTATE($s, \hat{a}$)
  Choose s' with probability $P(s'|s, \hat{a})$
  return s'

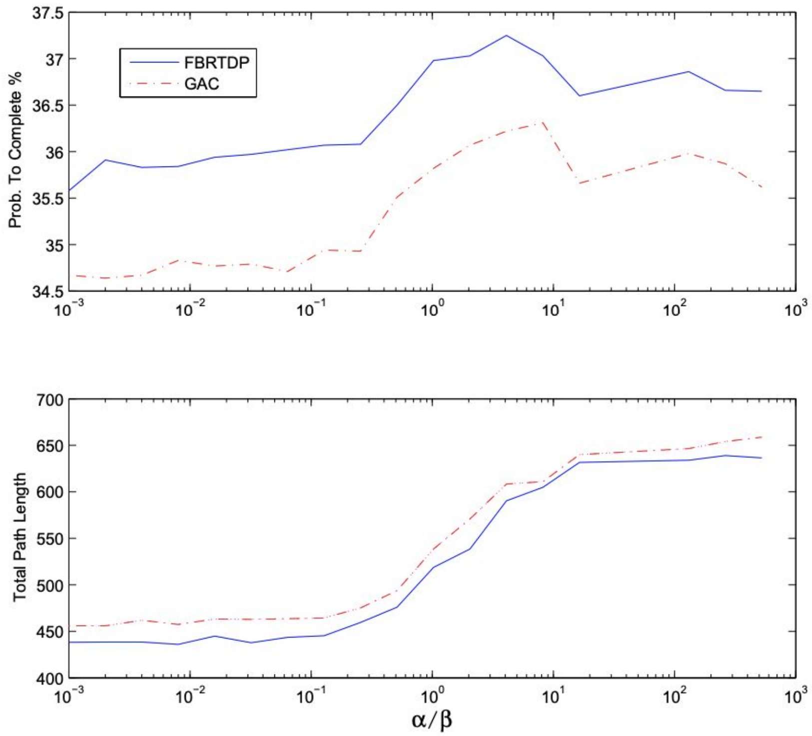
function UPDATEFRONTIER(s)
  Remove s from *frontier*
  for every successor state s' of s **do**
    if $s' \notin visited$ and $s' \notin frontier$ **then**
      Add s' to *frontier*

Empirical Evaluation

- Comparison of FBRTDP to:
  - RTDP
  - Labeled RTDP (LRTDP)
  - Value Iteration (VI)
  - Greedy Adversarial Coverage (GAC)



Larger Maps Comparison



Conclusion

- Model adversarial coverage problem as an MDP for robots
- Smaller maps vs. larger maps
- MDPs
- FBRTDP
  - Provides efficiency while having same optimal convergence as RTDP
  - Allows RTDP to solve the problem on larger maps
- In the future:
  - Handle different variants of the adversarial coverage problem
    - I.e. time delay for robot rather than full stop
    - evaluate FBRTDP on other planning problems and compare its performance to other heuristic algorithms for solving MDPs

Works Cited

[1] Roi Yehoshua, Noa Agmon and Gal A. Kaminka. 2015. Frontier-Based RTDP: A New Approach to Solving the Robotic Adversarial Coverage Problem. Retrieved March 2020 from https://www.researchgate.net/profile/Roi_Yehoshua/publication/281176286_Frontier-Based_RTDP_A_New_Approach_to_Solving_the_Robotic_Adversarial_Coverage_Problem/links/55d9b64208aec156b9ac4ae4.pdf.