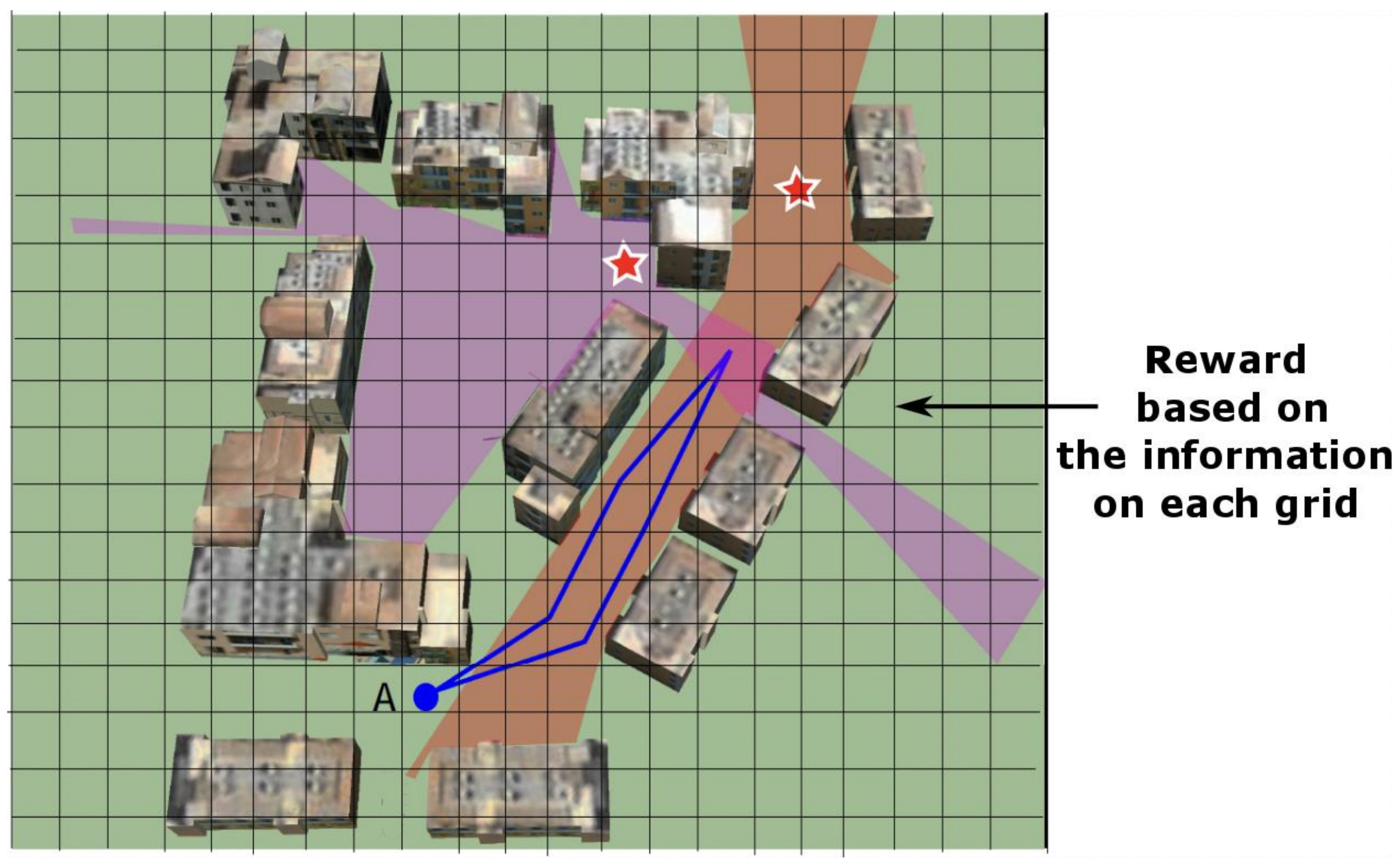
Tree Search Techniques for Minimizing Detectability and Maximizing Visibility

Z. Zhang[1], J. Lee[2], J. Smereka[2], Y. Sung[1], L. Zhou[1], and P. Tokekar[1]
Virginia Tech[1], USA; GVSC[2], USA

This research was supported in part by the Automotive Research Center (ARC) at the University of Michigan, with funding and support by the Department of Defense under Contract No. W56HZV-14-2-0001.

