Artificial Intelligence (16:198:520:01) Cynthia Zhu
Professor Cowan November 12, 2023

Project 2

Question 1

Bots

All the bots follow the same general algorithm. The general algorithm that all the bots follow is

```
WHILE there are crew members that need to be saved:

    num_time_steps += 1

    The bot decides what to do next.
    Based on the decision, the bot moves up, down, left, or right or stays in place.

    IF the bot is in the same cell as an alien:
        BREAK
    ELSE:
        Update the alien probabilities, given that the bot didn't find an alien.

    IF the bot is in the same cell as a crew member:
        num_saved += 1
        Remove the crew member.
        IF there are no more crew members that need to be saved:
            is_success = 1
            BREAK
    Update the crew probabilities, given that the bot didn't find a crew member.

    IF the bot detects an alien:
        Update the alien probabilities, given that the bot detected an alien.
    ELSE:
        Update the alien probabilities, given that the bot didn't detect an alien.

    IF the bot detects a crew member:
        Update the crew probabilities, given that the bot detected a crew member.
    ELSE:
        Update the crew probabilities, given that the bot didn't detect a crew member.

    The aliens move randomly up, down, left, or right.

    IF the bot is in the same cell as an alien:
        BREAK
    ELSE:
        Update the alien probabilities, given that the aliens moved.

RETURN num_saved, num_time_steps, is_success
```

The differences between the bots are

1. How the bots decide what to do next, which depends on the bot number
2. How the alien probabilities are updated, which depends on the number of aliens and the bot number
3. How the crew probabilities are updated, which depends on the number of crew members and the bot number

Bot 1

Bot 1 decides what to do next based on the following algorithm:

1. Get open neighbors.
2. Of these neighbors, get the neighbors with p_alien = 0.
   A. If there are no such neighbors, stay in place.
3. Of these neighbors, move to the neighbor with the maximum p_crew, breaking ties randomly.

Bot 1 updates probabilities based on “One crew member, one alien” (5–8).

Bot 2

Bot 2 decides what to do next based on the following idea:

- Intuitively, the bot should move to a neighbor with a high p_crew and a low p_alien.
- Computationally, the bot should move to the neighbor that maximizes the difference between p_crew and p_alien.

Bot 2 decides what to do next based on the following algorithm:

1. Get open neighbors.
2. For each open neighbor, calculate the difference between p_crew and p_alien.
3. Move to the neighbor with the maximum difference, breaking ties randomly.

Bot 2 updates probabilities based on “One crew member, one alien” (5–8).

Bot 3

Bot 3 decides what to do next based on the same algorithm as Bot 1.

Bot 3 updates probabilities based on “One crew member, one alien” (5–8).

Bot 4

Bot 4 decides what to do next based on the same algorithm as Bot 1.

When there are two crew members that need to be saved, Bot 4 updates probabilities based on "Two crew members, one alien" (9–12). When there is one crew member that needs to be saved, Bot 4 updates probabilities based on "One crew member, one alien" (5–8).

Bot 5

Bot 5 decides what to do next based on the following idea:

- Intuitively, if the bot detects an alien, the bot should move away from the alien. If the bot doesn't detect an alien, the bot should move toward the crew member.
- Computationally, if the bot detects an alien, the bot should move to the neighbor with the minimum p_alien. If the bot doesn't detect an alien, the bot should move to the neighbor with the maximum p_crew.

Bot 5 decides what to do next based on the following algorithm:

1. Get open neighbors.
2. If the bot detects an alien, move to the neighbor with the minimum p_alien, breaking ties randomly.
3. If the bot doesn't detect an alien, move to the neighbor with the maximum p_crew, breaking ties randomly.

When there are two crew members that need to be saved, Bot 5 updates probabilities based on "Two crew members, one alien" (9–12). When there is one crew member that needs to be saved, Bot 5 updates probabilities based on "One crew member, one alien" (5–8).

Bot 6

Bot 6 decides what to do next based on the same algorithm as Bot 1.

Bot 6 updates probabilities based on "One crew member, one alien" (5–8).

Bot 7

Bot 7 decides what to do next based on the same algorithm as Bot 1.

When there are two crew members that need to be saved, Bot 7 updates probabilities based on "Two crew members, two aliens" (12–15). When there is one crew member that needs to be saved, Bot 7 updates crew probabilities based on "One crew member, one alien" (5–8) and updates alien probabilities based on "Two crew members, two aliens" (12–15).

Bot 8

Bot 8 is a combination of Bot 2 and Bot 5.

Bot 8 decides what to do next based on the following idea:

- Intuitively, if the bot detects an alien, the bot should move to a neighbor with a high p_crew and a low p_alien. (This ensures that the bot is still making progress toward finding the crew member while avoiding the alien.) If the bot doesn't detect an alien, the bot should move toward the crew member.
- Computationally, if the bot detects an alien, the bot should move to the neighbor that maximizes the difference between p_crew and p_alien. If the bot doesn't detect an alien, the bot should move to the neighbor with the maximum p_crew.

Bot 8 decides what to do next based on the following algorithm:

1. Get open neighbors.
2. If the bot detects an alien,
   A. For each open neighbor, calculate the difference between p_crew and p_alien.
   B. Move to the neighbor with the maximum difference, breaking ties randomly.
3. If the bot doesn't detect an alien,
   A. Move to the neighbor with the maximum p_crew, breaking ties randomly.

When there are two crew members that need to be saved, Bot 8 updates probabilities based on "Two crew members, two aliens" (12–15). When there is one crew member that needs to be saved, Bot 8 updates crew probabilities based on "One crew member, one alien" (5–8) and updates alien probabilities based on "Two crew members, two aliens" (12–15).

Updating probabilities

Notation

i	the position of the bot
j	any open cell
$adj(c)$	the set of open cells adjacent to c
$adj\big((c,d)\big)$	$adj(c) \times adj(d)$ (Cartesian product)
$dist(c,d)$	the length of the shortest path from c to d
$DS(i)$	the set of cells in the alien detection square centered at i
$DS_{in}(i)$	$\{(j_1, j_2) \mid j_1 \in DS(i)$ or $j_2 \in DS(i)\}$
$DS_{out}(i)$	$\{(j_1, j_2) \mid j_1 \notin DS(i)$ and $j_2 \notin DS(i)\}$

One crew member, one alien

Initial conditions

$$\mathrm{P}(\text{crew in } j) = \begin{cases} 0 & \text{if } j = i \\ \dfrac{1}{\text{number of open cells} - 1} & \text{otherwise} \end{cases}$$

$$\mathrm{P}(\text{alien in } j) = \begin{cases} 0 & \text{if } j \in DS(i) \\ \dfrac{1}{\text{number of open cells outside } DS(i)} & \text{otherwise} \end{cases}$$

If the bot doesn't find the alien

$$\mathrm{P}(\text{alien in } j \mid \text{alien not in } i) = \begin{cases} 0 & \text{if } j = i \\ \dfrac{\mathrm{P}(\text{alien in } j)}{1 - \mathrm{P}(\text{alien in } i)} & \text{otherwise} \end{cases}$$

<u>Proof</u>

If $j \neq i$,

$$\begin{aligned} \mathrm{P}(\text{alien in } j \mid \text{alien not in } i) &= \frac{\mathrm{P}(\text{alien in } j \wedge \text{alien not in } i)}{\mathrm{P}(\text{alien not in } i)} \\ &= \frac{\mathrm{P}(\text{alien in } j)\mathrm{P}(\text{alien not in } i \mid \text{alien in } j)}{1 - \mathrm{P}(\text{alien in } i)} \\ &= \frac{\mathrm{P}(\text{alien in } j)}{1 - \mathrm{P}(\text{alien in } i)} \end{aligned}$$

If the bot doesn't find the crew member

$$\mathrm{P}(\text{crew in } j | \text{crew not in } i) = \begin{cases} 0 & \text{if } j = i \\ \dfrac{\mathrm{P}(\text{crew in } j)}{1 - \mathrm{P}(\text{crew in } i)} & \text{otherwise} \end{cases}$$

<u>Proof</u>

If $j \neq i$,

$$\begin{aligned} \mathrm{P}(\text{crew in } j | \text{crew not in } i) &= \frac{\mathrm{P}(\text{crew in } j \wedge \text{crew not in } i)}{\mathrm{P}(\text{crew not in } i)} \\ &= \frac{\mathrm{P}(\text{crew in } j)\mathrm{P}(\text{crew not in } i | \text{crew in } j)}{1 - \mathrm{P}(\text{crew in } i)} \\ &= \frac{\mathrm{P}(\text{crew in } j)}{1 - \mathrm{P}(\text{crew in } i)} \end{aligned}$$

