The Wumpus World:
A Demonstration of Probabilistic Decision Making

Rowan College at Burlington County

Bryan Bugyi

Advised by Jonathan Weisbrod

Rules of Wumpus World

Description

- A Wumpus World is a cave consisting of rooms connected by passageways.
- Lurking somewhere in the cave is the terrible wumpus, a beast that eats anyone who enters the room.
- Some rooms contain bottomless pits that will trap anyone who wanders into these rooms.
- The only mitigating feature of this bleak environment is the possibility of finding a heap of gold. The agent's goal is to find the gold without falling into a pit or being eaten by the wumpus.

Environment

- The environment is a 4×4 grid of rooms.
- The agent always starts in a corner square.
- The locations of the gold and the wumpus are chosen randomly, with uniform distribution, from the remaining squares.
- In addition, each remaining square has a 0.2 probability of being a pit.

Sensors

- If a square is adjacent (not diagonal) to a pit, the agent will sense a "Breeze" when at this square; if the square is adjacent to the wumpus, the agent will sense a "Stench".
- ***The agent is only aware of the contents of the square that it currently resides in.***

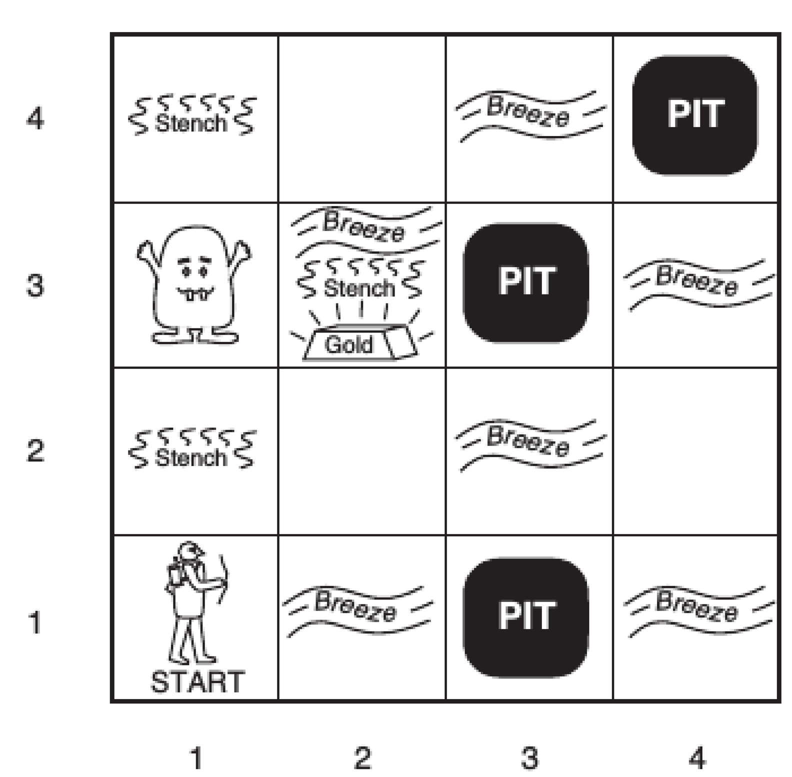


Figure 1 : An example wumpus world environment

Important Definitions

$$E = \{\boldsymbol{x} \mid \boldsymbol{x} \text{ is a square in the environment}\}$$

$$V = \{\boldsymbol{x} \mid \text{The agent has visited } \boldsymbol{x},\ \boldsymbol{x} \in E\}$$

$$N(A) = \{\boldsymbol{x} \mid \exists \boldsymbol{a} \in A \text{ where } \boldsymbol{x} \text{ is adjacent to } \boldsymbol{a} \text{ or } \boldsymbol{x} = \boldsymbol{a}\}$$

Wumpus Probability

Let $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m\}$ represent the set of all squares that the agent has visited where a "Stench" sense was *not* detected.

Let $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ represent the set of all squares that the agent has visited where a "Stench" sense *was* detected.