ABSTRACT

We introduce and study the problem of planning a trajectory for an agent to carry out a scouting mission while avoiding being detected by an adversarial guard. This introduces a multi-objective version of classical visibility-based target search and pursuit-evasion problem. In our formulation, the agent receives a positive reward for increasing its visibility(by exploring new regions) and a negative penalty every time it is detected by the guard. The objective is to find a finite-horizon path for the agent that balances the trade off between maximizing visibility and minimizing detectability



PROBLEM FORMULATION

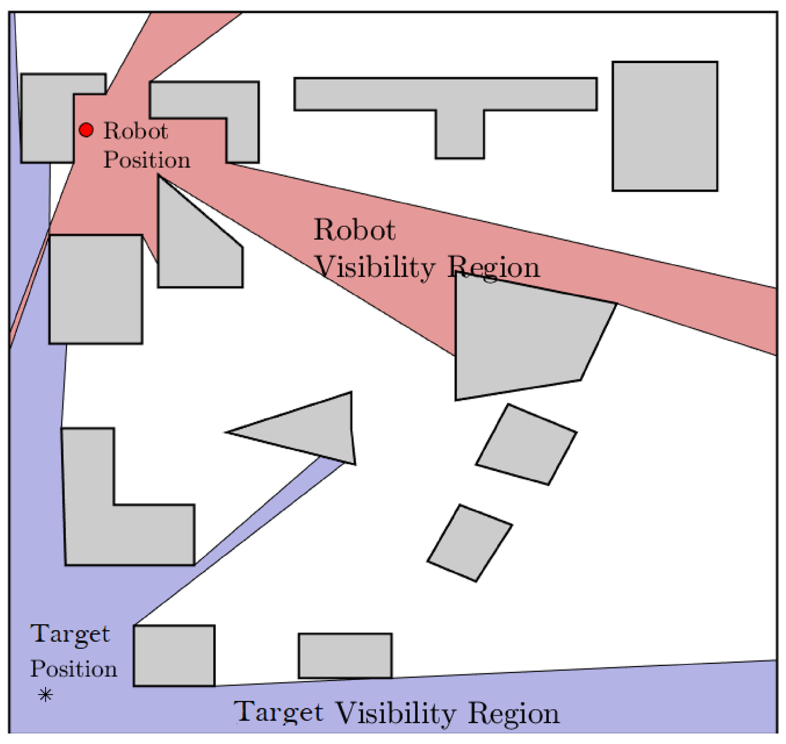
- Given an environment, design a path for an agent that collects as much information as possible (**maximize total visibility**) and avoids being detected by the target (**minimize detectability**).

$$\max_{\pi_a(t)} \min_{\pi_g(t)} \{R(\pi_a(t)) - \eta(\pi_a(t), \pi_g(t))P\}$$

$R(\pi_a(t))$: Positive reward equal to the total area seen by the robot from time 0 to t;

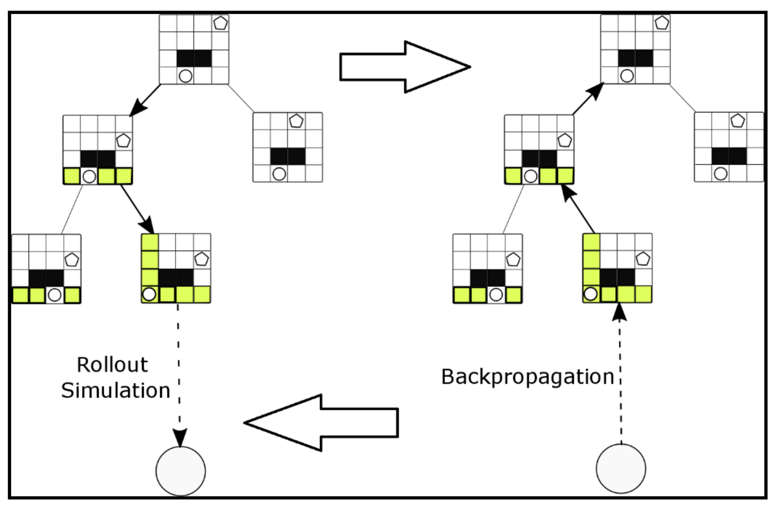
$\eta(\pi_a(t), \pi_g(t))$: The number of times that the agent is detected from time 0 to t;

P : Negative penalty value.