If the bot detects the alien

$$\mathrm{P}(\text{alien in } j | \text{alien detected from } i) = \begin{cases} \dfrac{\mathrm{P}(\text{alien in } j)}{\sum_{k \in DS(i)} \mathrm{P}(\text{alien in } k)} & \text{if } j \in DS(i) \\ 0 & \text{otherwise} \end{cases}$$

<u>Proof</u>

$$\begin{aligned} \mathrm{P}(\text{alien in } j | \text{alien detected from } i) &= \frac{\mathrm{P}(\text{alien in } j \wedge \text{alien detected from } i)}{\mathrm{P}(\text{alien detected from } i)} \\ &= \frac{\mathrm{P}(\text{alien in } j \wedge \text{alien detected from } i)}{\sum_k \mathrm{P}(\text{alien in } k \wedge \text{alien detected from } i)} \\ &= \frac{\mathrm{P}(\text{alien in } j)\mathrm{P}(\text{alien detected from } i | \text{alien in } j)}{\sum_k \mathrm{P}(\text{alien in } k)\mathrm{P}(\text{alien detected from } i | \text{alien in } k)} \\ &= \frac{\mathrm{P}(\text{alien in } j)\mathrm{P}(\text{alien detected from } i | \text{alien in } j)}{\sum_{k \in DS(i)} \mathrm{P}(\text{alien in } k)} \end{aligned}$$

$$\mathrm{P}(\text{alien detected from } i | \text{alien in } j) = \begin{cases} 1 & \text{if } j \in DS(i) \\ 0 & \text{otherwise} \end{cases}$$

If the bot doesn't detect the alien

$$\mathrm{P}(\text{alien in } j|\text{alien not detected from } i) = \begin{cases} \dfrac{\mathrm{P}(\text{alien in } j)}{\sum_{k \notin DS(i)} \mathrm{P}(\text{alien in } k)} & \text{if } j \notin DS(i) \\ 0 & \text{otherwise} \end{cases}$$

<u>Proof</u>

$$\mathrm{P}(\text{alien in } j|\text{alien not detected from } i) = \frac{\mathrm{P}(\text{alien in } j \wedge \text{alien not detected from } i)}{\mathrm{P}(\text{alien not detected from } i)}$$

$$= \frac{\mathrm{P}(\text{alien in } j \wedge \text{alien not detected from } i)}{\sum_k \mathrm{P}(\text{alien in } k \wedge \text{alien not detected from } i)}$$

$$= \frac{\mathrm{P}(\text{alien in } j)\mathrm{P}(\text{alien not detected from } i|\text{alien in } j)}{\sum_k \mathrm{P}(\text{alien in } k)\mathrm{P}(\text{alien not detected from } i|\text{alien in } k)}$$

$$= \frac{\mathrm{P}(\text{alien in } j)\mathrm{P}(\text{alien not detected from } i|\text{alien in } j)}{\sum_{k \notin DS(i)} \mathrm{P}(\text{alien in } k)}$$

$$\mathrm{P}(\text{alien not detected from } i|\text{alien in } j) = \begin{cases} 1 & \text{if } j \notin DS(i) \\ 0 & \text{otherwise} \end{cases}$$

If the bot detects the crew member

$$\mathrm{P}(\text{crew in } j|\text{crew detected from } i) = \frac{\mathrm{P}(\text{crew in } j) \times e^{-\alpha(dist(i,j)-1)}}{\sum_k \mathrm{P}(\text{crew in } k) \times e^{-\alpha(dist(i,k)-1)}}$$

<u>Proof</u>

$$\mathrm{P}(\text{crew in } j|\text{crew detected from } i) = \frac{\mathrm{P}(\text{crew in } j \wedge \text{crew detected from } i)}{\mathrm{P}(\text{crew detected from } i)}$$

$$= \frac{\mathrm{P}(\text{crew in } j \wedge \text{crew detected from } i)}{\sum_k \mathrm{P}(\text{crew in } k \wedge \text{crew detected from } i)}$$

$$= \frac{\mathrm{P}(\text{crew in } j)\mathrm{P}(\text{crew detected from } i|\text{crew in } j)}{\sum_k \mathrm{P}(\text{crew in } k)\mathrm{P}(\text{crew detected from } i|\text{crew in } k)}$$

$$= \frac{\mathrm{P}(\text{crew in } j) \times e^{-\alpha(dist(i,j)-1)}}{\sum_k \mathrm{P}(\text{crew in } k) \times e^{-\alpha(dist(i,k)-1)}}$$

If the bot doesn't detect the crew member

$$\text{P(crew in } j \text{|crew not detected from } i) = \frac{\text{P(crew in } j) \times \left(1 - e^{-\alpha(dist(i,j)-1)}\right)}{\sum_k \text{P(crew in } k) \times (1 - e^{-\alpha(dist(i,k)-1)})}$$

Proof

$$\text{P(crew in } j \text{|crew not detected from } i) = \frac{\text{P(crew in } j \wedge \text{crew not detected from } i)}{\text{P(crew not detected from } i)}$$

$$= \frac{\text{P(crew in } j \wedge \text{crew not detected from } i)}{\sum_k \text{P(crew in } k \wedge \text{crew not detected from } i)}$$

$$= \frac{\text{P(crew in } j)\text{P(crew not detected from } i \text{|crew in } j)}{\sum_k \text{P(crew in } k)\text{P(crew not detected from } i \text{|crew in } k)}$$

$$= \frac{\text{P(crew in } j) \times \left(1 - e^{-\alpha(dist(i,j)-1)}\right)}{\sum_k \text{P(crew in } k) \times (1 - e^{-\alpha(dist(i,k)-1)})}$$

The alien moves

Assume: The alien must move up, down, left, or right and cannot stay in place.

$$\text{P(alien in } j) = \sum_{k \in adj(j)} \text{P(alien in } k) \cdot \frac{1}{\text{number of open cells adjacent to } k}$$

Proof

$$\text{P(alien now in } j) = \sum_k \text{P(alien was in } k \wedge \text{alien now in } j)$$

$$= \sum_k \text{P(alien was in } k)\text{P(alien now in } j \text{|alien was in } k)$$

$$= \sum_{k \in adj(j)} \text{P(alien was in } k) \cdot \frac{1}{\text{number of open cells adjacent to } k}$$

$$\text{P(alien now in } j \text{|alien was in } k) = \begin{cases} \dfrac{1}{\text{number of open cells adjacent to } k} & \text{if } k \in adj(j) \\ 0 & \text{otherwise} \end{cases}$$

Two crew members, one alien

We now store a crew probability for each pair of open cells.

Assume: Crew members cannot be in the same cell.

The number of pairs is $\binom{\text{number of open cells}}{2}$.

Initial conditions

$$\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) = \begin{cases} 0 & \text{if } (j_1, j_2) \ni i \\ \dfrac{1}{\binom{\text{number of open cells}}{2} - (\text{number of open cells} - 1)} & \text{otherwise} \end{cases}$$

$$\mathrm{P}(\text{alien in } j) = \begin{cases} 0 & \text{if } j \in DS(i) \\ \dfrac{1}{\text{number of open cells outside } DS(i)} & \text{otherwise} \end{cases}$$

<u>Proof</u>

If $(j_1, j_2) \not\ni i$,

$$\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) = \frac{1}{\text{number of pairs of open cells} - \text{number of pairs of open cells containing bot}}$$

$$= \frac{1}{\binom{\text{number of open cells}}{2} - (\text{number of open cells} - 1)}$$

If the bot doesn’t find the alien

Same as One crew member, one alien.

If the bot doesn't find the crew member

$$\mathrm{P}(\text{crew in } (j_1, j_2) | \text{crew not in } i) = \begin{cases} 0 & \text{if } (j_1, j_2) \ni i \\ \dfrac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)}{1 - \sum_{(k_1, k_2) \ni i} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big)} & \text{otherwise} \end{cases}$$

<u>Proof</u>

If $(j_1, j_2) \not\ni i$,

$$\mathrm{P}(\text{crew in } (j_1, j_2) | \text{crew not in } i) = \frac{\mathrm{P}(\text{crew in } (j_1, j_2) \wedge \text{crew not in } i)}{\mathrm{P}(\text{crew not in } i)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)\mathrm{P}\big(\text{crew not in } i \big| \text{crew in } (j_1, j_2)\big)}{1 - \mathrm{P}(\text{crew in } i)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)}{1 - \sum_{(k_1, k_2) \ni i} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big)}$$

If the bot detects the alien

Same as One crew member, one alien.

If the bot doesn't detect the alien

Same as One crew member, one alien.

