CS - 520
Introduction to Artificial Intelligence
Project 2

By:

Deviram Kondaveti (dk1273)

Vaishnavi Madhavaram (vm695)

Puja Sridhar (ps1393)

PROBLEM STATEMENT:

The bot navigates through the open cells in a ship of size 35 X 35 with the objective to save the crew members and avoiding the aliens in the process. The bot gets updates from two sensors that detect the presences of aliens and crew members within a certain range of cells that are determined by values of k and alpha respectively. The assumptions that are made while implementing this project is given below:

- The sensor beeps if the alien is present within (2k+1) X (2k+1) squared centred at the bot's location.
- The location of the crew member is unknown to the Bot, and it receives a beep if it is d - distance away with a probability of $e^{-\propto(d-1)}$.
- The Bot uses a **probabilistic approach** because the location of alien and crew members on the ship layout is unknown to the Bot.

The project has three scenarios with varying number of aliens and crew members:

- One Alien and One Crew Member
- One Alien and Two Crew Members
- Two Aliens and Two Crew Members

As the number of aliens and crew members are changing, we are using a knowledge base to update the existing knowledge with the new knowledge as the Bot keeps progressing through the open cells in the ship to find the crew members by avoiding the aliens.

1. SETUP

- **Ship:** The Aliens, Bot and the crew members are placed on a 35 X 35 grid.
- **Bot:** The Bot can move in four directions (Up/Down/Left/Right). The Bot can access data from two sensors, where one detects the presence of aliens in a (2k+1) X (2k+1) square centred at its location and the other sensor detects the presence of crew members at d – distance.
- **Alien:** The Alien can move randomly to any adjacent empty cell every timestep. If the alien and the Bot occupy the same cell, then the execution is over.
- **One Alien, One Crew:** In this scenario, the Bot starts of with a belief that the alien is not within the buffer set and both the crew member and the alien are placed in any one of the open cells of the ship.
- **One Alien, Two Crew:** In this scenario, there is an addition of another crew member to be rescued by the Bot. The Bot's strategy is to take into account the presence of two crew members.
- **Two Aliens, Two Crew:** In this scenario, there is an addition of another alien to be avoided by the Bot. The Bot's strategy is to take into account the presence of two aliens and two crew members.

2. BOT CHALLENGES

- The ship's layout is in a grid format containing cells that could be either open or closed. The Bot is restricted to move only in any of the four directions (Up/Down/Right/Left).
- The primary objective of the Bot is to find the shortest path to save the crew member and avoid the aliens that move dynamically during the process.

- Initially the alien is placed in any one of the open cells outside the buffer zone and the crew is placed randomly on any open cell apart from the initial Bot cell.
- The aliens dynamically move through the open cells in the ship. The Bot has to be sure that it doesn't enter the same cell as the alien or the alien entering the Bot's cell. This can be only done if the Bot uses the information it receives from the alien detection sensor and also updating the probability of the alien moving to a cell i from a current alien position.
- The location of the crew member is updated based on the information received by the Bot from the crew member detection sensor. If the Bot receives a beep, then the crew member is likely to be at d-distance away and if the Bot makes a move in order to get closer to the crew member but this time it fails to sense a beep, this indicates that the crew member isn't anywhere at d-distance. This information is updated in knowledge base stating that the crew member is likely to be in any cell nearby the current Bot's cell.

3. DESIGN FRAMEWORK

- The basic objective is to build different Bots with a unique parameter whose ultimate focus is to save the crew members and avoid the aliens. The main challenge faced during the implementation phase was in updating the knowledge base of the Bot with the alien movements and the likelihood of the crew member being in cell i.
- The knowledge base of the Bot is updated using Belief Networks based on the inference made by the Bot using the alien detection sensor and the crew member detection sensor. The knowledge base contains information of each time step on the likelihood of both the alien and crew member being in a particular cell. Based on this the Bot is made to make a calculated movement.

4. DETECTION AND NAVIGATION PROCESS

- All the bots are implemented using A* algorithm to find the path from the bot to the captain. A* strikes a balance between time and space complexity, utilizing a consistent heuristic to guide the search efficiently. This informed search algorithm ensures not only the discovery of the optimal path but also considers the cost of reaching the goal, making it a suitable choice for navigating through the complex and changing ship environment, avoiding aliens, and successfully rescuing the captain.
- To update the knowledge base of the Bots, we have used Belief Network which is based on the probabilistic values of the alien/crew member in a given cell i. The knowledge base is updated every time step enabling the Bot to predict the next alien movement and the likelihood of the crew member in the nearby cells.
- The use of probabilistic values to determine the presence of aliens and the crew members in a specific range of cells determined by the value of k for aliens and d-distance for the crew members.
- The Belief Network acts as a learning mechanism as the probability values of alien/crew being in a cell i gets updated every timestep enabling the Bots to adapt to the dynamic environment.
- The stimulation is run multiple times for various alpha and k values making it a challenge to improve the success rate.
- The alien movements add an unpredictability factor because it is dynamic and probabilistic. Thereby, poses as a challenge to the Bot in the process of rescuing the crew member.

- The main objective of the Bot is to save the crew member and during the process avoid the aliens in order to increase the success rate.

KNOWLEDGE BASE:

1. INITIATING THE KNOWLEDGE BASE:

The Belief network, represented as a 2-D matrix or array, holds probabilities for the presence of Alien and Crew in each cell. This network is continually adjusted once the bot or alien moves. To begin with, the belief network is initialized by assigning probabilities to each cell indicating the chances of alien and crew presence.

ALIEN:

We know that the Alien cannot occupy a closed cell or the same cell as the Bot in the ship. Hence the Belief network will be initialized as follows:

$$\boldsymbol{P(Alien\ in\ cell\ j)\ \forall\ cell\ j} = \frac{1}{(\text{Total number of cells} - \text{number of closed cells} - 1)} = \frac{1}{(\text{Number of Open cells} - 1)}$$

AlienBelief[] is the probability matrix for Aliens.

CREW MEMBER:

The probability matrix for a Crew member is also constructed similar to the probability matrix of the Alien and updates the Belief network after the Bot has made a movement.

$$\boldsymbol{P(Crew\ in\ cell\ j)\ \forall\ cell\ j} = \frac{1}{(\text{Total number of cells} - \text{number of closed cells} - 1)} = \frac{1}{(\text{Number of Open cells} - 1)}$$

CrewBelief[] is the probability matrix for Crew members.

2. KNOWLEDGE BASE UPDATION:

We maintain belief networks to track the probabilities of both alien and crew presence within the grid environment. These networks are crucial for guiding the decisions and actions of our bot system. Following the occurrence of significant events such as bot movement, alien detection, crew beep detection, and alien movement, we update the belief networks using specific probability formulas.

- Whenever the **Bot moves** within the grid, it triggers an update in the belief networks. Bot movement alters the probabilities of alien and crew presence in various grid cells, prompting adjustments to our belief about their potential locations.

- Upon executing an **Alien detection** procedure, our bot receives readings from the sensor. We consider both true and false readings from the sensor, which influence our belief regarding the presence of aliens in different parts of the grid. These sensor readings contribute to the refinement of our belief network for aliens.
- When the **Bot listens to a crew beep**, it provides valuable information about the potential presence of crew members nearby. Whether the bot successfully receives the beep or not, this event triggers updates in our belief network for crew presence, influencing our estimations of crew locations within the grid.
- Following any **movement made by the Alien** within the grid, we update the belief network to reflect the altered probabilities of alien presence in different grid cells. Alien movement affects our understanding of where aliens may be located, prompting adjustments to our belief network accordingly.

The Knowledge Base is updated by using 3 standard probability formulas, which are:

MARGINALIZATION: $P(A) = P(A \wedge B) + P(A \wedge \neg B)$

CONDITIONAL FACTORING: $P(A \wedge B) = P(A) * P(B|A)$

CONDITIONAL PROBABILITY: $P(A|B) = P(A \wedge B) / P(B)$

3. BOT'S DECISION MAKING:

At each step, the bot decides its movement strategy by giving priority to cells where it suspects crew members might be present. When faced with multiple options that have comparable probabilities, the bot chooses randomly among them. It avoids cells where it is confident that aliens are present.

The bot fails if it occupies the same cell as an alien. The bot succeeds when it successfully reaches the cell of a crew member

BOT MODEL AND PROBABILITY UPDATE:

SCENARIO 1 – ONE ALIEN, ONE CREW

In this scenario the Bot has to save one crew member and avoid one alien. The calculations required for updating the Belief Network is explained below:

STEP 1: START OF THE STIMULATION:

Initially we have been given with the information that the Bot is at cell j and neither the alien or the crew member should be present at the initial Bot position. In addition to that the alien is placed outside the $(2k + 1) \times (2k + 1)$ region with cell i as the centre of the region. We have two probability matrices: one for the presence of aliens in each cell and another for the presence of the crew in each cell. These matrices are initialized as zero matrices, indicating that there is no initial belief regarding the presence of aliens or the crew in any cell.

From the above statement, we can write it mathematically as

$\boldsymbol{P(alien\ in\ cell\ j)\ \forall\ cell\ j = 0}$

$\boldsymbol{P(Crew\ in\ cell\ j)\ \forall\ cell\ j = 0}$

STEP 2: ALIEN DETECTION:

The first step involves activating the alien detection sensor, which scans a grid of the ship with the current position of the bot at its centre. This grid has dimensions of $(2k + 1)\ \times\ (2k + 1)$, where k represents the detection range.

(a) Alien in Detection Square or Alien Beep is heard

Calculating probability of Alien in each cell given Alien being in the Detection Square

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j)$$

By Conditional Probability:

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j) = \frac{P(Alien\ in\ cell\ i\ \wedge Alien\ beep\ in\ cell\ j)}{P(Alien\ beep\ in\ cell\ j)}$$

By Conditional Factoring:

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j) = \frac{P(Alien\ in\ cell\ i\ ) * P(Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ )}{P(Alien\ beep\ in\ cell\ j)}$$

Now for calculating $P(Alien\ beep\ in\ cell\ j)$, the probability of alien beep being heard when Bot is in cell j, we will be adding probabilities of alien in each cell inside detection square.