We can now use S and T to derive the set of all squares where the wumpus could possibly be found; we shall call this set W.

$$W = \begin{cases} N(\boldsymbol{s}_1) \cap \cdots \cap N(\boldsymbol{s}_n) - N(T) & \text{if } S \text{ is non-empty} \\ E - N(T) & \text{otherwise} \end{cases}$$

Final Result

Therefore, by the axioms of probability, we can deduce that

$$P(w_{\boldsymbol{x}}) = \begin{cases} \frac{1}{|W|} & \text{if } \boldsymbol{x} \in W \\ 0 & \text{if } \boldsymbol{x} \notin W \end{cases}$$

where $P(w_{\boldsymbol{x}})$ represents the probability of a wumpus being located at square $\boldsymbol{x}$.

Utility Function

When deciding on which square to move to next, the agent will consider the square $\boldsymbol{x}$ only if $\boldsymbol{x} \in V'$ where $V' = N(V) - V$.

The agent will consider two factors when deciding on the best next square: *death probability* and *distance*, where the former takes precedence over the latter. If all squares considered have a death probability of 0.5 or higher, the agent will forfeit.

Once decided on the next square to travel to, the agent must find the shortest path to said square. . .

Graph Structure / Shortest Path

After the next square is chosen, the agent models a portion of the environment using a graph-like structure where the nodes are $\{\boldsymbol{x} \mid \boldsymbol{x} \text{ is the target square or } \boldsymbol{x} \in V\}$.

Figure 2 shows an example of how this structure can be visualized using an undirected graph.

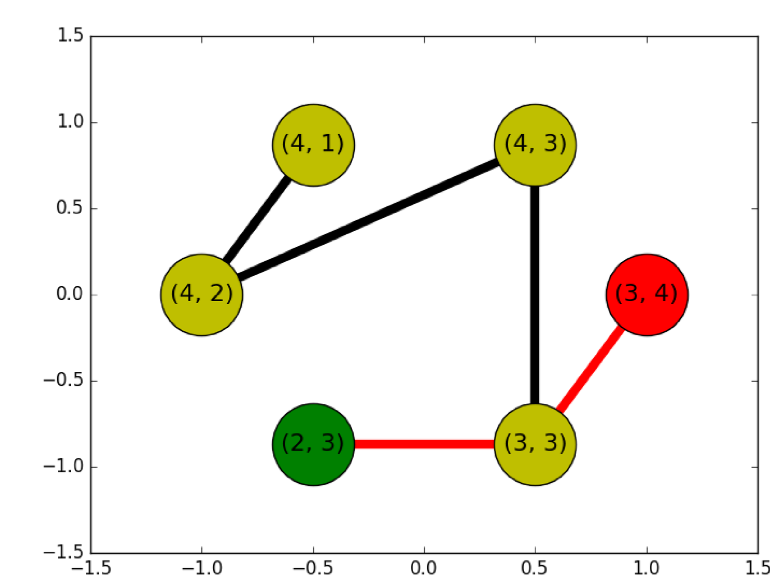


Figure 2 : Visualization of the graph structure used to determine the shortest path from $(2, 3)$—the agent's location in figure 3a—to $(3, 4)$—the agent's location in figure 3b.

The agent then calculates the shortest path (shown in red in figure 2) with the help of Dijkstra's algorithm.

Pit Probability

Calculating $P(p_{\boldsymbol{x}})$, the probability that a pit is located at square $\boldsymbol{x}$, is a much more difficult task.

During gameplay, the agent builds a set of models (one for each possible world), and then selectively filters this set as the game goes on. Essentially, it can be shown that $P(p_{\boldsymbol{x}}) = \sum_i^n P(M_{i\boldsymbol{x}})$ for all n models where $\boldsymbol{x}$ contains a pit.

Important Note

To optimize this process, the agent only models worlds for the squares contained in V'. This set includes all possible "next move" squares. Otherwise, a 10×10 environment like that shown in Figure 4 would require 2^{100} models be built!