If the bot detects the crew member

P(crew in (j_1, j_2)|crew detected from i)

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) \times \left[e^{-\alpha(dist(i,j_1)-1)} + e^{-\alpha(dist(i,j_2)-1)} - \left(e^{-\alpha(dist(i,j_1)-1)} \times e^{-\alpha(dist(i,j_2)-1)}\right)\right]}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big) \times [e^{-\alpha(dist(i,k_1)-1)} + e^{-\alpha(dist(i,k_2)-1)} - (e^{-\alpha(dist(i,k_1)-1)} \times e^{-\alpha(dist(i,k_2)-1)})]}$$

Proof

P(crew in (j_1, j_2)|crew detected from i)

$$= \frac{\mathrm{P}(\text{crew in } (j_1, j_2) \wedge \text{crew detected from } i)}{\mathrm{P}(\text{crew detected from } i)}$$

$$= \frac{\mathrm{P}(\text{crew in } (j_1, j_2) \wedge \text{crew detected from } i)}{\sum_{(k_1,k_2)} \mathrm{P}(\text{crew in } (k_1, k_2) \wedge \text{crew detected from } i)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)\mathrm{P}\big(\text{crew detected from } i \big| \text{crew in } (j_1, j_2)\big)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big)\mathrm{P}\big(\text{crew detected from } i \big| \text{crew in } (k_1, k_2)\big)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) \times \left[e^{-\alpha(dist(i,j_1)-1)} + e^{-\alpha(dist(i,j_2)-1)} - \left(e^{-\alpha(dist(i,j_1)-1)} \times e^{-\alpha(dist(i,j_2)-1)}\right)\right]}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big) \times [e^{-\alpha(dist(i,k_1)-1)} + e^{-\alpha(dist(i,k_2)-1)} - (e^{-\alpha(dist(i,k_1)-1)} \times e^{-\alpha(dist(i,k_2)-1)})]}$$

If the bot doesn't detect the crew member

P(crew in (j_1, j_2)|crew not detected from i)

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) \times \left(1 - e^{-\alpha(dist(i,j_1)-1)}\right) \times \left(1 - e^{-\alpha(dist(i,j_2)-1)}\right)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big) \times (1 - e^{-\alpha(dist(i,k_1)-1)}) \times (1 - e^{-\alpha(dist(i,k_2)-1)})}$$

Proof

P(crew in (j_1, j_2)|crew not detected from i)

$$= \frac{\mathrm{P}(\text{crew in } (j_1, j_2) \wedge \text{crew not detected from } i)}{\mathrm{P}(\text{crew not detected from } i)}$$

$$= \frac{\mathrm{P}(\text{crew in } (j_1, j_2) \wedge \text{crew not detected from } i)}{\sum_{(k_1,k_2)} \mathrm{P}(\text{crew in } (k_1, k_2) \wedge \text{crew not detected from } i)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)\mathrm{P}\big(\text{crew not detected from } i \big| \text{crew in } (j_1, j_2)\big)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big)\mathrm{P}\big(\text{crew not detected from } i \big| \text{crew in } (k_1, k_2)\big)}$$

$$= \frac{\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) \times \left(1 - e^{-\alpha(dist(i,j_1)-1)}\right) \times \left(1 - e^{-\alpha(dist(i,j_2)-1)}\right)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew in } (k_1, k_2)\big) \times (1 - e^{-\alpha(dist(i,k_1)-1)}) \times (1 - e^{-\alpha(dist(i,k_2)-1)})}$$

The alien moves

Same as One crew member, one alien.

Two crew members, two aliens

We now store an alien probability for each pair of open cells.

Assume: Aliens can be in the same cell.

The number of pairs is $\binom{\text{number of open cells}+1}{2}$.

Initial conditions

$$\mathrm{P}\big(\text{crew in } (j_1, j_2)\big) = \begin{cases} 0 & \text{if } (j_1, j_2) \ni i \\ \dfrac{1}{\binom{\text{number of open cells}}{2} - (\text{number of open cells} - 1)} & \text{otherwise} \end{cases}$$

$$\mathrm{P}\big(\text{alien in } (j_1, j_2)\big) = \begin{cases} 0 & \text{if } j_1 \in DS(i) \text{ or } j_2 \in DS(i) \\ \dfrac{1}{\binom{\text{number of open cells outside } DS(i)+1}{2}} & \text{otherwise} \end{cases}$$

<u>Proof</u>

For proof of $\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)$ initial conditions, see Two crew members, one alien.

If the bot doesn’t find the alien

$$\mathrm{P}(\text{alien in } (j_1, j_2) | \text{alien not in } i) = \begin{cases} 0 & \text{if } (j_1, j_2) \ni i \\ \dfrac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)}{1 - \sum_{(k_1,k_2)\ni i} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big)} & \text{otherwise} \end{cases}$$

<u>Proof</u>

If $(j_1, j_2) \not\ni i$,

$$\mathrm{P}(\text{alien in } (j_1, j_2) | \text{alien not in } i) = \frac{\mathrm{P}(\text{alien in } (j_1, j_2) \wedge \text{alien not in } i)}{\mathrm{P}(\text{alien not in } i)}$$

$$= \frac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)\mathrm{P}\big(\text{alien not in } i \big| \text{alien in } (j_1, j_2)\big)}{1 - \mathrm{P}(\text{alien in } i)}$$

$$= \frac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)}{1 - \sum_{(k_1,k_2)\ni i} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big)}$$

If the bot doesn't find the crew member

Same as Two crew members, one alien.

If the bot detects the alien

$\mathrm{P}(\text{alien in }(j_1, j_2)|\text{alien detected from }i)$

$$= \begin{cases} \dfrac{\mathrm{P}\big(\text{alien in }(j_1, j_2)\big)}{\sum_{(k_1,k_2)\in DS_{in}(i)} \mathrm{P}\big(\text{alien in }(k_1, k_2)\big)} & \text{if } j_1 \in DS(i) \text{ or } j_2 \in DS(i) \\ 0 & \text{otherwise} \end{cases}$$

<u>Proof</u>

$\mathrm{P}(\text{alien in }(j_1, j_2)|\text{alien detected from }i)$

$$= \frac{\mathrm{P}(\text{alien in }(j_1, j_2) \wedge \text{alien detected from }i)}{\mathrm{P}(\text{alien detected from }i)}$$

$$= \frac{\mathrm{P}(\text{alien in }(j_1, j_2) \wedge \text{alien detected from }i)}{\sum_{(k_1,k_2)} \mathrm{P}(\text{alien in }(k_1, k_2) \wedge \text{alien detected from }i)}$$

$$= \frac{\mathrm{P}\big(\text{alien in }(j_1, j_2)\big)\mathrm{P}\big(\text{alien detected from }i\big|\text{alien in }(j_1, j_2)\big)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{alien in }(k_1, k_2)\big)\mathrm{P}\big(\text{alien detected from }i\big|\text{alien in }(k_1, k_2)\big)}$$

$$= \frac{\mathrm{P}\big(\text{alien in }(j_1, j_2)\big)\mathrm{P}\big(\text{alien detected from }i\big|\text{alien in }(j_1, j_2)\big)}{\sum_{(k_1,k_2)\in DS_{in}(i)} \mathrm{P}\big(\text{alien in }(k_1, k_2)\big)}$$

$$\mathrm{P}\big(\text{alien detected from }i\big|\text{alien in }(j_1, j_2)\big) = \begin{cases} 1 & \text{if } j_1 \in DS(i) \text{ or } j_2 \in DS(i) \\ 0 & \text{otherwise} \end{cases}$$

If the bot doesn't detect the alien

P(alien in (j_1, j_2)|alien not detected from i)

$$= \begin{cases} \dfrac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)}{\sum_{(k_1,k_2)\in DS_{out}(i)} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big)} & \text{if } j_1 \notin DS(i) \text{ and } j_2 \notin DS(i) \\ 0 & \text{otherwise} \end{cases}$$

Proof

P(alien in (j_1, j_2)|alien not detected from i)

$$= \frac{\mathrm{P}(\text{alien in } (j_1, j_2) \wedge \text{alien not detected from } i)}{\mathrm{P}(\text{alien not detected from } i)}$$

$$= \frac{\mathrm{P}(\text{alien in } (j_1, j_2) \wedge \text{alien not detected from } i)}{\sum_{(k_1,k_2)} \mathrm{P}(\text{alien in } (k_1, k_2) \wedge \text{alien not detected from } i)}$$

$$= \frac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)\mathrm{P}\big(\text{alien not detected from } i \big| \text{alien in } (j_1, j_2)\big)}{\sum_{(k_1,k_2)} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big)\mathrm{P}\big(\text{alien not detected from } i \big| \text{alien in } (k_1, k_2)\big)}$$

$$= \frac{\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)\mathrm{P}\big(\text{alien not detected from } i \big| \text{alien in } (j_1, j_2)\big)}{\sum_{(k_1,k_2)\in DS_{out}(i)} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big)}$$

$$\mathrm{P}\big(\text{alien not detected from } i \big| \text{alien in } (j_1, j_2)\big) = \begin{cases} 1 & \text{if } j_1 \notin DS(i) \text{ and } j_2 \notin DS(i) \\ 0 & \text{otherwise} \end{cases}$$

If the bot detects the crew member

Same as Two crew members, one alien.