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j) = \frac{AlienBelief[i] * P(Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ )}{\sum_k P(Alien\ in\ cell\ k)}$$

- Here ‘k’ is the cells in the alien detection square
- If cell ‘i’ is any cell in the alien detection square, then $P(Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ ) = 1$

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j) = \frac{AlienBelief[i] * 1}{\sum_k AleinBelief[k]}$$

- If cell 'i' is any cell outside the alien detection square, then $P(Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ ) = 0$

$$P(Alien\ in\ cell\ i \mid Alien\ beep\ in\ cell\ j) = 0$$

This analysis enables us to evaluate the probability of alien presence in specific cells based on detected alien beeps. Integrating this probabilistic approach into our detection system enhances its precision and effectiveness in identifying potential alien activity during the detection operations.

(b) <u>Alien not in Detection Square or Alien Beep is not heard</u>

AlienBelief[] is the probability matrix for Aliens.

Calculating probability of Alien in each cell given Alien not being in the Detection Square

$$P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j)$$

By Conditional Probability:

$$P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j) = \frac{P(Alien\ in\ cell\ i\ \wedge No\ Alien\ beep\ in\ cell\ j)}{P(No\ Alien\ beep\ in\ cell\ j)}$$

By Conditional Factoring:

$$P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j)$$

$$= \frac{P(Alien\ in\ cell\ i\ ) * P(No\ Alien\ beep\ in\ cell\ j \mid No\ Alien\ in\ cell\ i\ )}{P(No\ Alien\ beep\ in\ cell\ j)}$$

Now for calculating $P(No\ Alien\ beep\ in\ cell\ j)$, the probability of no alien beep being heard when Bot is in cell j, we will be adding probabilities of alien in each cell outside detection square.

$$P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j)$$

$$= \frac{AlienBelief[i] * P(No\ Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ )}{\sum_k P(Alien\ in\ cell\ k)}$$

- Here 'k' is the cells outside the alien detection square

- If cell 'i' is any cell in the alien detection square, then
  $P(No\ Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ ) = 0$

$$\boxed{P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j) = 0}$$

- If cell 'i' is any cell outside the alien detection square, then
  $P(No\ Alien\ beep\ in\ cell\ j \mid Alien\ in\ cell\ i\ ) = 1$

$$\boxed{P(Alien\ in\ cell\ i \mid No\ Alien\ beep\ in\ cell\ j) = \frac{AlienBelief[i] * 1}{\sum_k AleinBelief[k]}}$$

This analysis enables us to evaluate the probability of alien presence in specific cells based on detected alien beeps. Integrating this probabilistic approach into our detection system enhances its precision and effectiveness in identifying potential alien activity during detection operations

STEP 3: CREW MEMBER DETECTION

CrewBelief[] is the probability matrix for Crew members.

(a) Crew Beep is heard

Calculating probability of Crew member in each cell given Alien being in the Detection Square

$$P(Crew\ in\ cell\ i \mid Crew\ Beep\ in\ cell\ j\ )$$

Conditional Probability:

$$P(Crew\ in\ cell\ i \mid Crew\ Beep\ in\ cell\ j\ ) = \frac{P(Crew\ in\ cell\ i\ \wedge\ Crew\ Beep\ in\ cell\ j)}{P(Crew\ Beep\ in\ cell\ j)}$$

By Conditional Factoring:

$$P(Crew\ in\ cell\ i \mid Crew\ Beep\ in\ cell\ j\ ) = \frac{P(Crew\ in\ cell\ i\ ) * P(Crew\ Beep\ in\ cell\ j \mid Crew\ in\ cell\ i\ )}{P(Crew\ Beep\ in\ cell\ j)}$$

We can compute, $P(Crew\ Beep\ in\ cell\ j)$

$$= \sum_k P(beep\ in\ cell\ j\ and\ crew\ in\ cell\ k)$$

$$= \sum_k P(crew\ in\ cell\ k) * P(beep\ in\ cell\ j \mid crew\ in\ cell\ k\ )$$

$$P(Crew\ in\ cell\ i \mid Crew\ Beep\ in\ cell\ j\ )$$

$$= \frac{P(Crew\ in\ cell\ i\ ) * P(Crew\ Beep\ in\ cell\ j \mid\ Crew\ in\ cell\ i\ )}{\sum_k P(Crew\ in\ cell\ k) * P(Crew\ Beep\ in\ cell\ j \mid Crew\ in\ cell\ k\ )}$$

$$\boxed{P(Crew\ in\ cell\ i \mid Crew\ Beep\ in\ cell\ j\ ) = \frac{CrewBelief[j] * e^{-\alpha(dist(i,j)-1)}}{\sum_k CrewBelief[k] * e^{-\alpha(dist(k,j)-1)}}}$$

(b) <u>Crew Beep is not heard</u>

Calculating probability of Crew member in each cell given Alien being in the Detection Square

$$P(Crew\ in\ cell\ i \mid No\ Crew\ Beep\ in\ cell\ j\ )$$

Conditional Probability:

$$P(Crew\ in\ cell\ i \mid No\ Crew\ Beep\ in\ cell\ j\ )$$

$$= \frac{P(Crew\ in\ cell\ i\ \ \wedge\ \ No\ Crew\ Beep\ in\ cell\ j)}{P(No\ Crew\ Beep\ in\ cell\ j)}$$

By Conditional Factoring:

$$P(Crew\ in\ cell\ i \mid No\ Crew\ Beep\ in\ cell\ j\ )$$

$$= \frac{P(Crew\ in\ cell\ i\ ) * P(No\ Crew\ Beep\ in\ cell\ j \mid Crew\ in\ cell\ i\ )}{P(No\ Crew\ Beep\ in\ cell\ j)}$$

We can compute, $P(No\ Crew\ Beep\ in\ cell\ j)$

$$= \sum_k P(No\ Crew\ Beep\ in\ cell\ j\ and\ Crew\ in\ cell\ k)$$

$$= \sum_k P(Crew\ in\ cell\ k) * P(No\ Crew\ Beep\ in\ cell\ j \mid Crew\ in\ cell\ k\ )$$

$$P(Crew\ in\ cell\ i \mid No\ Crew\ Beep\ in\ cell\ j\ )$$

$$= \frac{P(Crew\ in\ cell\ i\ )*P(No\ Crew\ Beep\ in\ cell\ j \mid \ Crew\ in\ cell\ i\ )}{\sum_k P(Crew\ in\ cell\ k)*P(No\ Crew\ Beep\ in\ cell\ j \mid Crew\ in\ cell\ k\ )}$$

$$\boxed{P(Crew\ in\ cell\ i \mid No\ Crew\ Beep\ in\ cell\ j\ ) = \frac{CrewBelief[j]*e^{-\alpha(dist(i,j)-1)}}{\sum_k CrewBelief[k]*e^{-\alpha(dist(k,j)-1)}}}$$

STEP 4: NORMALIZATION AFTER BOT MOVEMENT

When the bot moves to a new cell, if it doesn't find a crew member or doesn't get attacked by an alien in the new cell, we can make the probability of that cell having crew or alien as 0. In order to distribute this probability into other cells, we do normalization such that the sum of all probabilities after the bot moves sum to 1. For $i \neq j$,

$$P(Alien\ is\ in\ cell\ i \mid Alien\ is\ not\ in\ cell\ j)$$

$$= \frac{P(Alien\ is\ in\ cell\ i\ \wedge\ ALien\ is\ not\ in\ cell\ j)}{P(Alien\ is\ not\ in\ cell\ c)}$$

$$= \frac{P(Alien\ is\ in\ cell\ i).P(Alien\ is\ not\ in\ cell\ i \mid Alien\ is\ in\ cell\ j)}{1 - P(Alien\ is\ in\ cell\ j)}$$

$$= \frac{P(Alien\ is\ in\ cell\ i)}{1 - P(Alien\ is\ in\ cell\ j)}$$

STEP 5: UPDATE BELIEF AFTER ALIEN MOVEMENT

After the alien makes a move, the bot has to account for the alien movement and update the Alien Belief.

Let N(i) be the neighbours of the cell i,

$$\boxed{P(alien\ in\ cell\ i\ after\ movement) = \sum_{cell\ x\ \in N(i)} \frac{P(alien\ in\ cell\ x)}{|Number\ of\ open\ neighbors\ of\ cell\ x|}}$$

BOT 1:

In the implementation outlined in the above section, our focus lies in continuously updating the Belief Network of both the Alien and crew as we navigate through the open cells of the ship. Bot 1 adopts an approach emphasizing dynamic adjustments in knowledge matrices based on available information collected from the sensors. Upon potential alien movement, the bot promptly revises its understanding of alien locations.

Following this, Bot 1 employs a movement strategy geared towards approaching cells with the highest probability of containing crew members. Throughout its movement, the bot actively steers clear of cells confidently identified as alien habitats. This adaptable and probabilistic methodology empowers Bot 1 to move through the game environment, making decisions informed by evolving information. The aim is to optimize the likelihood of successfully reaching and rescuing the crew member while minimizing encounters with aliens.

This strategy embodies a proactive and cautious decision-making approach, leveraging real-time data to guide the bot towards accomplishing its rescue mission effectively.

The A* algorithm is utilized to determine the shortest path from the bot's current position to the cell with the highest crew probability. The heuristic employed for this purpose is the Manhattan distance from the bot's position to the target cell.

SCOPE OF IMPROVEMENT FOR BOT 1:

Bot 1 prioritizes of finding the shortest path to the cell containing highest crew probability by avoiding the aliens in neighbouring cells. However, this strategy may result in higher chance of getting attacked by the alien at a later point of time. To enhance efficiency, introducing a weight parameter could enable the Bot to choose shorter paths to reach crew members and also avoid alien probability.

BOT 2:

Bot 2 has to save a single crew member and stay away from a single alien, and the belief network of this Bot is updated the same way it was done for Bot 1. This Bot maximizes the chance of finding the shortest path to the crew member while minimizing the chance of encountering the alien during the traversal based on the information received from the sensors and updating the Belief network accordingly which was a drawback in Bot 1. This strategy is implemented by utilizing A* algorithm.

