CS440: Project 1

Ivan Stadnik and Shreya Adupa

June 14, 2024

Question 1

The way bot 4 works is that for each adjacent cell of the bot it calculates the distance to the button and distance to the nearest fire cell, both of which have a weight associated with them. The weights are recalculated for each adjacent cell, depending on how close the adjacent cell is to the button and how close the adjacent cell is to the nearest fire cell. The closer the adjacent cell is to the button, the greater the weight it has compared to the other factor. The weights of each are calculated by adding the distance of each and then dividing the distance of each by the total distance. The cost of moving to an adjacent cell is calculated by adding weight*distance of adjacent cell to bot and weight*distance of adjacent cell to nearest fire cell. The bot moves to the adjacent cell with the least cost until it reaches the button cell.

Question 2

For bot 1, with 10 randomly generated q values, the average success rate for each q value of 20 trials was alway 1. The graph would be a straight line. For bots 2 and 3, with 10 randomly generated values q values, the average success rate decreased as the q value increased.

Question 3

There are multiple reasons for bot failures and the failures themselves depend on the type of bot. For bot 1, it calculates its path to the end all in the beginning and follows the path no matter what. So regardless of the change in the environment or fire movement, the path has been set in stone for the bot. Therefore, this can cause the bot to become trapped if it doesn't change its plan. Yes, there is a better decision that could have been made for bot 1. Bot 1 could have responded to changes in fire spread by recalculating at regular intervals or after each step to adapt to the changing environment. For Bot 2 and 3, even though their paths are adjusted to change based on the change in the environment, they still have a chance to fail if the fire spreads quickly

and surrounds them. To improve these bots, the algorithms could have been utilizing predictive analytics and using machine learning to help the bots figure out where and when the fire is going to spread to help make a successful decision.

Question 4

Other than the already optimized solution given for the bots of using Dijkstra's and BFS to find the best path, I can also add multi-bot coordination. To construct the ideal bot it's always a good way to adopt the strategy - two heads are better than one. So in the scenario of having multiple bots at the same time, if they are able to communicate with each other and share data of where the fire is or in which direction the fire is spreading then these alerts can help every bot make a better decision for itself and complete a successful mission. Also, in future models I can simulate the fire and calculate the probability of fire spreading to the next cell to help the bot make a more calculated decision when trying to figure out its path.