If the bot doesn't detect the crew member

Same as Two crew members, one alien.

The aliens move

Assume: The aliens must move up, down, left, or right and cannot stay in place.

$$\mathrm{P}\big(\text{alien in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)\in adj((j_1,j_2))} \begin{matrix} \mathrm{P}\big(\text{alien in } (k_1, k_2)\big) \\ \times \dfrac{1}{\text{number of open cells adjacent to } k_1} \\ \times \dfrac{1}{\text{number of open cells adjacent to } k_2} \end{matrix}$$

<u>Proof</u>

$$\mathrm{P}\big(\text{alien now in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)} \mathrm{P}\big(\text{alien was in } (k_1, k_2) \wedge \text{alien now in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)} \mathrm{P}\big(\text{alien was in } (k_1, k_2)\big)\mathrm{P}\big(\text{alien now in } (j_1, j_2) \big| \text{alien was in } (k_1, k_2)\big)$$

$$= \sum_{(k_1,k_2)\in adj((j_1,j_2))} \begin{matrix} \mathrm{P}\big(\text{alien was in } (k_1, k_2)\big) \\ \times \dfrac{1}{\text{number of open cells adjacent to } k_1} \\ \times \dfrac{1}{\text{number of open cells adjacent to } k_2} \end{matrix}$$

$$\mathrm{P}\big(\text{alien now in } (j_1, j_2) \big| \text{alien was in } (k_1, k_2)\big)$$

$$= \begin{cases} \dfrac{1}{\text{number of open cells adjacent to } k_1} \times \dfrac{1}{\text{number of open cells adjacent to } k_2} & \text{if } (k_1, k_2) \in adj\big((j_1, j_2)\big) \\ 0 & \text{otherwise} \end{cases}$$

Question 2

Implementation

Spaceship

A spaceship is a `Spaceship` object.

A `Spaceship` object has the following fields:

`grid_size`

Integer.

The number of rows (or the number of columns) of the grid.

`grid`

Two-dimensional boolean array of size `grid_size` × `grid_size`.

The grid. `grid[x, y]` is True if the cell is open and False if the cell is closed.

`num_open_cells`

Integer.

The number of open cells in the grid.

`grid_neighbors`

(Can be thought of as) Two-dimensional array of size `grid_size` × `grid_size`.

`grid_neighbors[x, y]` is a list of the open neighbors of `grid[x, y]`.

`pairs_neighboring_pairs`

Dictionary in which key is pair of open cells and value is list of neighboring pairs of open cells. Specifically, key is tuple of tuples `((x1, y1), (x2, y2))` and value is list of tuples of tuples `[((x3, y3), (x4, y4)), ((x5, y5), (x6, y6)), ...]`. This field is initialized only if the number of aliens is 2.

grid_prob_alien_move_neighbor

Two-dimensional float array of size grid_size × grid_size.

grid_prob_alien_move_neighbor[x, y] is the probability that an alien moves to grid[x, y]. In other words,

$$\frac{1}{\text{number of open cells adjacent to grid[x,y]}}$$

distances

Nested dictionary that stores distances between all pairs of open cells. Key is tuple (x1, y1) and value is dictionary in which key is tuple (x2, y2) and value is distance between (x1, y1) and (x2, y2). In particular, distances[(x1, y1)][(x2, y2)] returns the distance between (x1, y1) and (x2, y2).

Bot

A bot is a tuple (x, y), where (x, y) are the indices of the bot. To check whether a bot is in the same cell as an alien, check whether the bot tuple is in the list of alien tuples. To check whether a bot is in the same cell as a crew member, check whether the bot tuple is in the list of crew member tuples.

Crew members

The crew members are a list of tuples (crew members). A crew member is a tuple (x, y), where (x, y) are the indices of the crew member. To check whether a bot is in the same cell as a crew member, check whether the bot tuple is in the list of crew member tuples.

Aliens

The aliens are a list of tuples (aliens). An alien is a tuple (x, y), where (x, y) are the indices of the alien. To check whether a bot is in the same cell as an alien, check whether the bot tuple is in the list of alien tuples.

Probabilities

Notation

grid_size refers to spaceship.grid_size. grid refers to spaceship.grid.

One crew member

prob_crew_1 is a two-dimensional array of size grid_size × grid_size. prob_crew_1[x, y] is the probability that the crew member is in grid[x, y]. The sum of prob_crew_1 is 1.

Two crew members

`prob_crew_2` is a four-dimensional array of size `grid_size` × `grid_size` × `grid_size` × `grid_size`. If x1 != x2 and y1 != y2, `prob_crew_2[x1, y1, x2, y2]` is the probability that one of the crew members is in `grid[x1, y1]` and the other crew member is in `grid[x2, y2]`. Otherwise, `prob_crew_2[x1, y1, x2, y2]` is 0. (Crew members cannot be in the same cell.) Because `prob_crew_2` is symmetric, the sum of `prob_crew_2` is 2.

One alien

`prob_alien_1` is a two-dimensional array of size `grid_size` × `grid_size`. `prob_alien_1[x, y]` is the probability that the alien is in `grid[x, y]`. The sum of `prob_alien_1` is 1.

Two aliens

`prob_alien_2` is a four-dimensional array of size `grid_size` × `grid_size` × `grid_size` × `grid_size`. If x1 != x2 and y1 != y2, `prob_alien_2[x1, y1, x2, y2]` is the probability that one of the aliens is in `grid[x1, y1]` and the other alien is in `grid[x2, y2]`. Otherwise, `prob_alien_2[x1, y1, x2, y2]` is double the probability that both of the aliens are in `grid[x1, y1]`. Because `prob_alien_2` is symmetric, the sum of `prob_alien_2` is 2.

Optimizations

Due to the time-consuming computations, several optimizations have been implemented to improve the efficiency of the code.

`distances` dictionary

At each time step, the distance between the bot and each crew member must be computed for two reasons:

1. To determine whether the bot detects a crew member
2. To update the crew probabilities based on whether the bot detects a crew member

The distance between any two open cells is constant. When a spaceship is initialized, the distances between all pairs of open cells are computed and stored in a nested dictionary called `distances`. In particular, `distances[(x1, y1)][(x2, y2)]` returns the distance between `(x1, y1)` and `(x2, y2)`.

Boolean indexing

At each time step, the probabilities must be updated for each open cell. A simple solution is to iterate over all cells in a nested for loop. This is computationally inefficient because this results in a run time of $O(n^2)$ for one crew member or one alien and a run time of $O(n^4)$ for two crew members or two aliens.

In the given implementation, the probability update functions make heavy use of boolean indexing, specifically for updating the alien probabilities. This reduces the time needed to update the probabilities.

`grid_neighbors` two-dimensional array

For one alien, when the alien moves, the alien probabilities are updated. This update to the alien probabilities is a function of the neighboring cells, for each cell. (See “The alien moves” on page 8.) The neighboring cells of each cell are constant. When a spaceship is initialized, the neighboring cells of all cells are computed, and lists of the neighboring cells are stored in a two-dimensional array called `grid_neighbors`. In particular, `grid_neighbors[x, y]` returns a list of the open neighbors of `grid[x, y]`.

`pairs_neighboring_pairs` dictionary

For two aliens, when the aliens move, the alien probabilities are updated. This update to the alien probabilities is a function of the neighboring pairs of cells, for each pair of cells. (See “The aliens move” on page 15.) The neighboring pairs of cells of each pair of cells are constant. When a spaceship is initialized, the neighboring pairs of all pairs are computed, and lists of the neighboring pairs are stored in a dictionary called `pairs_neighboring_pairs`. In particular, `pairs_neighboring_pairs[((x1, y1), (x2, y2))]` returns a list of the neighboring pairs of the pair `(x1, y1)` and `(x2, y2)`.