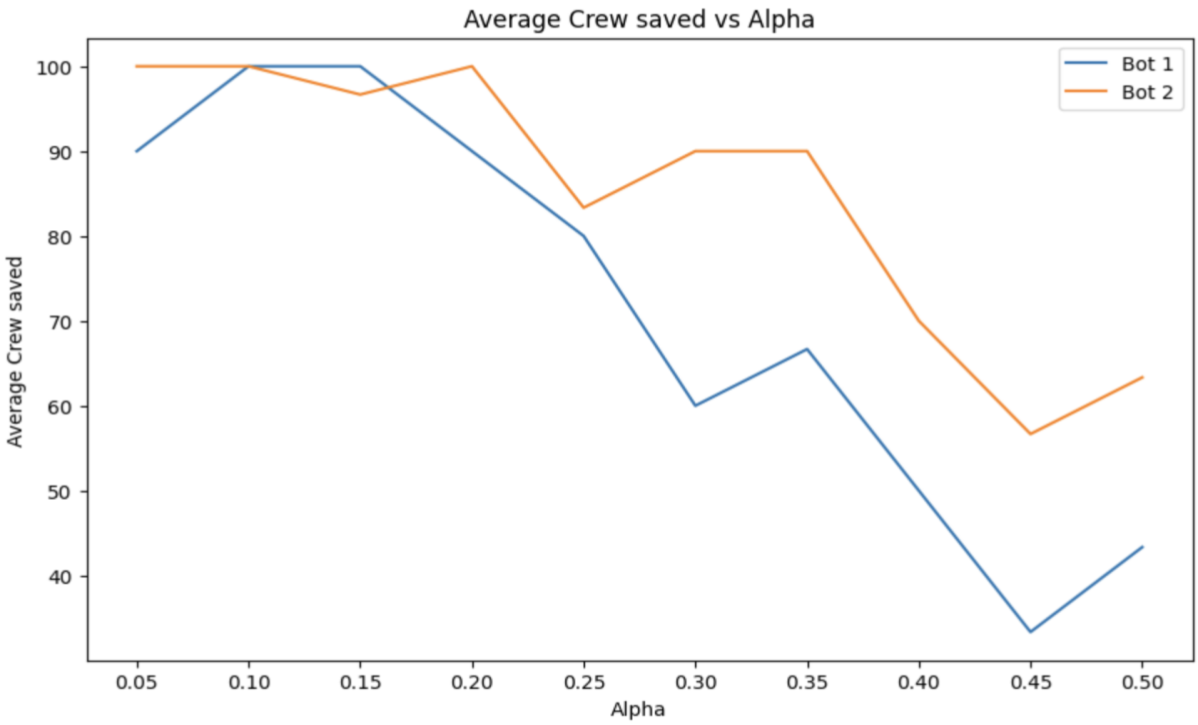
Furthermore, the heuristic value for the A* algorithm includes the Manhattan distance, the density of alien probability in the surrounding area for each neighbouring cell, and the crew probability associated with neighbouring cells.

The density of alien probability is considered to prevent the Bot in encountering the alien while taking into account the future steps. The bot also considers choosing the next cell to move with a higher crew probability.

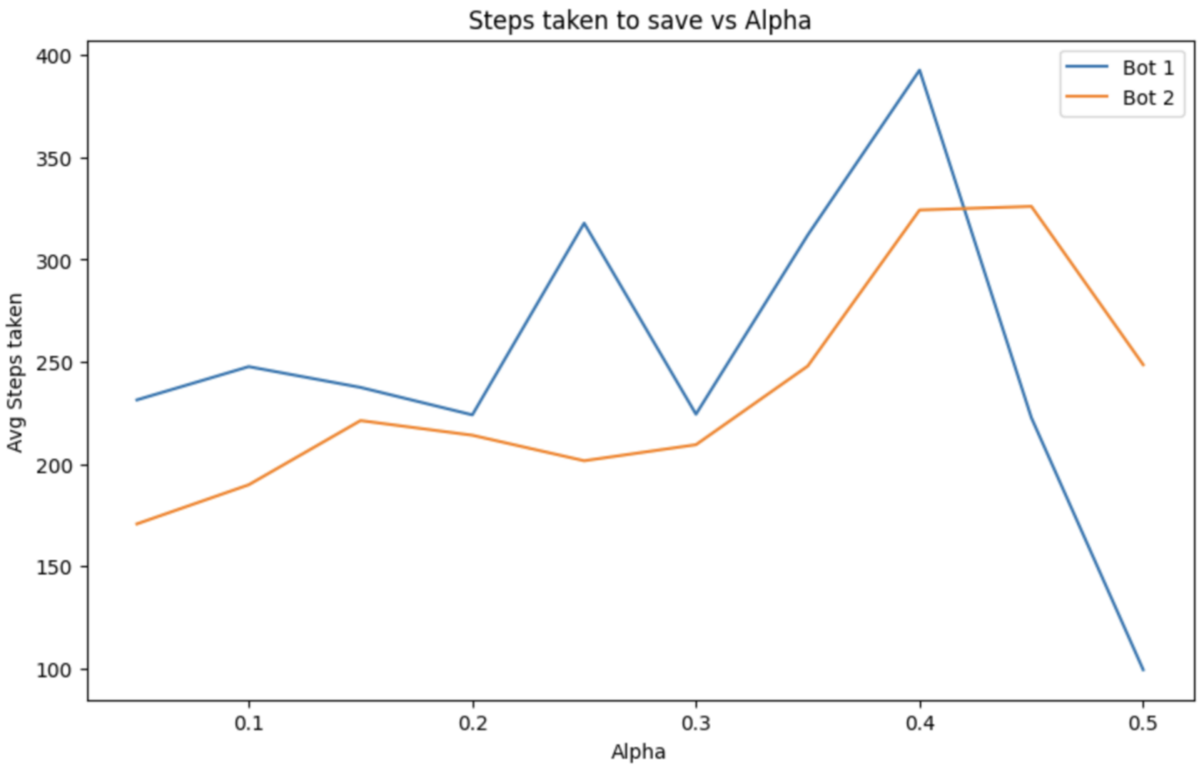
PERFORMANCE MEASURE OF BOT 1 AND BOT 2:

The performance measure of Bot 1 and Bot 2 have been calculated by running it for various values of α and k and the following observations was inferred.

For k = 8

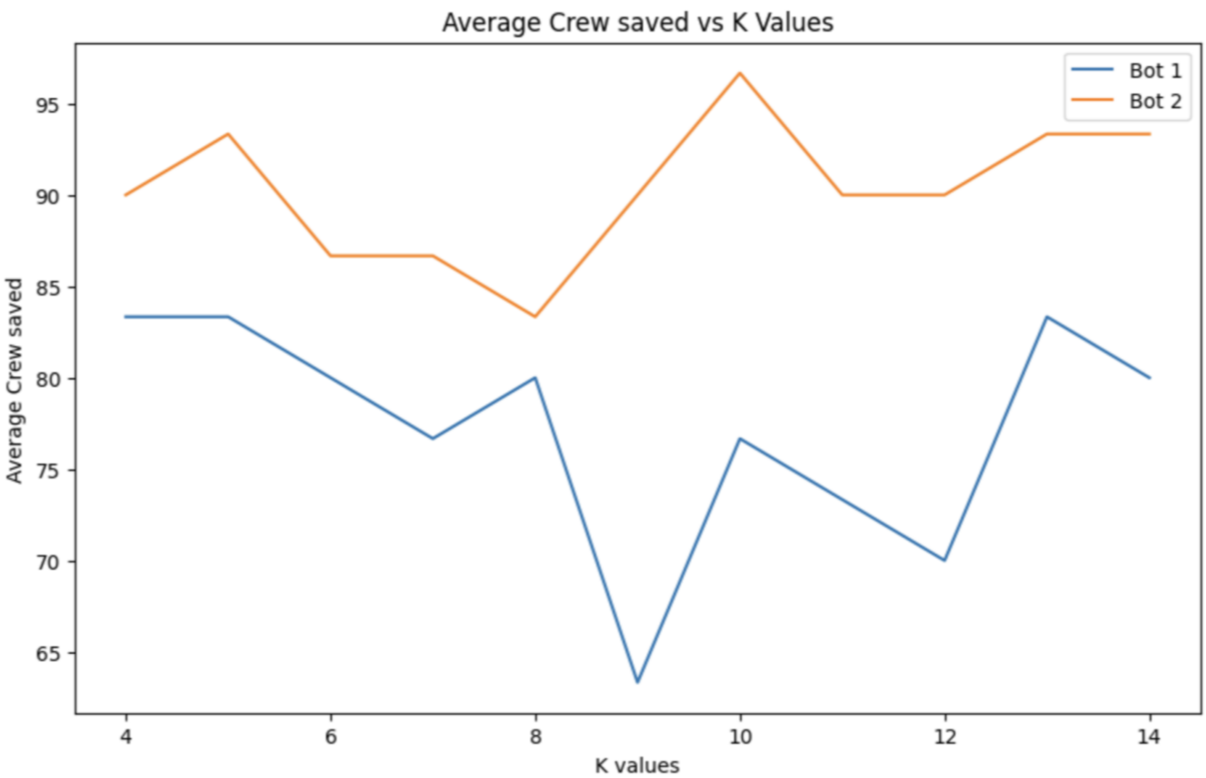


Graph 1 compares Bot 1 and Bot 2 based on the total crew saved across different alpha values for a constant k value throughout 20 iterations. Bot 2 shows superior performance, especially noticeable within the alpha range of [0.20,0.50], where it significantly increases the average crew members saved. Conversely, Bot 1 tends to detect more crew signals in the alpha range [0.10,0.15], even when crew members are far from the bot, while Bot 2 detects crew signals at higher alpha values (>0.45) even if they are close or nearby to the Bot.