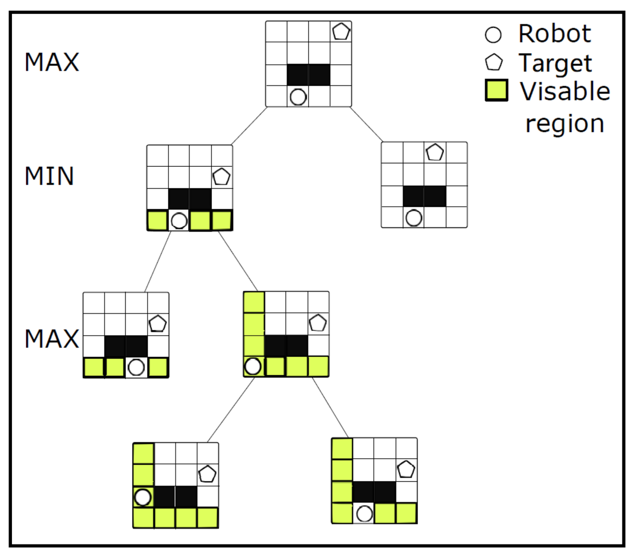


APPROACHES

- We solve this problem by modeling it as a turn-based game between an agent and a target.
- We apply **Minimax search tree** and **Monte Carlo search tree**



MCTS: Run random simulations and build a search tree from the results.



Minimax tree with Pruning: The full minimax search explores redundant parts of the tree.

PROPOSED PRUNING STRATEGIES

Contributions:

- Faster computation Pruning strategy that reduces the search space by ignoring (provably) suboptimal strategies

Proposed Pruning Strategies:

- A node can be pruned if even in the most optimistic future scenario it is dominated by some other node at the same level.
- If two nodes have the same agent and target position, and one node has a better history, then the other can be pruned away.

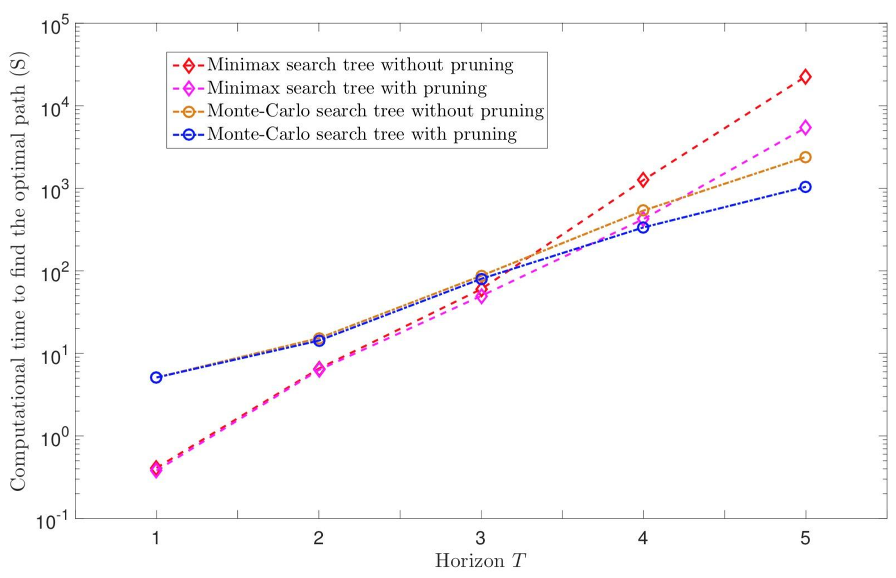


TABLE I

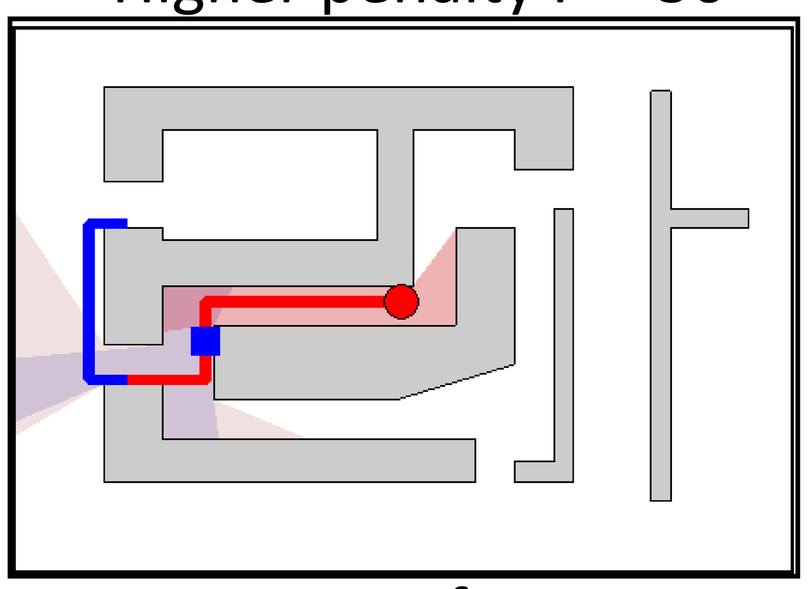
COMPARISION OF THE NUMBER OF NODES GENERATED BY DIFFERENT PRUNING TECHNIQUES, FROM $T = 3$ TO $T = 6$.