`grid_prob_alien_move_neighbor` two-dimensional array

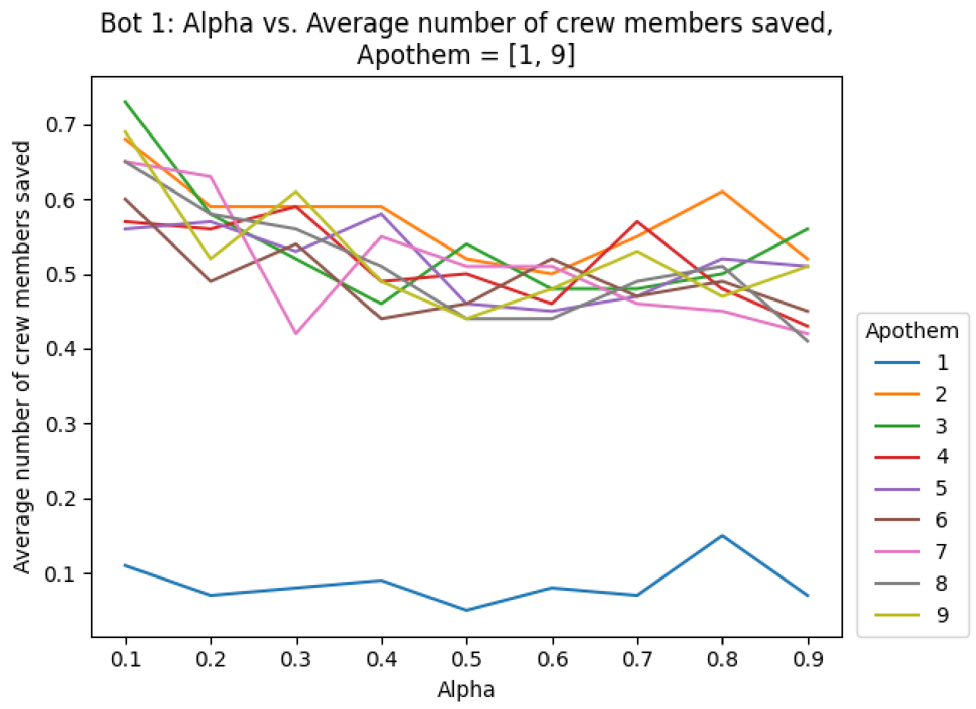
For both one alien and two aliens, when the aliens move, the alien probabilities are updated. As part of this update, the probability that an alien moves to a specific cell must be computed, for each cell. (See “The alien moves” on page 8 and “The aliens move” on page 15.) This probability is constant for each cell. When a spaceship is initialized, the probability that an alien moves to each cell is computed and stored in a two-dimensional array called `grid_prob_alien_move_neighbor`. In particular, `grid_prob_alien_move_neighbor[x, y]` returns the probability that an alien moves to `grid[x, y]`.

Question 3

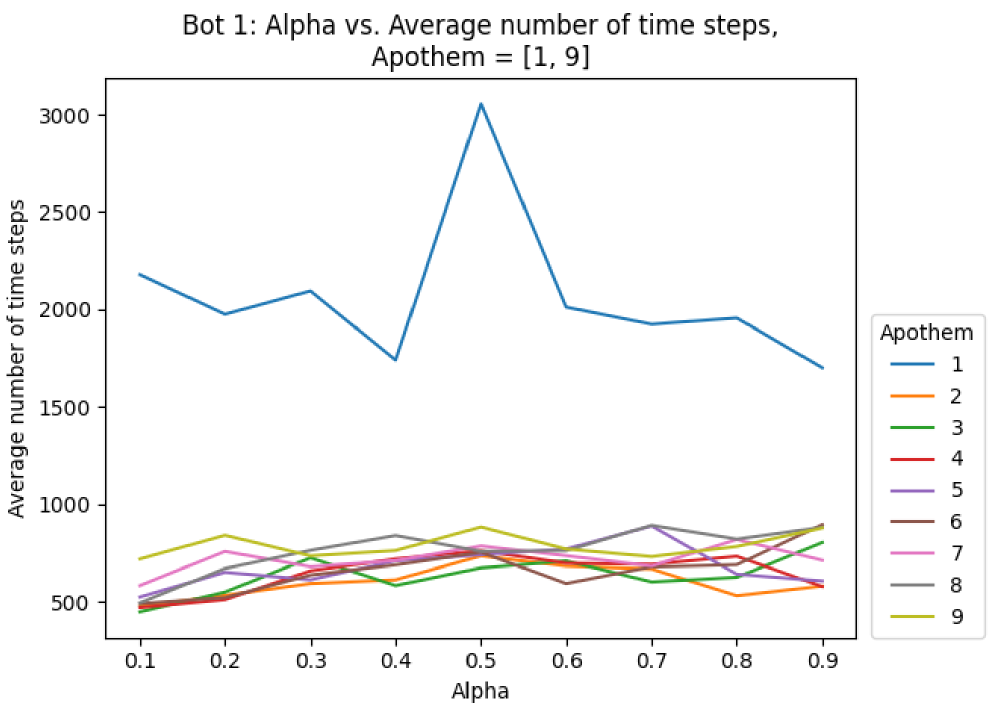
Apothem

For Bot 1, I ran 100 trials on a 20 × 20 spaceship for apothem = [1, 2, 3, 4, 5, 6, 7, 8, 9] and alpha = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Each trial was run with a random spaceship.

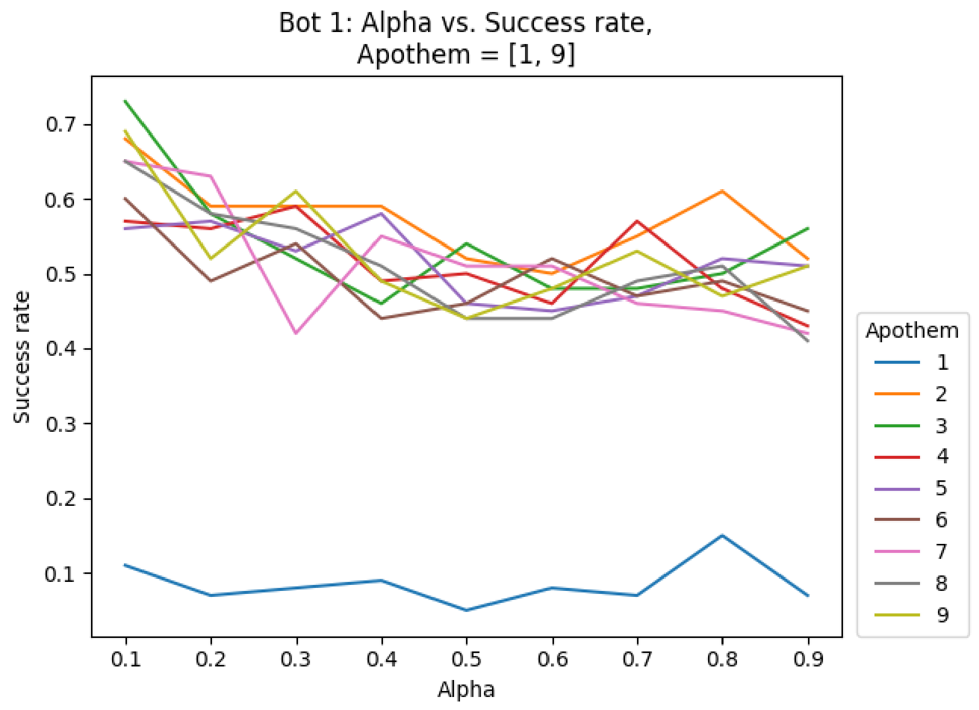
The running time was 4 hours, 10 minutes.



In terms of average number of crew members saved, apothem = 1 has the worst performance, and apothem = [2, 3, 4, 5, 6, 7, 8, 9] have about the same performance.



In terms of average number of time steps, apothem = 1 has the best performance, and apothem = [2, 3, 4, 5, 6, 7, 8, 9] have about the same performance.