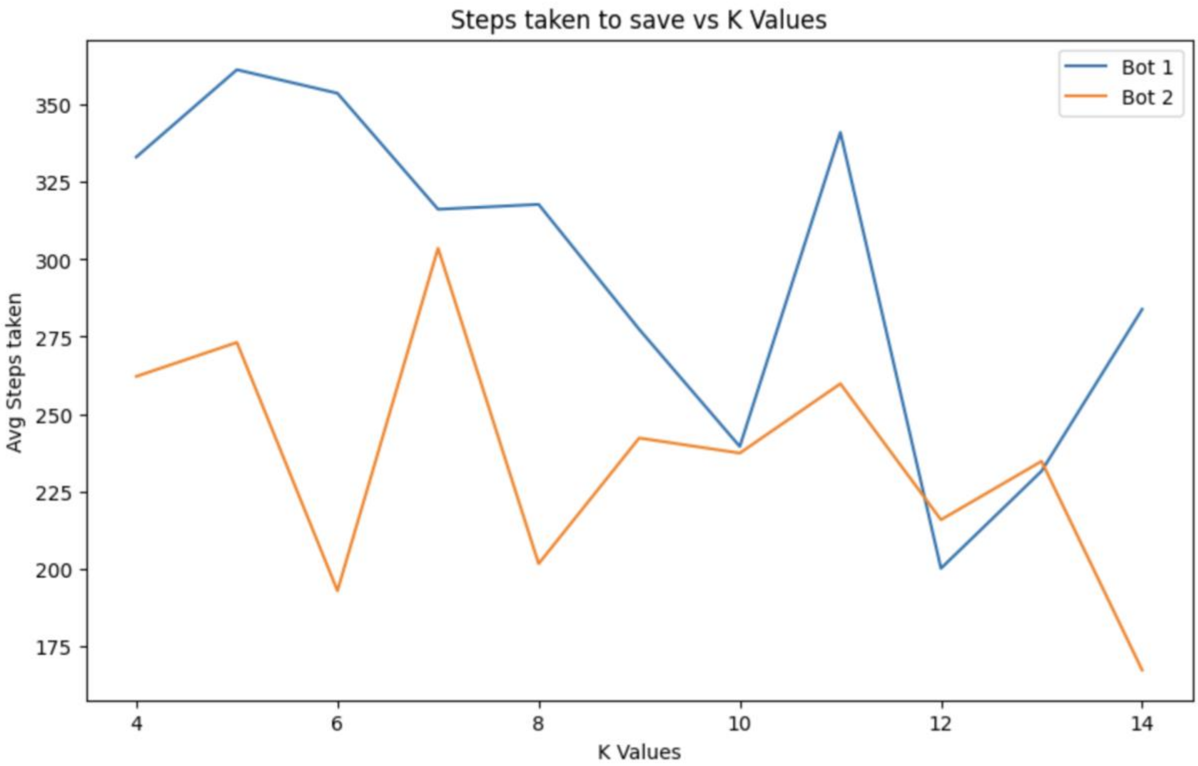


Graph 2 displays the comparison between Bot 1 and Bot 2 regarding the average number of steps taken to save the crew across various alpha values and for a constant k value. The optimization of the weights added to the A* algorithm of Bot 2 , leads to an decrease in the average number of steps required to save the crew.

For α = 0.25



Graph 3 illustrates the comparison between Bot 1 and Bot 2 concerning the average crew saved across different k values, focusing on the optimal alpha value of 0.25. Bot 2 outperforms Bot 1 for k values ranging from 8 to 14. However, k values lower than 8 reduce bot awareness of alien probabilities, resulting in more bot deaths.



Graph 4 displays the comparison regarding the average number of steps taken to save the crew across various k values and for a constant optimal α value for Bot 1 and Bot 2. The optimization of the weights added to the A* algorithm of Bot 2, significantly reduces the average number of steps taken by the Bot in rescuing the crew member.

In summary, Bot 2's updated heuristic demonstrates superior performance over Bot 1 across various α and k values.

SCENARIO 2 – ONE ALIEN, TWO CREW

In this scenario the calculations related to alien remain the same as the previous scenario and since this scenario has two crew members, the probability calculations for 2 crew members is given below:

(a) Crew Beep is heard

$$P(Crew\ 1\ \ in\ cell\ j, Crew\ 2\ in\ cell\ k \mid Crew\ beep\ in\ cell\ i)$$

$$= \frac{P(Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k\ \ \wedge\ \ Crew\ Beep\ in\ cell\ i)}{P(Crew\ Beep\ in\ cell\ k)}$$

$$= \frac{P(Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k\ ) * P(Crew\ Beep\ in\ cell\ i\ |\ Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k)}{P(Crew\ Beep\ in\ cell\ i)}$$

We can compute $P(Crew\ Beep\ in\ cell\ i)$ as follows,

$$P(Crew\ Beep\ in\ cell\ i)$$

$$= \quad \textstyle\sum_j \sum_k P(Crew\ in\ cell\ j, Crew\ in\ cell\ k\ \ \wedge\ Beep\ in\ cell\ i)$$

$$= \sum_j \sum_k (Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k) * P(Crew\ Beep\ in\ cell\ i\ |\ Crew\ 1\ in\ cell\ j, Crew\ 2\ in\ cell\ k)$$

$$P(crew\ 1\ \ in\ cell\ i, crew\ 2\ in\ cell\ j\ |\ beep\ in\ cell\ k)$$

$$= \frac{P(crew\ 1\ in\ cell\ i\ , crew\ 2\ in\ cell\ j\ ) * P(crew\ beep\ in\ cell\ k\ |\ crew\ 1\ in\ cell\ i\ , crew\ 2\ in\ cell\ j)}{\sum_i \sum_j (crew\ 1\ in\ cell\ i\ , crew\ 2\ in\ cell\ j\ ) * P(crew\ beep\ in\ cell\ k\ |\ crew\ 1\ in\ cell\ i\ , crew\ 2\ in\ cell\ j)}$$

Since crew 1 in cell j and crew 2 in cell k are independent, we can write as follows,

$$P(Crew\ Beep\ in\ cell\ k\ |\ Crew\ 1\ in\ cell\ i\ , Crew\ 2\ in\ cell\ j)$$

$$= \quad P(Beep\ in\ cell\ k\ |\ Crew\ 1\ in\ cell\ i\ )\ \vee\ P(Beep\ in\ cell\ k\ |\ Crew\ 2\ in\ cell\ j)$$

$$= \quad P(Beep\ in\ cell\ k\ |\ Crew\ 1\ in\ cell\ i\ ) + \ \ P(Beep\ in\ cell\ k\ |\ Crew\ 2\ in\ cell\ j) - P(Beep\ in\ cell\ k\ |\ Crew\ 1\ in\ cell\ i\ ) * P(Beep\ in\ cell\ k\ |\ Crew\ 2\ in\ cell\ j)$$

$$= \quad e^{-\alpha(dist(k,i)-1)}\ +\ e^{-\alpha(dist(k,j)-1)} - (e^{-\alpha(dist(k,i)-1)} *\ e^{-\alpha(dist(k,j)-1)})$$

(b) Crew Beep is not heard

Similarly, as above,

$$P(crew\ 1\ \ in\ cell\ i, crew\ 2\ in\ cell\ j\ |\ no\ beep\ in\ cell\ k)$$

$$= \frac{P(Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k\ \ \wedge\ \ \ No\ beep\ in\ cell\ i)}{P(No\ beep\ in\ cell\ i)}$$

$$= \frac{P(Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k\ ) * P(No\ beep\ in\ cell\ i \mid Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k)}{P(No\ beep\ in\ cell\ i)}$$

$$P(crew\ 1\ \ in\ cell\ i, crew\ 2\ in\ cell\ j \mid no\ beep\ in\ cell)$$
$$= \frac{P(Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k\ ) * [1 - \ P(Beep\ in\ cell\ i \mid Crew\ 1\ in\ cell\ j\ , Crew\ 2\ in\ cell\ k)]}{1 - \ P(Beep\ in\ cell\ i)}$$

$$P(beep\ in\ cell\ k)$$
$$=$$
$$\sum_i \sum_j (crew1\ in\ cell\ i\ , crew2\ in\ cell\ j\ ). P(beep\ in\ cell\ k \mid crew1\ in\ cell\ i\ , crew2\ in\ cell\ j)$$

And,

$$P(beep\ in\ cell\ k \mid crew\ 1\ in\ cell\ i\ , crew\ 2\ in\ cell\ j)$$
$$= \ e^{-\alpha(dist(k,i)-1)} + e^{-\alpha(dist(k,j)-1)} - (e^{-\alpha(dist(k,i)-1)} * e^{-\alpha(dist(k,j)-1)})$$

BOT 3:

BOT 3, an extension of BOT 1, except for the fact that it was under the assumption of only one crew member to be rescued, despite there being two crew members present. Similar to the previous section's calculations, we continuously update the Belief Network for both the Alien and crew positions as we navigate the ship.
Bot 3 looks for one crew member irrespective of the presence of two crews. It primarily searches for 1 crew. After reaching the first crew member, the bot will be hearing the crew beeps from the second crew member and continues towards it. It is unaware of the fact that there are two crew members on the ship.

The core strategy of Bot 3 revolves around dynamically updating its knowledge regarding the positions of aliens and crew members based on real-time data. When the potential movement of an alien occurs, Bot 3 promptly adjusts its knowledge matrix to reflect the updated alien locations. Subsequently, Bot 3 employs a movement strategy focused on approaching the cell with the highest probability of containing a crew member. The strategy embodies a proactive and cautious decision-making process, utilizing evolving information to guide Bot 3 towards a successful rescue mission.
The A* algorithm is utilized to determine the shortest path from the bot's current position to the cell with the highest crew probability. The heuristic employed for this purpose is the Manhattan distance from the bot's position to the target cell.

SCOPE OF IMPROVEMENT FOR BOT 3:

Bot 3 initially assumes there is only one crew member that has to be rescued and after rescuing one crew member, Bot 3 realises there is another crew member. This Bot can be improved by updating the Belief Network on the presence of two crew members that have to be rescued.

BOT 4:

Bot 4 is an extension of Bot1 but with the extension of rescuing two crew members instead of one. The Bot performs the calculations for rescuing both the crew members based on the information it perceives from the crew member detection sensor. As previously described, we continuously update the Belief Network for both the Alien and crew positions throughout our traversal of the ship.
Here Bot 4 considers the joint probability of the existence of 2 crew members on the ship. Subsequently, Bot 4 implements a movement strategy tailored to approach the cell deemed most probable for containing one of the crew members. Throughout this process, the bot actively avoids cells confidently identified as alien habitats. This adaptive and probabilistic approach enables Bot 4 to navigate the environment effectively.
Utilizing the A* algorithm, Bot 4 determines the shortest path from its current position to a cell with the highest crew probability. The heuristic employed for this purpose is the Manhattan distance from the bot's position to the target cell.

SCOPE OF IMPROVEMENT FOR BOT 4:

Bot 4 prioritizes avoiding cells with a higher probability of containing aliens while also aiming to reach cells with a high likelihood of containing crew members. However, this strategy may result in taking longer paths. To enhance efficiency, introducing a weight parameter could enable the Bot to choose shorter paths to reach crew members.