Conclusion

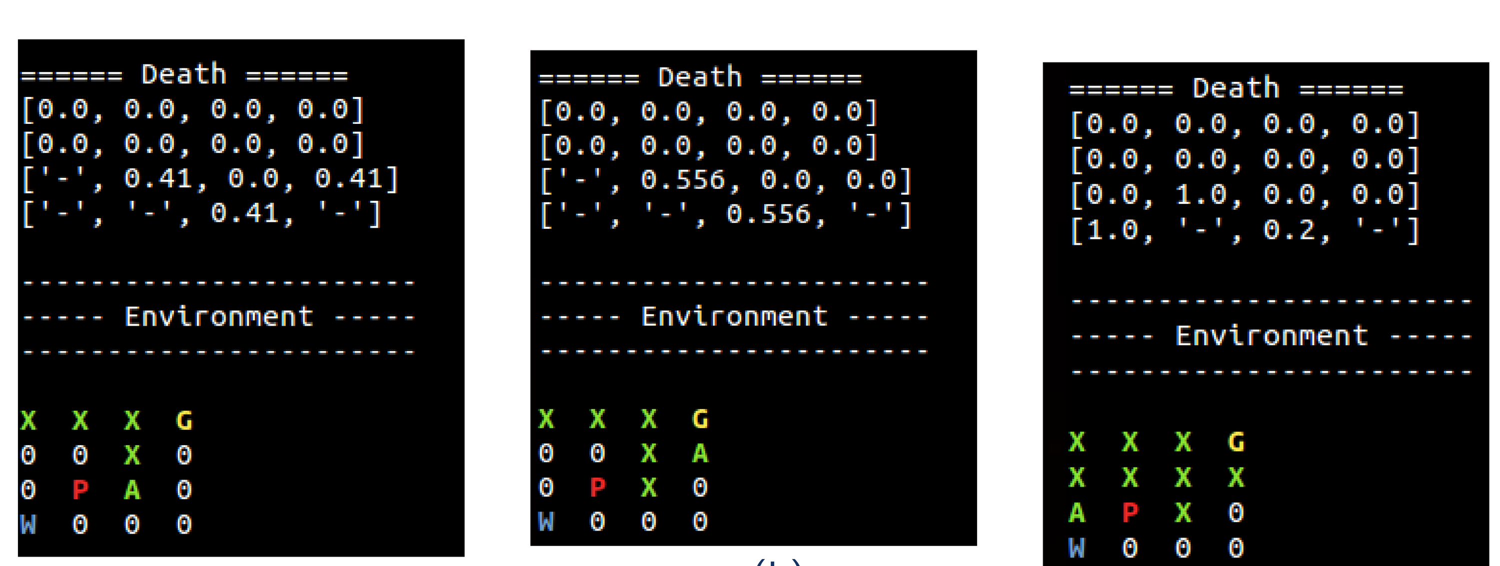
A 5000 trial test was run on this implementation of the wumpus world agent (using a 4×4 environment). The results were as follows:

- Found Gold: 2568 (51%)
- Flopped: 2013 (40%)
- Death: 237 (4%)
- Forfeited: 182 (3%)

I have defined "flop" in this context to indicate that the agent forfeited on the first move (as it will if it spawns directly adjacent to a pit or wumpus).

Therefore, of the environments that did not result in a flop, the agent found the gold 86% of the time, died 8% of the time, and forfeited 6% of the time.

Running Examples



(a) (b) (c)

Figure 3 : A test run showing WW environment along with the agent's perceived death probabilities $[P(w_{\boldsymbol{x}}) + P(p_{\boldsymbol{x}})]$

Figure 4 : A 10×10 WW environment

References

[1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2009

[2] Sheldon Ross. *A First Course in Probability.* Prentice Hall, 2009

[3] Kenneth H. Rosen. *Discrete Math and Its Applications.* McGraw-Hill, 2012

Contact Information

- Web: bryanbugyi.com
- Email: bryan_bugyi@mymail.rcbc.edu

The source code for this project can be found on my GitHub page at github.com/bbugyi200/WumpusWorld.