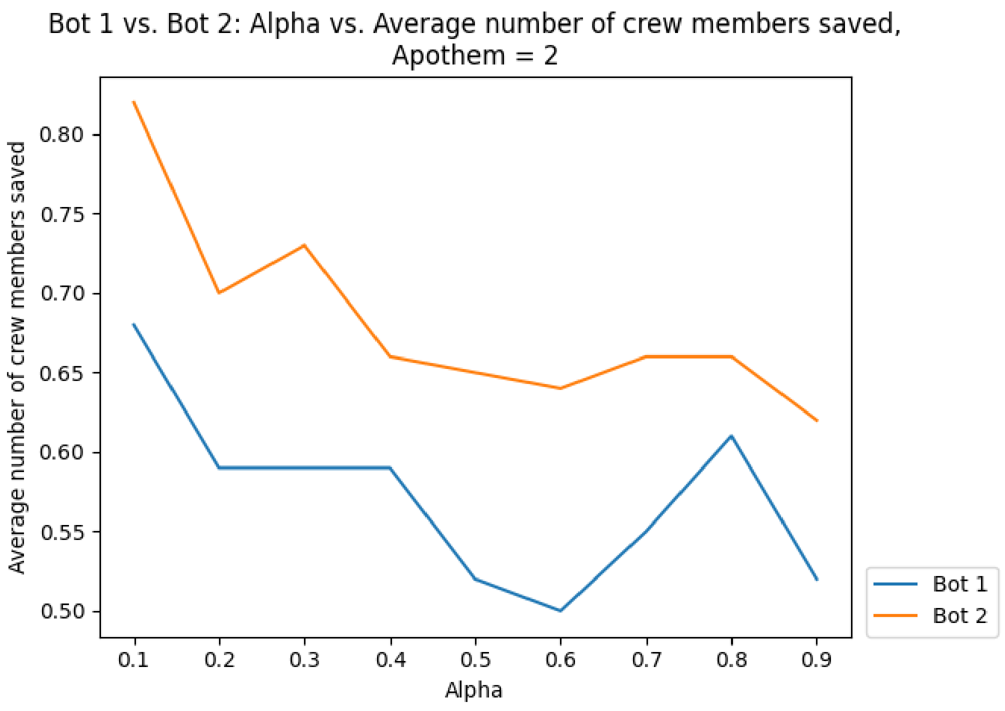


In terms of success rate, apothem = 1 has the worst performance, and apothem = [2, 3, 4, 5, 6, 7, 8, 9] have about the same performance.

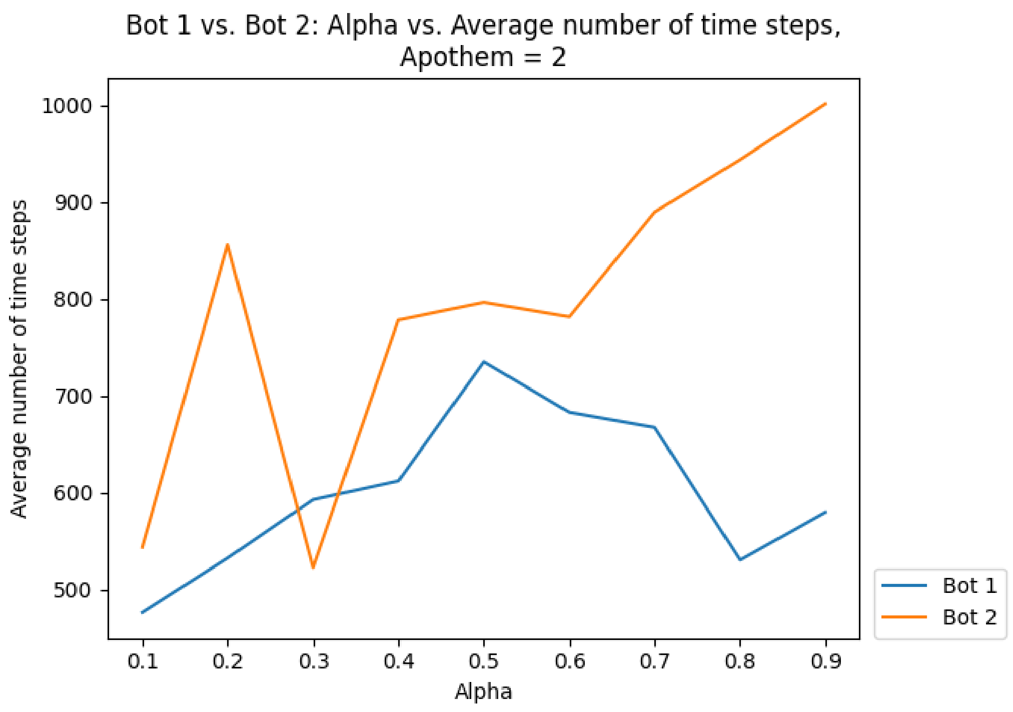
Bot 1 vs. Bot 2

For Bot 1 and Bot 2, I ran 100 trials on a 20 × 20 spaceship for apothem = 2 and alpha = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Each trial was run with a random spaceship.

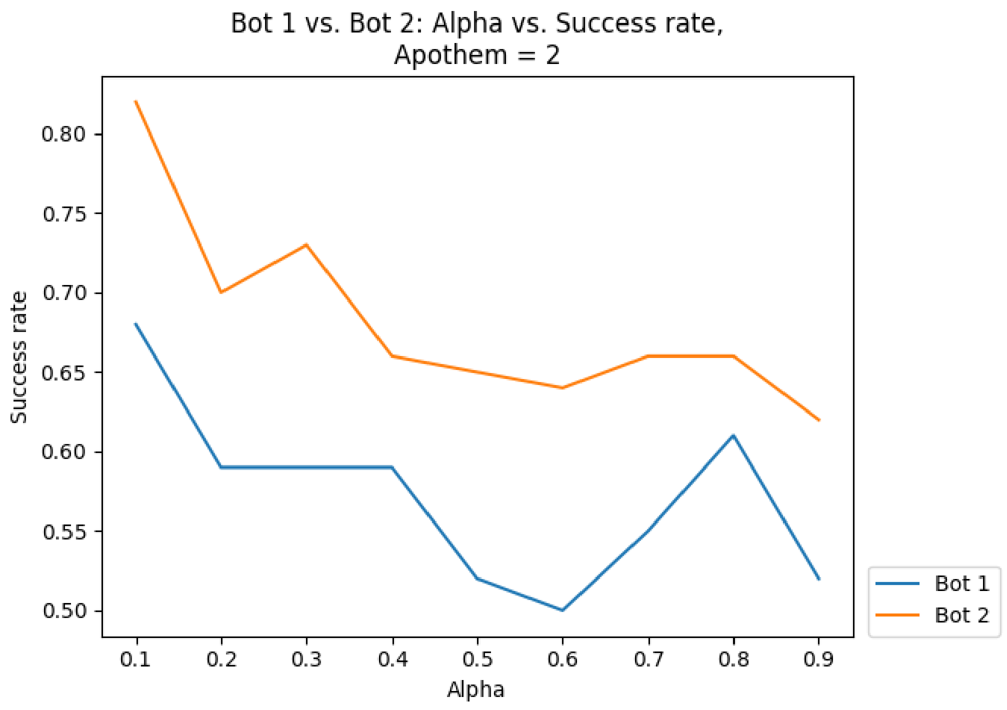
	Running time
Bot 1	
Bot 2	30 minutes



In terms of average number of crew members saved, Bot 2 performs better than Bot 1.



In terms of average number of time steps, Bot 2 performs better than Bot 1.



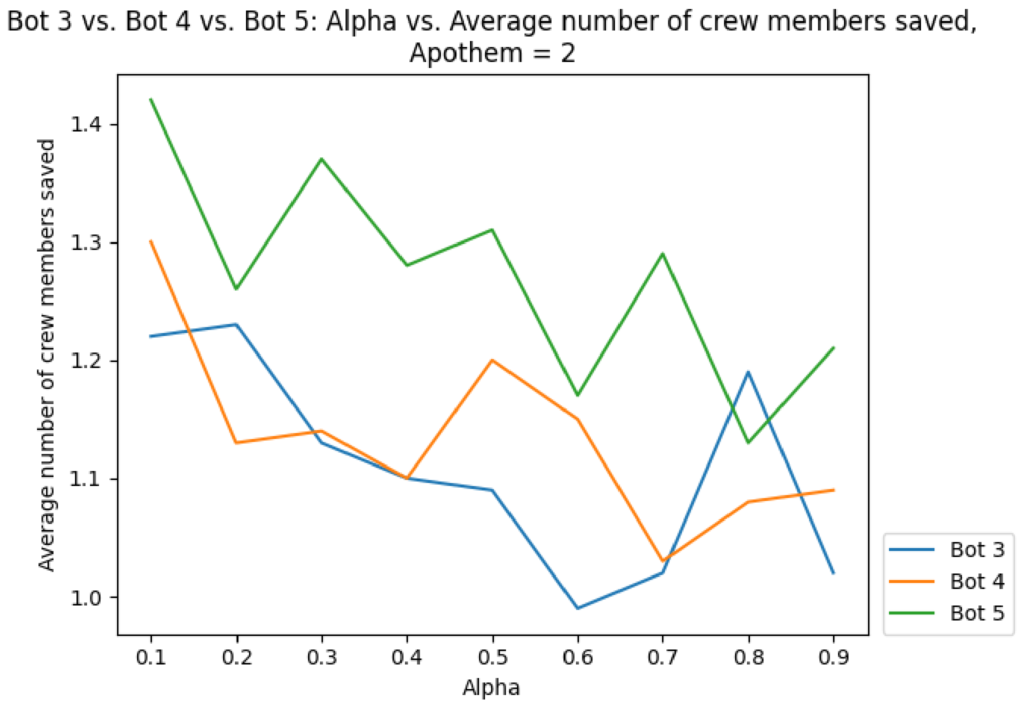
In terms of success rate, Bot 2 performs better than Bot 1.