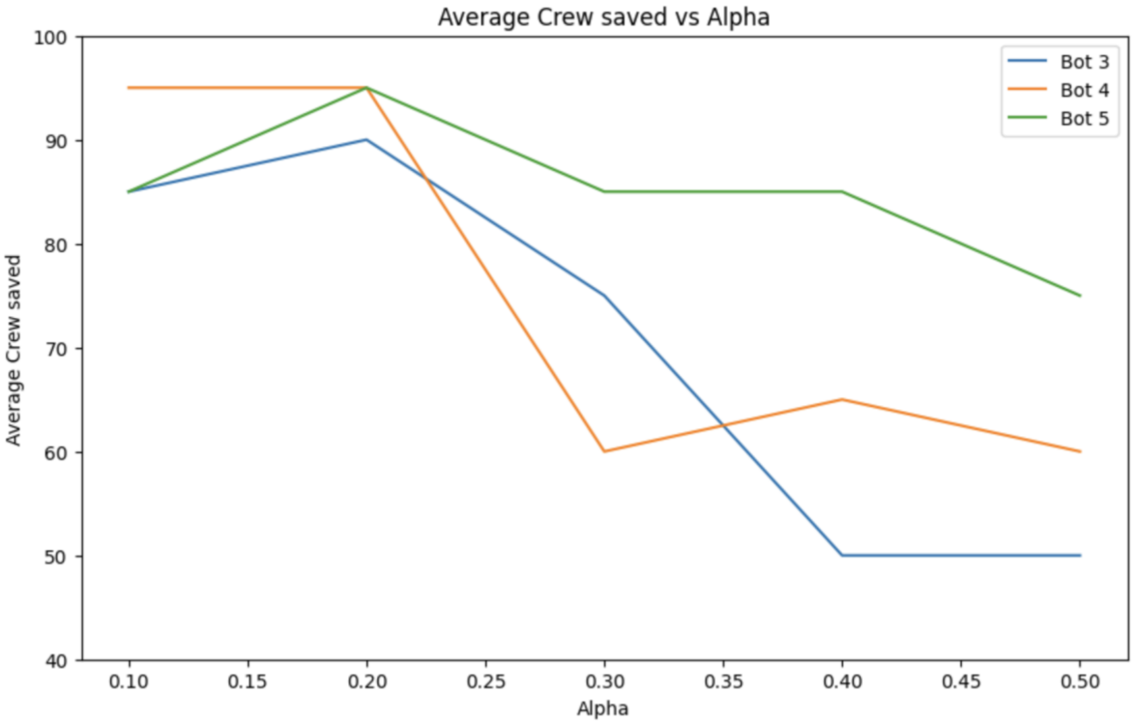
BOT 5:

Bot 5 follows a strategy to save two crew members while avoiding the alien, updating its belief network similarly to Bot 4. This Bot aims to maximize the likelihood of finding the shortest path to the crew member while minimizing the chance of encountering the alien using information from sensors and belief network updates which was a drawback in Bot 4. This strategy is executed using the A* algorithm. Additionally, the heuristic value for the A* algorithm considers Manhattan distance, the density of alien probability in neighbouring cells, and crew member probabilities. The consideration of alien probability density helps the Bot avoid encountering aliens in future steps. In cases where there are multiple potential locations for the crew member, the Bot selects the one closest to its current position.

PERFORMANCE MEASURE OF BOT 3, BOT 4 AND BOT 5:

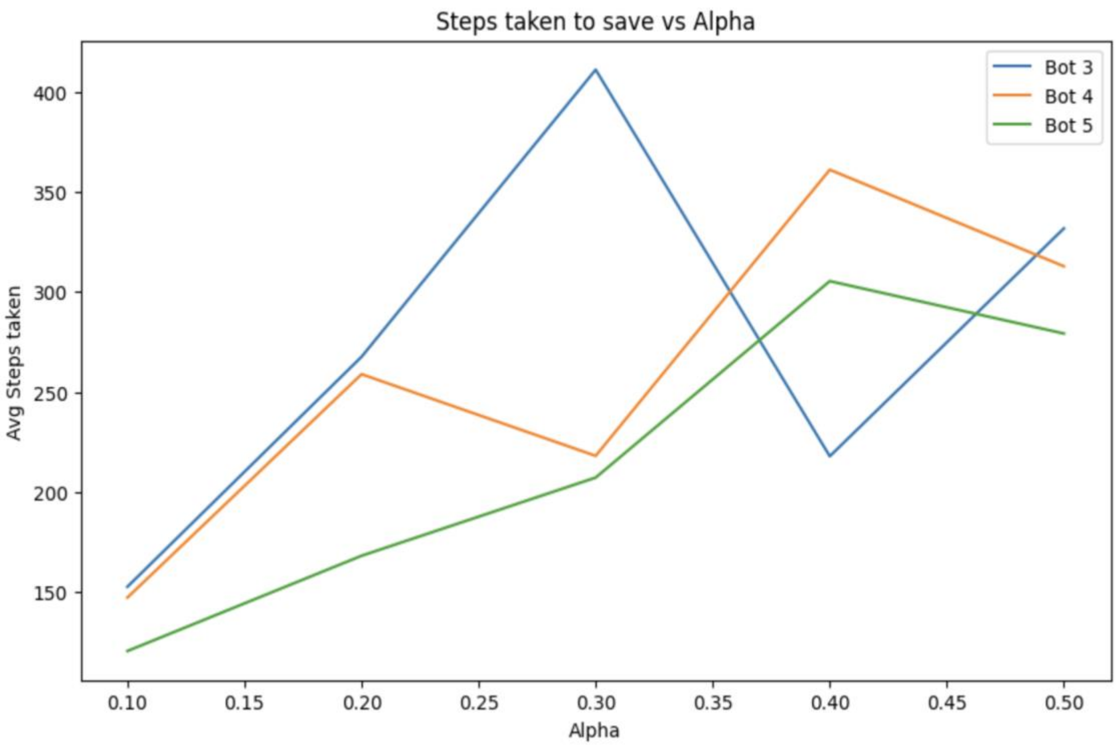
The performance measure of Bot 1 and Bot 2 have been calculated by running it for various values of α and k and the following observations was inferred.

For k = 9



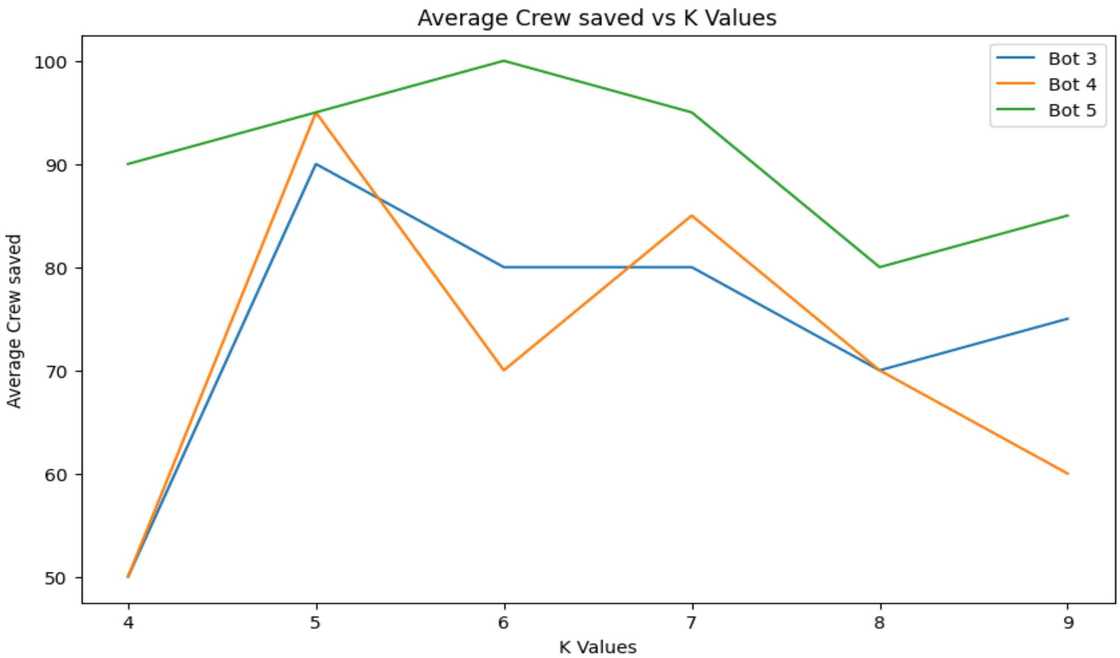
Graph 5

Graph 5 compares Bot 3, Bot 4 and Bot 5 based on the average percentage of the number of crew saved across different α values for a constant k = 9 for 20 iterations. Bot 5 outperforms, especially noticeable within the α range of [0.20,0.50], and at α = 0.35 it significantly increases the average percentage of the number of crew members saved. Conversely, Bot 4 tends to detect more crew signals for α less than 0.20 even when crew members are far from the bot, while Bot 3 tends to be performing better than Bot 4 within the α range of [0.25,0.35].



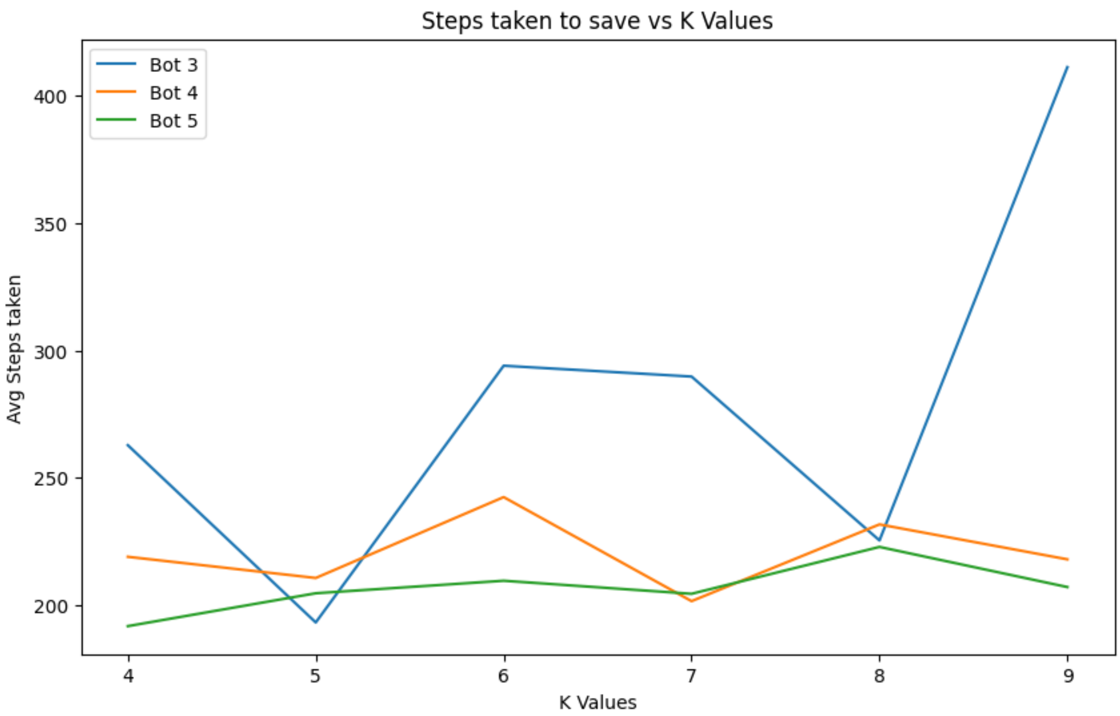
Graph 6

Graph 6 displays the comparison between Bot 3, Bot 4 and Bot 5 regarding the average number of steps taken by Bot 3, Bot 4 and Bot 5 to save the crew members across various α values and for a constant k =9. The optimization of the weights added to the A* algorithm of Bot 5 within the α range [0.10,0.35], leads to a decrease in the average percentage of the number of steps required to save the crew. Whereas, Bot 4 takes reduced average steps compared to Bot 3 in the α range of [0.20,0.35]. Bot 3 outperforms for α range [0.37,0.45].



