Project 2: Aliens - Rematch

16:198:520

It is another day on the deep space vessel *Archaeopteryx*, and you are a tiny bot. You're responsible for the safety and security of the ship and crew. The aliens are back.

1 The Ship

The ship layout is as in Project 1. For the purpose of this project, set the dimension to 50×50.

There are aliens loose on the ship. There are crew members hidden in the ship. You want to avoid the former and rescue the latter.

2 The Bot, the Sensors, the Aliens, the Crew

The bot occupies an open cell somewhere in the ship (to be determined). The bot can move as in Project 1 (up/down/left/right/stay in place).

The bot has access to two kinds of sensor data at every timestep:

- For a given k, the bot can sense whether there is an alien in the $(2k+1) \times (2k+1)$ square centered at its location.
- For a given $\alpha > 0$, if there is a crewmember hidden d-steps away from the bot, the bot receives a beep with probability $e^{-\alpha(d-1)}$.

As before, aliens move randomly to an adjacent empty square every timestep. If an alien enters the bot cell, or the bot enters an alien cell, the bot is destroyed and the game is over.

At the start of the game:

- The bot is placed on a random open cell.
- Crew members are placed at random cells distinct from the bot's cell.
- Aliens are placed at random cells, chosen outside the $(2k+1) \times (2k+1)$ of the initial bot position.

3 One Alien, One Crew Member

In this case, we have a single alien in play, and a single crew member that needs rescuing.

- **Bot 1:** At the start, the bot knows that the alien is equally likely to be in any open cell outside the bot's detection square, and the crew member is equally likely to be in any cell other than the bot's initial cell. At every point in time, update what is known about the crew member and the alien based on the data received (*How?*). (Note: the bot necessarily has perfect knowledge of the cell that it is currently in.) Note, when the alien has the opportunity to move, the bot's knowledge of the alien should be updated accordingly (*How?*). The bot should proceed by moving toward the cell most likely to contain the crew member (breaking ties at random), sticking to cells that definitely do not contain the alien.

- **Bot 2:** A bot of your own design.

NOTE: There is a correct way to update the bot's knowledge of the alien and crew member. You need to work out what the correct updates are.

4 One Alien, Two Crew Members

In this case, we have a single alien in play, and two crew members that needs rescuing. Note that a crew-detection-beep is received if the bot receives a beep from either crew member.

- **Bot 3:** Bot 3 is just Bot 1 applied in this new setting, but when the first crew member is found, they are teleported away, and the updates continue until the second crewmember is found.
- **Bot 4:** Bot 4 is Bot 1, *except that the probabilities of where the crew members are account for the fact that there are two of them (**How?**), and are updated accordingly.*
- **Bot 5:** A bot of your own design.

NOTE: There is a correct way to update the bot's knowledge of the alien and crew member. You need to work out what the correct updates are.

5 Two Aliens, Two Crew Members

In this case, we have two aliens in play, and two crew members that needs rescuing. Note that a crew-detection-beep is received if the bot receives a beep from either crew member. Note that the alien-detection square does not distinguish between one or two aliens detected.

- **Bot 6:** Bot 6 is just Bot 1, applied in this new setting, but when the first crew member is found, they are teleported away, and the updates continue until the second crewmember is found.
- **Bot 7:** Bot 7 is Bot 1, *with correct modeling and updates for the fact that there are now two aliens and two crew members.*
- **Bot 8:** A bot of your own design.

6 Data and Analysis

In your writeup, consider and address the following:

1) Explain the design and algorithm for the bots you designed, being as specific as possible as to what your bots are actually doing. How do your bots factor in the available information (deterministic or probabilistic) to make more informed decisions about what to do next?
2) Be clear about how you are modeling and updating your probabilities in each bot, being explicit how the data collected (including beeps or lack of beeps) is taken into account.

3) Generate test environments to evaluate and compare the performances of your bots. Your measures of performance here are *average number of moves needed to rescue all crew members*, *probability of successfully avoiding the alien and rescuing the crew*, and *average number of crew members saved.* You should experiment with different k values (if k is too small, you get no information, but if k is too large, you don't get useful information), and plot your results as a function of α, which can be thought of as crew detection sensitivity. You should compare the following:

   - Bot 1 vs Bot 2.
   - Bot 3 vs Bot 4 vs Bot 5.
   - Bot 6 vs Bot 7 vs Bot 8.

4) Speculate on how you might construct the ideal bot. What information would it use, what information would it compute, and how?

Note that over all, your goal should be for the bots of your design to beat the other bots as much as is possible - but you should be clear about what metrics you are trying to improve on, how, and what tradeoffs you are considering.

Bonus: *Suppose the crew members can also move one cell per timestep, at random. What changes, and how should your bots change to account for it?*