Bot 3 vs. Bot 4 vs. Bot 5

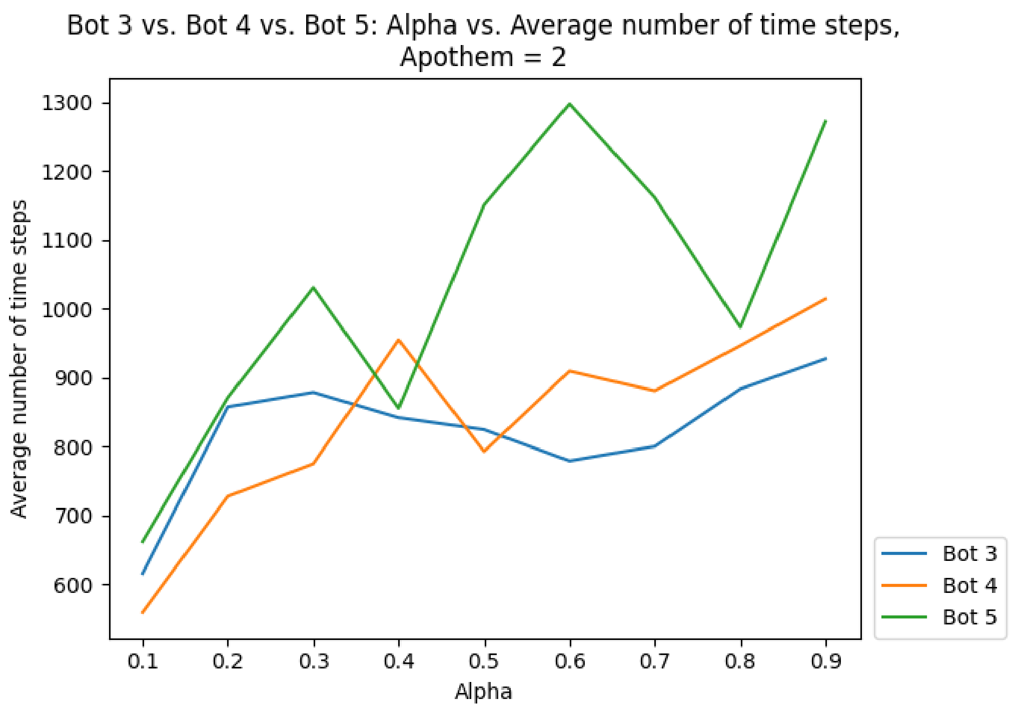
For Bot 3, Bot 4, and Bot 5, I ran 100 trials on a 20 × 20 spaceship for apothem = 2 and alpha = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Each trial was run with a random spaceship.

	Running time
Bot 3	30 minutes
Bot 4	50 minutes
Bot 5	55 minutes

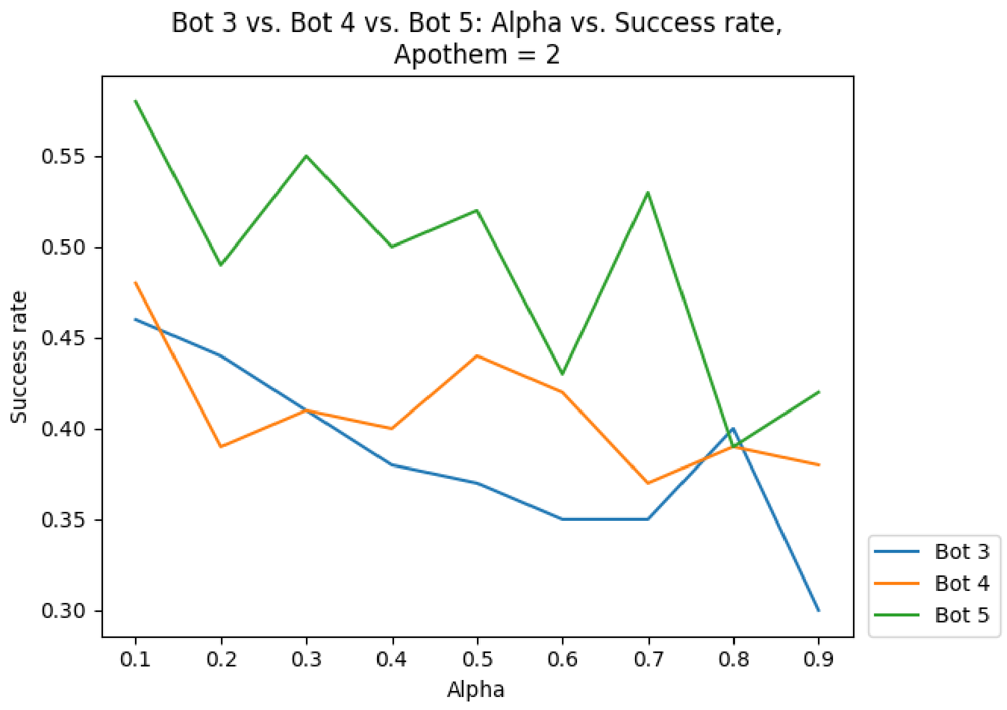
Bot 3 ran the fastest, Bot 4 ran the second fastest, and Bot 5 ran the slowest.



In terms of average number of crew members saved, Bot 5 has the best performance, Bot 4 has the second best performance, and Bot 3 has the worst performance.



In terms of average number of time steps, Bot 5 has the best performance, and Bot 3 and Bot 4 have about the same performance.



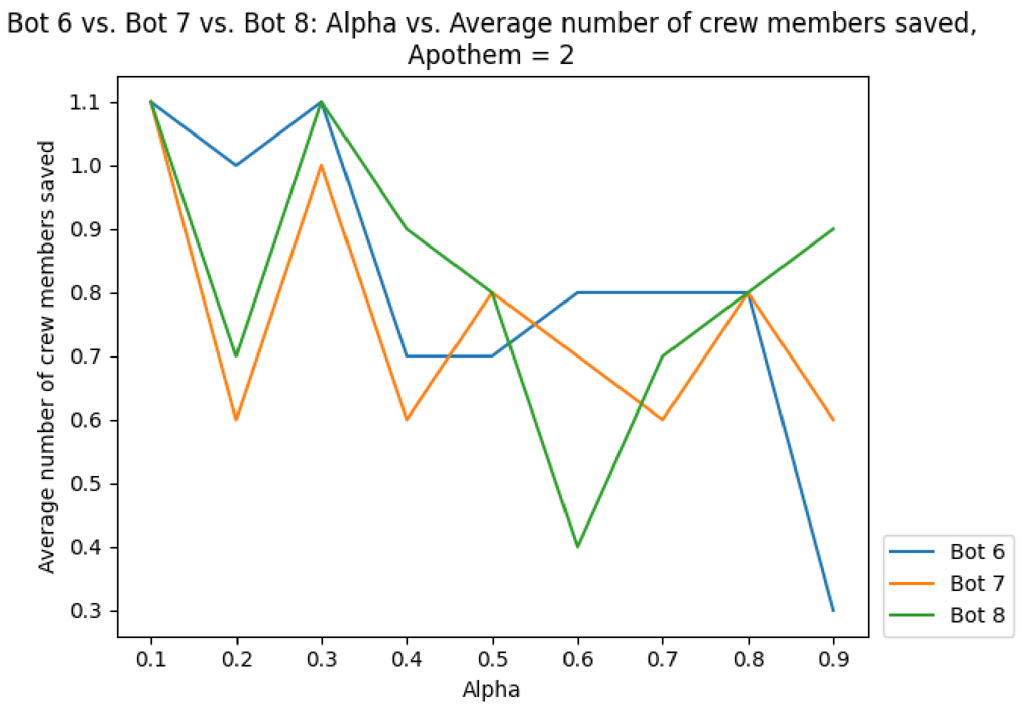
In terms of success rate, Bot 5 has the best performance, Bot 4 has the second best performance, and Bot 3 has the worst performance.

