Planning in probabilistic environment

15. dubna 2019

B4M36PUI/BE4M36PUI — Planning for Artificial Intelligence

Outline

- Organization of second part of tuorials in B4M36PUI/BE4M36PUI
- Probabilistic planning tutorials overview

- Dijkstra/A* replan
- (Builiding a partial policy - Robust replan)

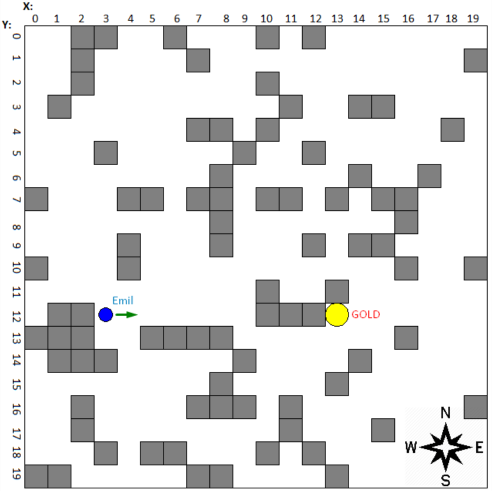
Organization

Organization

- tutorials in EN
- 1 assignment in three parts - 20 points
- In-class bonus points per assignments (code, proofs, ...)
- deadlines are soft (- points per day)
- Language: Java
- Recommended setup for tutorials: download JDK8 and Intellij idea

From classical planning to probabilistic planning

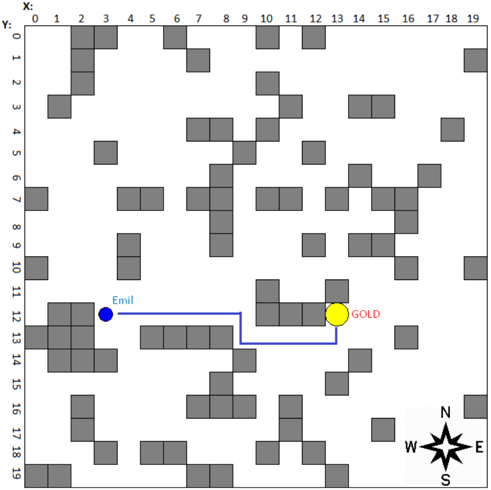
Emil in the gridworld



- Emil likes the siny gold.

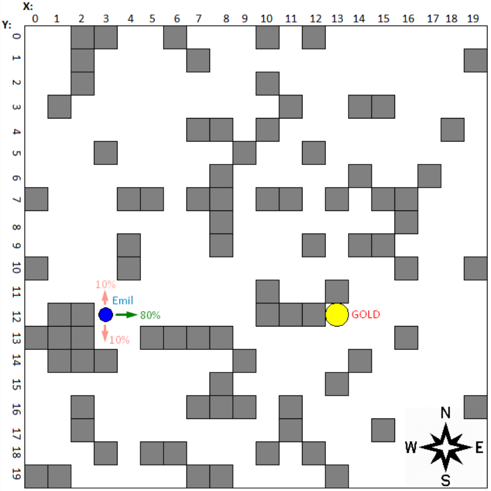
Question: How do you find a (shortest) plan from current position to the gold?

Emil in the gridworld



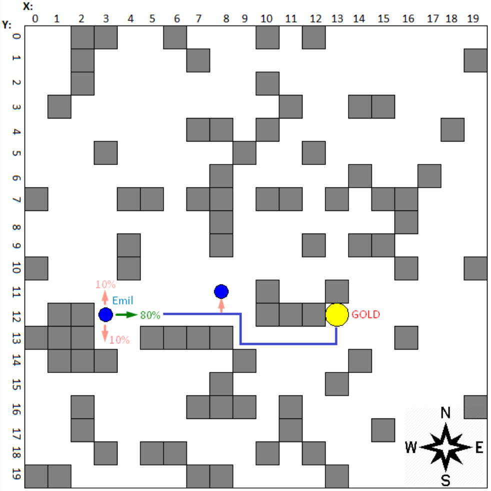
- Dijkstra/A*, or FF in more general domain.

Emil in the stochastic gridworld



- Specified actions are not guaranteed to execute.

Emil in the stochastic gridworld



- Specified actions are not guaranteed to execute.

Question: What to you do when Emil deviates from the plan?

Replanning

1. Find a plan in determinized environment
2. If you detect a deviation, replan from new state

Finish implementation of `RobotEmilAgent`

Implement the deviation detection and replanning in method `nextStep` of `RobotEmilAgent` in file `RobotEmilAgent.java`. To get the plan, you can use any planning algorithm, such as Dijkstra or A*.

Question: What are the drawbacks of replanning? **Question 2:** How can we fix them?

Robust replanning

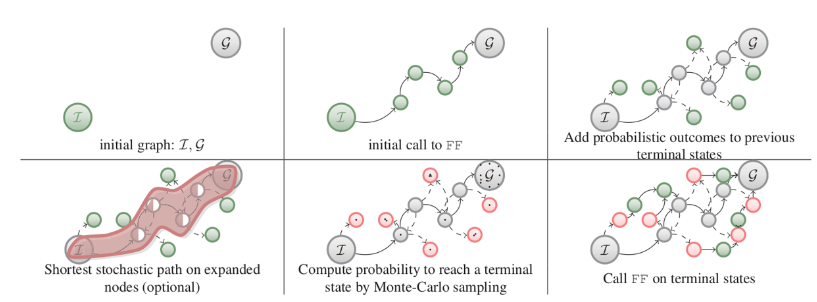
Robust Replanning

Plan with contingencies is called **policy**.

Robust replanning for partial policies:

1. Find a plan in determinized environment. Save this plan as partial policy.
2. Find failure nodes (locations) agent can reach to by deviating from the partial policy.
3. Find plans from failure nodes and add them to the partial policy.
4. Iterate over steps 2. and 3.

Robust Replanning



initial graph: $\mathcal{I}, \mathcal{G}$

initial call to FF

Add probabilistic outcomes to previous terminal states

Shortest stochastic path on expanded nodes (optional)

Compute probability to reach a terminal state by Monte-Carlo sampling

Call FF on terminal states

Robust Replanning

Download modified environmnet from courseware and:

- Implement robust replanning algorithm.
- Report number of plan failures as dependent on number of policy update steps.

Implement robust replanning for `RobotEmilAgent`

Implement robust replanning in `nextStep` of `RobotEmilAgent` in file `RobotEmilAgent.java`. Feel free to reuse parts of the reference solutions.

Question: Is the shortest path different in probabilistic environment?