		Number of nodes generated			
Planning horizon		$T = 3$	$T = 4$	$T = 5$	$T = 6$
Brute force		625	1.56E4	3.90E5	9.76E6
With only alpha-beta	Maximum	403	3844	7.08E4	1.70E6
	Median	206	2822	1.80E4	2.46E5
	Minimum	104	1444	7860	1.86E5
With all pruning techniques	Maximum	388	1389	3.3E4	4.81E5
	Median	105	639	4064	3.74E4
	Minimum	78	563	3016	2.94E4

RESULTS

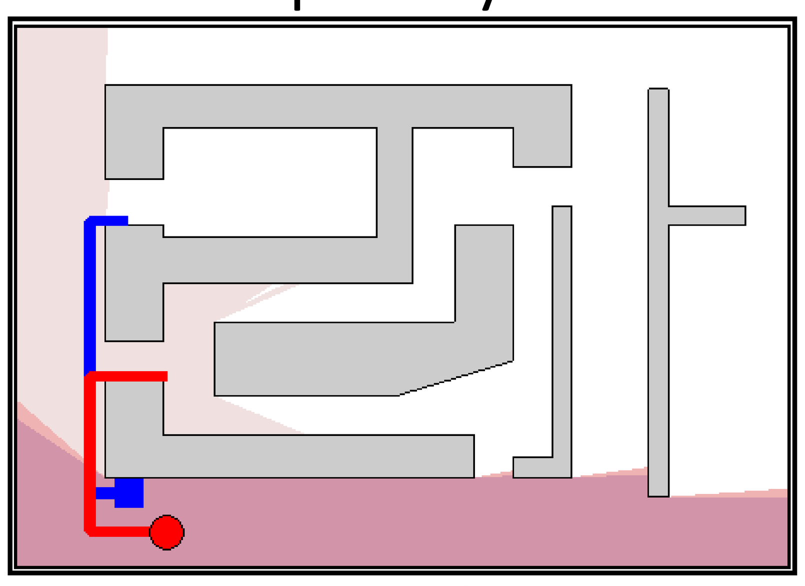
Qualitative examples:

Higher penalty P = 30



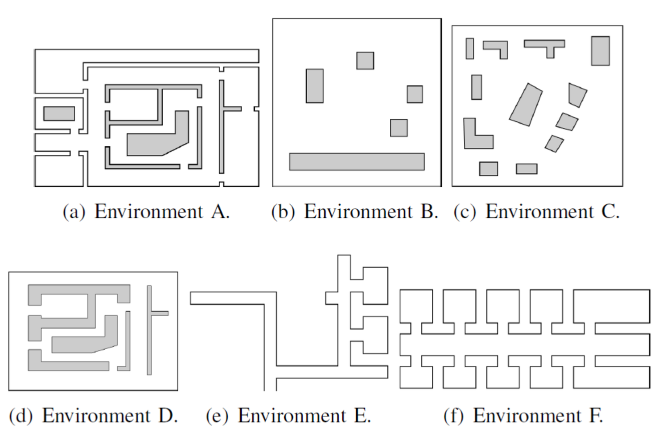
agent prefers to avoid detection by **adversary**

Lower penalty P = 3



agent prefers to explore new areas

Online Path Planning examples:



Online path planning environments

Envir-onment	Number of times, greedy preforms better	Number of times, minimax preforms as well or better
A	74 (4.45%)	1590 (95.55%)
B	41 (41.0%)	59 (59.0%)
C	35 (9.7%)	325 (90.3%)
D	43 (27.0%)	116 (73.0%)
E	17 (36.2%)	30 (63.8%)
F	37 (11.6%)	283 (88.4%)

Online minimax search tree VS. Greedy