Bot 6 vs. Bot 7 vs. Bot 8

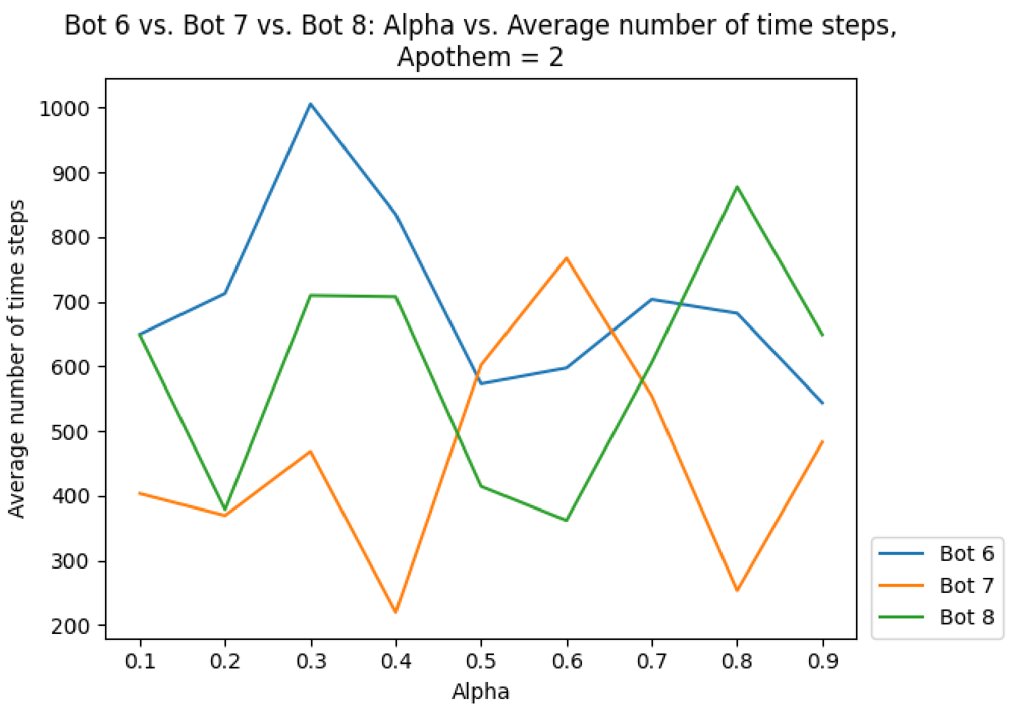
For Bot 6, Bot 7, and Bot 8, I ran 10 trials on a 20 × 20 spaceship for apothem = 2 and alpha = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Each trial was run with a random spaceship.

	Running time
Bot 6	5 minutes
Bot 7	1 hour, 55 minutes
Bot 8	2 hours, 20 minutes

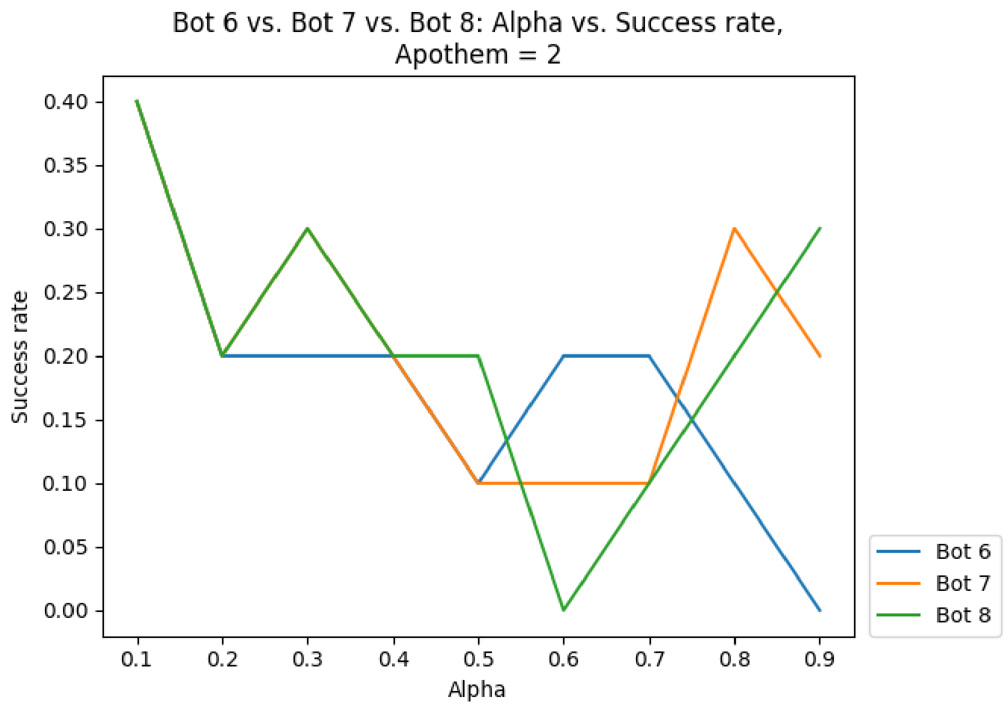
Bot 6 ran the fastest, Bot 7 ran the second fastest, and Bot 8 ran the slowest.



In terms of average number of crew members saved, Bot 6 and Bot 8 have about the same performance, and Bot 7 has the worst performance.



In terms of average number of time steps, Bot 6 has the best performance, Bot 8 has the second best performance, and Bot 7 has the worst performance.



In terms of success rate, Bot 7 and Bot 8 have about the same performance, and Bot 6 has the worst performance.

Simplifications

Due to limitations in time and compute,

- The grid size was reduced from 50 × 50 to 20 × 20.
- For Bot 6, Bot 7, and Bot 8, the number of trials was reduced from 100 to 10.

Question 4

The ideal bot could run simulations to decide what to do next. Because the alien moves randomly, the ideal bot could run simulations for the next time step in which the position of the alien may vary. The ideal bot could then move to the cell least likely to contain the alien at the next time step based on the simulations.

Question 5

We should update the crew member probabilities after the crew members move.

One crew member: The crew member moves

Assume: The crew member must move up, down, left, or right and cannot stay in place.

$$\mathrm{P}(\text{crew in } j) = \sum_{k \in adj(j)} \mathrm{P}(\text{crew in } k) \cdot \frac{1}{\text{number of open cells adjacent to } k}$$

<u>Proof</u>

$$\begin{aligned}
\mathrm{P}(\text{crew now in } j) &= \sum_{k} \mathrm{P}(\text{crew was in } k \wedge \text{crew now in } j) \\
&= \sum_{k} \mathrm{P}(\text{crew was in } k)\mathrm{P}(\text{crew now in } j | \text{crew was in } k) \\
&= \sum_{k \in adj(j)} \mathrm{P}(\text{crew was in } k) \cdot \frac{1}{\text{number of open cells adjacent to } k}
\end{aligned}$$

$$\mathrm{P}(\text{crew now in } j | \text{crew was in } k) = \begin{cases} \dfrac{1}{\text{number of open cells adjacent to } k} & \text{if } k \in adj(j) \\ 0 & \text{otherwise} \end{cases}$$

Two crew members: The crew members move

Assume: The crew members must move up, down, left, or right and cannot stay in place.
Assume: Crew members can be in the same cell.

$$\mathrm{P}\big(\text{crew in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)\in adj((j_1,j_2))} \begin{aligned} &\mathrm{P}\big(\text{crew in } (k_1, k_2)\big) \\ &\times \frac{1}{\text{number of open cells adjacent to } k_1} \\ &\times \frac{1}{\text{number of open cells adjacent to } k_2} \end{aligned}$$

<u>Proof</u>

$$\mathrm{P}\big(\text{crew now in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew was in } (k_1, k_2) \wedge \text{crew now in } (j_1, j_2)\big)$$

$$= \sum_{(k_1,k_2)} \mathrm{P}\big(\text{crew was in } (k_1, k_2)\big)\mathrm{P}\big(\text{crew now in } (j_1, j_2) \big| \text{crew was in } (k_1, k_2)\big)$$

$$= \sum_{(k_1,k_2)\in adj((j_1,j_2))} \begin{aligned} &\mathrm{P}\big(\text{crew was in } (k_1, k_2)\big) \\ &\times \frac{1}{\text{number of open cells adjacent to } k_1} \\ &\times \frac{1}{\text{number of open cells adjacent to } k_2} \end{aligned}$$

$$\mathrm{P}\big(\text{crew now in } (j_1, j_2) \big| \text{crew was in } (k_1, k_2)\big)$$

$$= \begin{cases} \dfrac{1}{\text{number of open cells adjacent to } k_1} \times \dfrac{1}{\text{number of open cells adjacent to } k_2} & \text{if } (k_1, k_2) \in adj\big((j_1, j_2)\big) \\ 0 & \text{otherwise} \end{cases}$$