Graph 7

Graph 7 illustrates the comparison between Bot 3, Bot 4 and Bot 5 concerning the average percentage of the number of crew members saved across different k values, focusing on the optimal alpha value of 0.3. Bot 5 outperforms Bot 3 and Bot 4 for k values in range of [5,9].



Graph 8

Graph 8 depicts the comparison regarding the average number of steps taken by Bot 3, Bot 4 and Bot 5 to save the crew members across various k values and for a constant optimal α value. The optimization of the weights added to the A* algorithm of Bot 2, significantly reduces the number of steps taken by the Bot in rescuing the crew member.

In summary, Bot 5's updated heuristic enables this bot to outperform Bot 3 and Bot 4 across various α and k values.

SCENARIO 3 – TWO ALIEN, TWO CREW

(a) Alien not in Detection Square or no Alien Beep

D(i) = {open cells in the detection square centred at i}

Initial Calculation – Conditional Probability:

$$\boldsymbol{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i))}$$

$$= \frac{P\big(Alien\ in\ cell\ j, Alien\ in\ cell\ k\ \wedge\ No\ beep\ in\ D(i)\big)}{P(No\ beep\ in\ D(i))}$$

By Conditional Factoring:

$$\boldsymbol{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i))}$$

$$= \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(\ No\ beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{P(No\ beep\ in\ D(i))}$$

Now, let us calculate $P(No\ beep\ in\ D(i))$

$$P\big(No\ beep\ in\ D(i)\big) = \sum_p \sum_q P\big(Alien\ in\ cell\ p, Alien\ in\ cell\ q\ \wedge\ No\ beep\ in\ D(i)\big)$$

$$= \sum_p \sum_q P(Alien\ in\ cell\ p, Alien\ in\ cell\ q) * P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q)$$

$$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i))$$
$$= \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(\ No\ beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_p \sum_q P(Alien\ in\ cell\ p, Alien\ in\ cell\ q) * P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q)}$$

If cells p, q are in D(i) then, $P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q) = 0$

If cells p, q are not in D(i) then, $P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q) = 1$

$$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i))$$
$$= \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(\ No\ beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_{p \notin DS(t)} \sum_{q \notin DS(t)} P(alien\ in\ cell\ p, alien\ in\ cell\ q)}$$

If cells j, k are in D(i) then, $P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k) = 0$

$$\boxed{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i)) = 0}$$

If cells j, k are not in D(i) then, $P(No\ beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k) = 1$

$$\boxed{(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ beep\ in\ D(i)) = \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_{p \notin DS(t)} \sum_{q \notin DS(t)} P(alien\ in\ cell\ p, alien\ in\ cell\ q)}}$$

Alien in detection square or alien beep case:

D(i) = {open cells in the detection square centered at i}

Initial Calculation – Conditional Probability:

$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid Beep\ in\ D(i))$

$$= \frac{P\big(Alien\ in\ cell\ j, Alien\ in\ cell\ k\ \wedge\ Alien\ beep\ in\ D(i)\big)}{P(Beep\ in\ D(i))}$$

By Conditional Factoring:

$$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid Beep\ in\ D(i)) = \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(Beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{P(Beep\ in\ D(i))}$$

Now, let us calculate $P(No\ beep\ in\ D(i))$

$$P\big(Beep\ in\ D(i)\big) = \sum_p \sum_q P\big(Alien\ in\ cell\ p, Alien\ in\ cell\ q\ \wedge Beep\ in\ D(i)\big)$$

$$= \sum_p \sum_q P(Alien\ in\ cell\ p, Alien\ in\ cell\ q) * P(alien\ beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q)$$

$$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid Beep\ in\ D(i)) = \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(Beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_p \sum_q P(Alien\ in\ cell\ p, Alien\ in\ cell\ q) * P(Beep\ in\ D(i) | Alien\ in\ cell\ p, Alien\ in\ cell\ q)}$$

If cells p, q are in D(i) then, $P(Beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q) = 1$

If cells p, q are not in D(i) then, $P(Beep\ in\ D(i) \mid Alien\ in\ cell\ p, Alien\ in\ cell\ q) = 0$

$$P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid Beep\ in\ D(i)) = \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k) * P(Beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_{p \in DS(t)} \sum_{q \in DS(t)} P(Alien\ in\ cell\ p, Alien\ in\ cell\ q)}$$

If cells j, k are in D(i) then, $P(Beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k) = 1$

$$\boxed{(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid Beep\ in\ D(i)) = \frac{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k)}{\sum_{p \in DS(t)} \sum_{q \in DS(t)} P(Alien\ in\ cell\ p, Alien\ in\ cell\ q)}}$$

If cells j, k are not in D(i) then, $P(No\ Beep\ in\ D(i) \mid Alien\ in\ cell\ j, Alien\ in\ cell\ k) = 0$

$$\boxed{P(Alien\ in\ cell\ j, Alien\ in\ cell\ k \mid No\ Beep\ in\ D(i)) = 0}$$

BOT 6:

BOT 6, an extension of BOT 3, as it is designed to operate in a scenario with two crew members and two aliens. As detailed in the preceding section, our approach involves continually updating the Belief Network to track the positions of both aliens and crew members as we traverse through the ship.

The core strategy of Bot 6 revolves around dynamically updating its knowledge of alien and crew member positions based on information perceived from both the alien detection and crew member detection sensors. In response to potential alien movements, the bot promptly adjusts its knowledge matrix to reflect the updated locations of the aliens.

Following this, Bot 6 implements a movement strategy focused on approaching the cell with the highest probability of containing one of the crew members. Throughout this movement, the bot actively avoids cells confidently identified as alien habitats. This adaptive and probabilistic approach enables Bot 6 to navigate the game environment effectively.

Utilizing the A* algorithm, Bot 6 determines the shortest path from its current position to a cell with the highest crew probability. The heuristic employed in this computation combines the Manhattan distance from the bot's position to the target. This strategic framework embodies a proactive and cautious decision-making process, utilizing evolving information to guide Bot 6 towards a successful rescue mission.

SCOPE OF IMPROVEMENT FOR BOT 6:

Bot 6 assumes initially that the sensors beep for only crew member or beeps for only alien whereas in this scenario, it has two crew members and two aliens. This Bot can be improved by updating the Belief Network with the presence of two aliens and two crew members.

BOT 7:

BOT 7, extending the capabilities of BOT 6, operates on the assumption of two crew members and integrates policies for updating both alien and crew belief networks. As outlined in our mathematical calculations, we continually adjust the Belief Network for both aliens and crew members as we navigate through the ship.

Subsequently, BOT 7 adopts a movement strategy aimed at approaching the cell with the highest probability of containing one of the crew members. Throughout this process, the bot actively steers clear of cells confidently identified as alien habitats. This adaptive and probabilistic approach empowers BOT 7 to navigate effectively.

Crew and alien probabilities are stored separately in four-dimensional matrices, with crew probabilities represented as pairs of cells containing the crew members. To determine the shortest path from the BOT's current position to a cell with the highest crew probability, we employ the A* algorithm. The heuristic utilizes the Manhattan distance from the BOT's position to the target cell.

SCOPE OF IMPROVEMENT FOR BOT 7:

Like Bot 1, Bot 7 is primarily concerned with avoiding cells with heightened alien probability while also prioritizing cells with a high likelihood of containing crew members. However, this strategy may result in choosing longer routes. To enhance efficiency, incorporating a weight parameter could enable the Bot to favour shorter paths when reaching crew members.

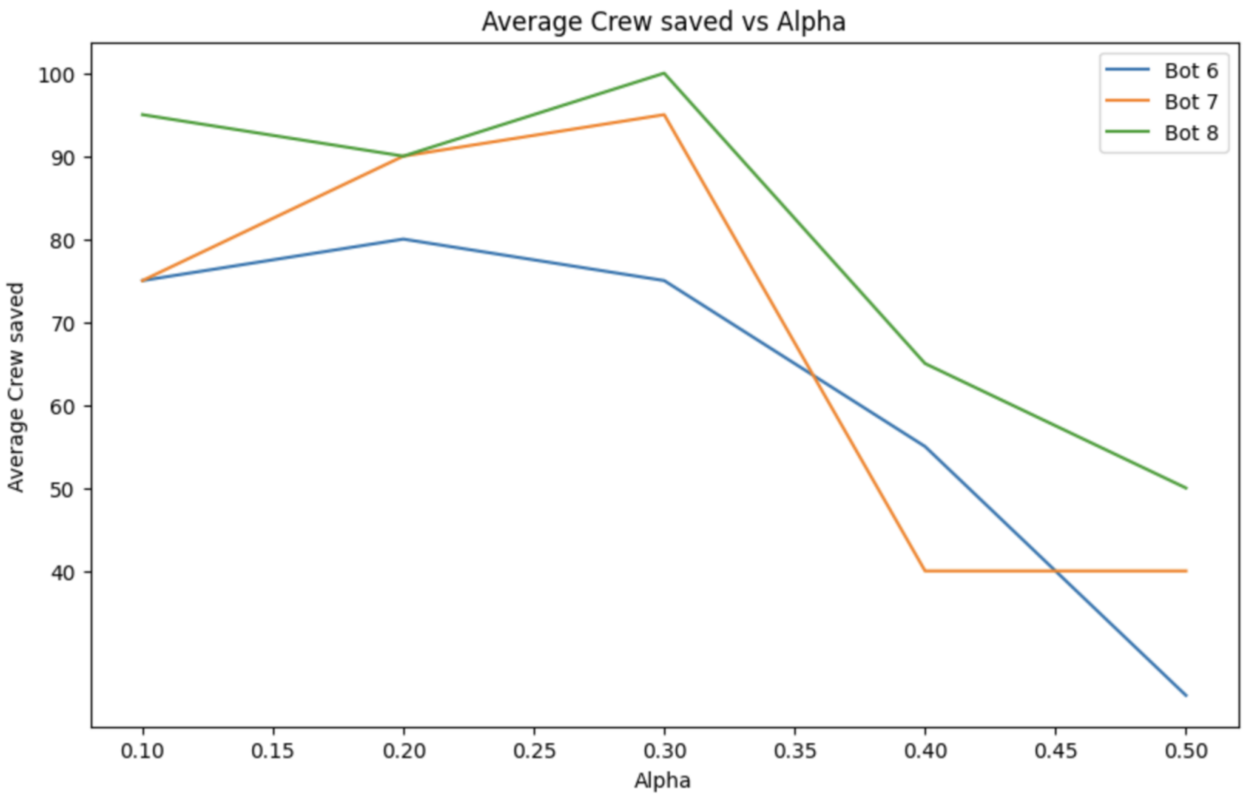
BOT 8:

Bot 8 is guided by a strategy aimed at rescuing two crew members while evading two aliens, with its belief network updates mirroring that of Bot 7. Unlike Bot 7, this Bot addresses the previous drawback by maximizing the probability of finding the shortest path to the crew members and minimizing the risk of alien encounters through sensor data and belief network adjustments. This approach is implemented using the A* algorithm.

Furthermore, the heuristic value for A* incorporates factors such as Manhattan distance, the density of alien probability in neighbouring cells, and crew member probabilities. By considering alien probability density, the Bot can navigate future steps while avoiding alien encounters. When faced with multiple potential crew member locations, the Bot prioritizes the one closest to its current position.

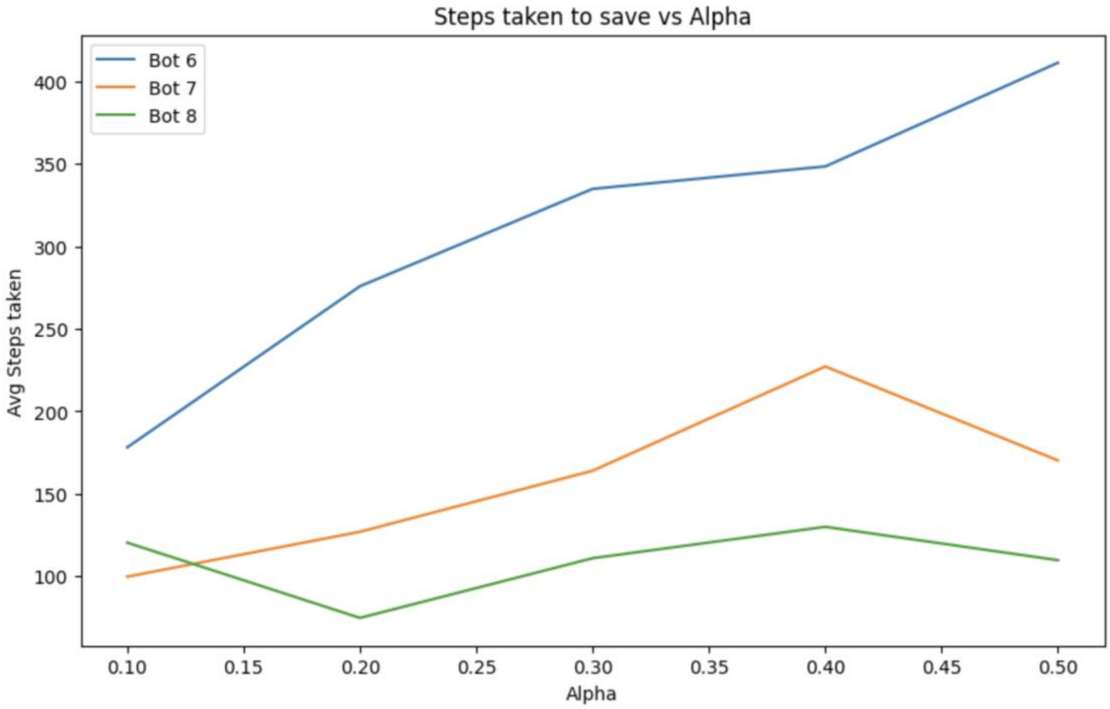
PERFORMANCE MEASURE OF BOT 6, BOT 7 AND BOT 8:

k = 7



Graph 9

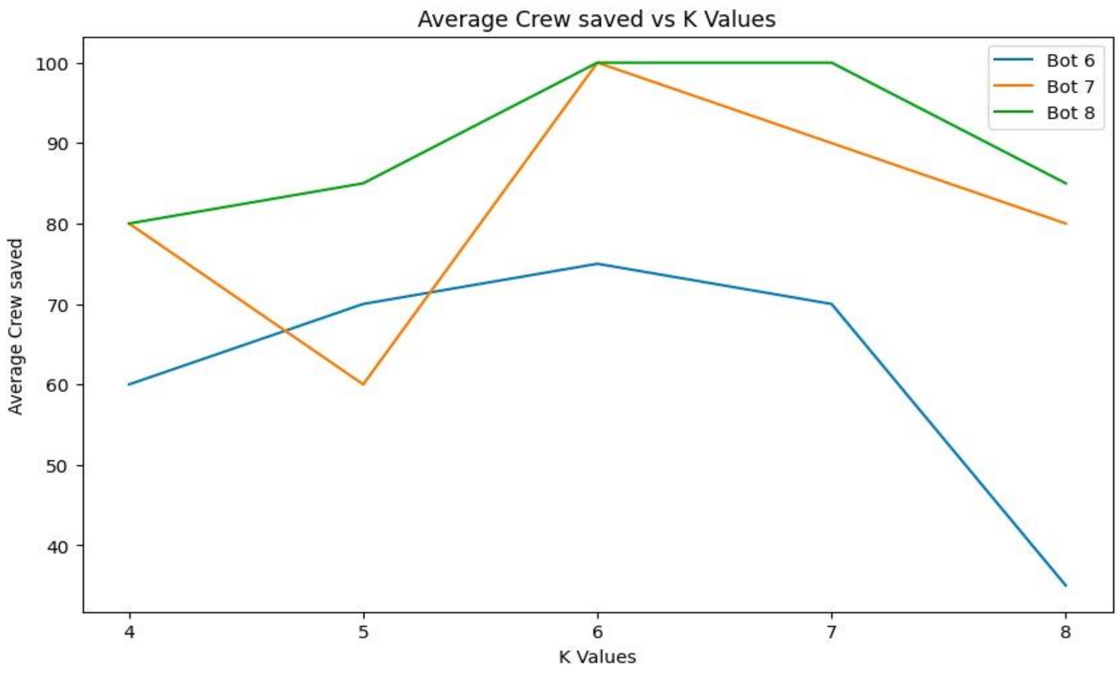
Graph 9 compares Bot 6, Bot 7 and Bot 8 based on the average percentage of the number of crew members saved across different α values for a constant k value throughout 20 iterations. Bot 8 shows better performance, especially noticeable within the alpha range of [0.30,0.50], where it significantly increases the average percentage of the number of crew members saved. Conversely, Bot 7 tends to detect more crew signals in the α range [0.10,0.35], even when crew members are far from the bot, while Bot 6 performs better than Bot 7 for α range in [0.35,0.45] if they are either close or father away from the Bot.



Graph 10

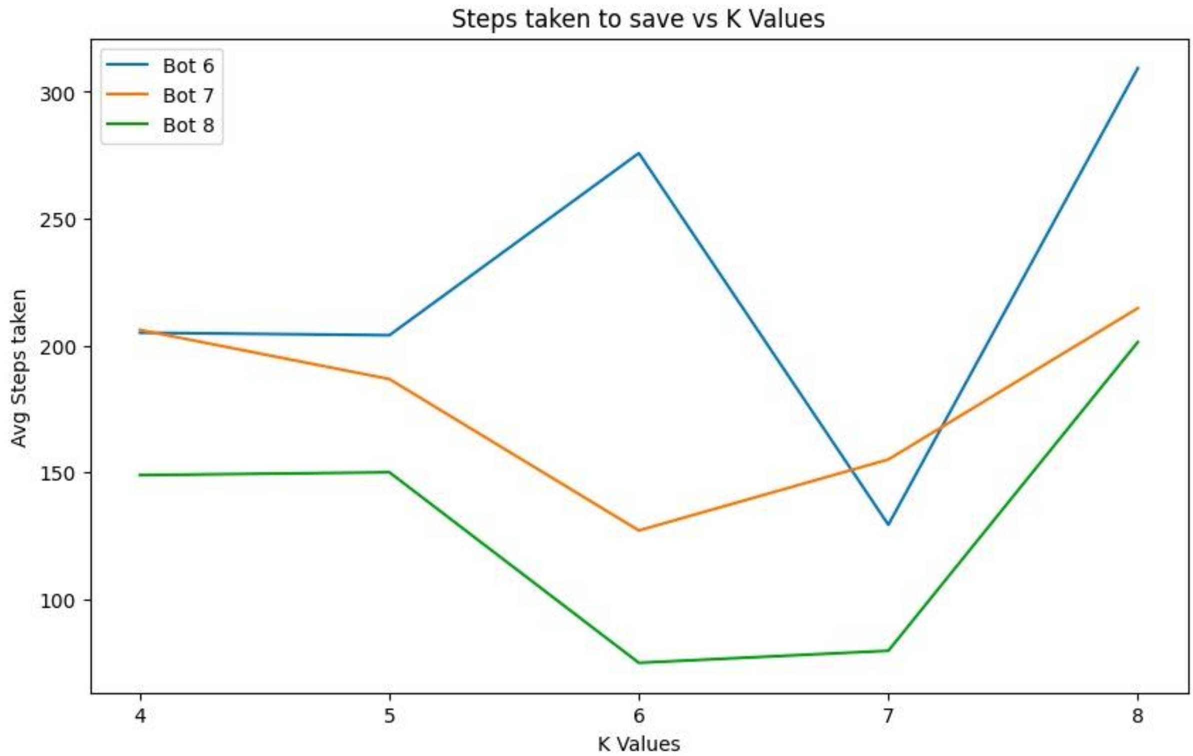
Graph 10 displays the comparison between Bot 6, Bot 7 and Bot 8 regarding the average number of steps taken to save the crew members across various α values and for a constant k = 7. The optimization of the weights added to the A* algorithm of Bot 8 within the α range [0.15,0.50], leads to a decrease in the average percentage of the number of steps required to save the crew. Whereas, Bot 7 takes reduced average steps α values less than 0.15.

α = 0.3



Graph11

Graph 11 illustrates the comparison between Bot 6, Bot 7 and Bot 8 concerning the average percentage of the number of crew members saved across different k values, focusing on the optimal alpha value of 0.3. Bot 8 outperforms Bot 6 and Bot 7 for k values in range of [5.5,8].



Graph 12

Graph 12 depicts the comparison regarding the average number of steps taken by Bot 3, Bot 4 and Bot 5 to save the crew members across various k values and for a constant optimal α value. The optimization of the weights added to the A* algorithm of Bot 8, significantly reduces the number of steps taken by the Bot in rescuing the crew member.

In summary, Bot 8's updated heuristic enables this bot to outperform Bot 6 and Bot 7 across various α and k values.

IDEAL BOT:

The ideal bot for this scenario would be one that effectively combines deterministic and probabilistic reasoning to make informed decisions while considering the safety of itself and rescuing the crew members. An ideal Bot will have the following the characteristics:

- The primary objective of the ideal bot is to rescue crew members efficiently. It should prioritize actions that **maximize the probability of locating and rescuing crew members** while **minimizing the time taken** to do so. The bot's success in this regard can be measured by the average number of crew members rescued per mission and the speed at which they are rescued.

- The ideal bot demonstrates adaptability by **dynamically adjusting its movement decisions** based on real-time sensor data and probabilistic models. It strategically navigates towards areas with higher crew probabilities while actively avoiding regions with elevated alien probabilities. This dynamic decision-making enables the bot to optimize crew rescue opportunities while minimizing the risk of encountering aliens, showcasing a good success rate.
- The bot **continuously learns** from its experiences, refining its decision-making strategies based on past outcomes and environmental dynamics. It adapts its behaviour over time to improve performance, incorporating lessons learned from previous missions to enhance its efficiency and effectiveness in future scenarios. This continuous learning and adaptation ensure that the bot remains resilient and capable of handling evolving challenges in the ship environment.

BONUS QUESTION:

STEP 1: START

Initially, we know that the Bot occupies cell i. There are two matrices one for updating the probabilities of both the Crew member and the alien. We also know that during the start of the stimulation, the crew member and the alien can't be present in the same cell as that of the Bot.

Therefore, $P(Crew\ member\ in\ cell\ j)\ \forall\ j\ \ = 0$

$P(Alien\ member\ in\ cell\ j)\ \forall\ j\ \ = 0$

The above two matrices are zero matrices as their initial probabilities are equal to zero.

STEP 2: ALIEN DETECTION

The alien detection step follows the same math calculations like we did for the previous stimulations.

STEP 3: CREW MEMBER DETECTION

The crew is detected based on the probability detection factor given as $e^{-\alpha(d-1)}$. The Bot receives a Beep if the crew member is at d-distance otherwise no beep is heard.

CASE 1: A BEEP IS HEARD WHEN THE BOT IS AT CELL i

We know that during the start of the stimulation the crew member can't occupy the Bot's location due to which the game isn't over.

P(Crew in cell i) = 0

The remaining open cell in the ship will have the probability, P(Crew in cell j | Beep at cell i) can be derived by using a set of processes.

By Conditional Probability:

P(Crew in cell j | Beep at cell i) = P(Crew in cell j ^ Beep at cell i) / P(Beep at cell i)

= P(Crew in cell j) * P(Beep at cell i | Crew in cell j) / P(Beep at cell i)

We know that, P(Beep at cell i | Crew in cell k) = $e^{-\alpha(d(i,j)-1)}$

= P(Crew in cell j) * $e^{-\alpha(d(i,j)-1)}$ / P(Beep at cell i)

Taking the denominator, The probability of the beep at cell i can be rewritten as the sum of all the probability of crew member being in all the cells and the beep is received from cell i, we get

P(Beep at cell i) = $\sum_k$ P(Crew in cell k ^ Beep at cell i)

<u>By Conditional Factoring:</u>

P(Beep at cell i) = $\sum_k$ P(Crew in cell k) $*$ P(Beep at cell i | Crew in cell k)

We know that, P(Beep at cell i | Crew in cell k) = $e^{-\alpha(d(i,k)-1)}$

Therefore,

P(Beep at cell i) = $\sum_k$ P(Crew in cell k) $*$ $e^{-\alpha(d(i,k)-1)}$

Substituting the obtained expansion of the denominator in P(Crew in cell j | Beep at cell i) equation, we get

$$\boxed{P(\text{Crew in cell } j \mid \text{Beep at cell } i) = \frac{P(\text{Crew in cell } j) * e^{-\alpha(d(i,j)-1)}}{\sum_k P(\text{Crew in cell } k) * e^{-\alpha(d(i,k)-1)}}}$$

<u>CASE 2: NO BEEP IS HEARD WHEN THE BOT IS AT CELL i</u>

We know that during the start of the stimulation the crew member can't occupy the Bot's location due to which the game isn't over.

P(Crew in cell i) = 0

The remaining open cell in the ship will have the probability, P(Crew in cell j | No beep at cell i) can be derived by using a set of processes.

<u>By Conditional Probability:</u>

P(Crew in cell j | No beep at cell i) = P(Crew in cell j ^ No beep at cell i) / P(No beep at cell i)

= P(Crew in cell j) * P(No beep at cell i | Crew in cell j) /P(No beep at cell i)

We know that, P(No beep at cell i | Crew in cell k) = $1 - e^{-\alpha(d(i,j)-1)}$

= P(Crew in cell j) * (1 - $e^{-\alpha(d(i,j)-1)}$) / P(No beep at cell i)

Taking the denominator, the probability of no beep at cell i can be rewritten as the sum of all the probability of crew member being in all the cells and no beep is received from cell i, we get

P(No beep at cell i) = $\sum_k$ P(Crew in cell k ^ No beep at cell i)

<u>By Conditional Factoring:</u>

P(No beep at cell i) = $\sum_k$ P(Crew in cell k) $*$ P(No beep at cell i | Crew in cell k)

We know that, P(No beep at cell i | Crew in cell k) = 1 - $e^{-\alpha(d(i,k)-1)}$

Therefore,

P(No beep at cell i) = $\sum_k$ P(Crew in cell k) $*$ ($1 - e^{-\alpha(d(i,k)-1)}$)

Substituting the obtained expansion of the denominator in P(Crew in cell j | No beep at cell i) equation, we get

$$P(\text{Crew in cell } j \mid \text{No beep at cell } i) = P(\text{Crew in cell } j) * (1 - e^{-\alpha(d(i,j)-1)}) \; / \; \sum_k P(\text{Crew in cell } k) * (1 - e^{-\alpha(d(i,k)-1)})$$

STEP 4: CREW MEMBER BELIEF NETWORK UPDATION

The Belief network for the crew members are only updated once the crew has made a movement and mathematically, it can be derived as

P(Crew in cell j) $\forall$ cell j =

$$\sum_{a \,\varepsilon\, valid\_neighbours(j)} P(\text{Crew in cell } a) \;/\; len(\,valid\_neighbours(a))$$

We know that, P($Crew\ in\ cell\ a$) = $e^{-\alpha(d(i,a)-1)}$

$$P(\text{Crew in cell } j) \; \forall \text{ cell } j = \sum_{a \,\varepsilon\, valid\_neighbours(j)} e^{-\alpha(d(i,a)-1)} / \, len(\,valid\_neighbours(a))$$

STEP 5: BOT MOVEMENT

Upon moving into a new cell, if the bot remains operational and the execution continues, it's determined that there are no crew members present in that cell, although it previously held some probability of housing crew.

STEP 6: NORMALIZATION OF CREW MEMBER PROBABILITY

After the Bot made a movement and the crew member wasn't rescued, we can say that the probability of that cell containing the crew member should be updated to zero.

P(Crew in a new cell) = 0

The above situation creates an uneven distribution of probabilities as the remaining probabilities fail to total 1. To rectify this, we initiate a normalization process that distributes the probability mismatch throughout the matrix, ensuring that all probabilities sum up to 1 once again. This normalization procedure is explained below:

P(Crew in cell j | Crew not in a new cell) is being derived

By Conditional Probability:

P(Crew in cell j | Crew not in a new cell)

= P(Crew in cell j $\wedge$ Crew not in a new cell) / P(Crew not in a new cell)

By Conditional Factoring:

P(Crew in cell j | Crew not in a new cell)

= P(Crew in cell j) * P(Crew not in a new cell | Crew in cell j) / P(Crew not in a new cell)

We can say that, P(Crew not in a new cell | Crew in cell j) = 1 since the we have only one crew member and if its not in the new cell when the Bot moved to that new cell.

P(Crew in cell j | Crew not in a new cell) = P(Crew in cell j) / P(Crew not in a new cell)

P(Crew in cell j | Crew not in a new cell) can also be expressed differently by using the probability rule : P(Not A) = 1- P(A)

Therefore,

P(Crew in cell j | Crew not in a new cell) = P(Crew in cell j) / 1 - P(Crew in a new cell)

We know that,

P(Crew in cell j) = $e^{-\alpha(d(new\ cell, j)-1)}$

Hence,

$$\boxed{\textit{P(Crew in cell j | Crew not in a new cell)} = e^{-\alpha(d(new\ cell, j)-1)} / \textit{1 - P(Crew in a new cell)}}$$

STEP 7: ALIEN MOVEMENT

The mathematical calculations involved in the step of moving aliens are identical to those outlined in the moving alien's step of the One Alien One Crew Scenario described earlier in the report.

CONTRIBUTIONS:

Puja Sridhar:

- Implemented Bot 1 and Bot 2 ; Bonus
- Testing and analysis of the above bot's performance.

Vaishnavi Madhavaram:

- Implemented Bot 3, Bot 4 and Bot 5.
- Testing and analysis of the above bot's performance.

Kondaveti Deviram:

- Implemented Bot 6, Bot 7 and Bot 8.
- Testing and analysis of the above bot's performance.

All three of us worked on the report in their respective bots.