# CS 520 Final: Question 1 - Decision Making

Diksha Prakash - dp978

December 2019

**Abstract**

You are taking part in the 2019 Sheep Dog Bot competition. A sheep is released into an 8x8 cell grid, and you have two sheep dog bots under your control. Your goal is to pin the sheep in the upper left corner of the field (position (0,0))

## 1 Compute a strategy for controlling your dog bots that attempts to minimize the number of rounds needed to corner the sheep. How did you formulate this, and how did you solve it?

**Formulation:** For devising a strategy to corner the sheep, I emulated the movements of the dogs and the sheep on the chessboard. After multiple attempts, I figured out that once one of the four patterns shown in Figure 1 is achieved, the sheep can be cornered in finite number of steps. Now, the optimal strategy becomes to achieve one of these patterns. To form any of these patterns, I would first need to align the dogs in a diagonal fashion at (3,4) and (4,3) as that location is central w.r.t the location of the sheep.

**Solution :** The dog bots are initially controlled with the aim of aligning them in a diagonal fashion along the cells (3,4) and (4,3) i.e. they move independent of each other and the sheep, at each step ensuring that they are making valid moves such that two dogs are never at the same position and neither do they coincide with the sheep. The reason for the same is to get the dogs together at the centre first as these blocks are roughly equidistant from all the corners on the 8x8 grid.

After the dogs have reached the central location and are aligned along a diagonal, the next step is to minimize their (Manhattan) distance from the sheep. This is achieved by generating a list of all the neighbours for the dogs and finding that particular neighbour which minimizes the distance from the sheep.

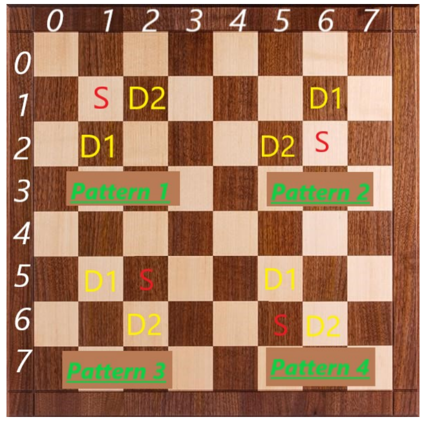


Figure 1: Possible Patterns for the orientation of Sheep and Dogs in the Grid

**NOTE:** For choosing whether the diagonal should be oriented either along $y = x$ line or $y = -x$ line, the program determines the Manhattan distance between each dog and the sheep. When these distances are unequal, the orientation of the diagonal is changed with an aim to make these distances equal.

The sheep moves freely in the grid to any feasible neighbour (i.e. to any neighbour except where the dogs are) and its the dogs whose movements are restricted so as to ensure their diagonal alignment and consistent reduction in the distance from the current location of the sheep.

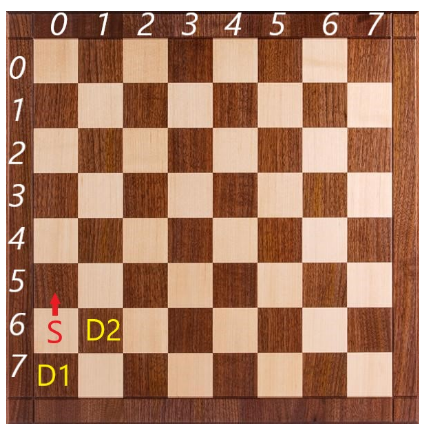


Figure 2: Illustration of Sheep Movement in Restricted Corner Case

Once any one of these patterns is formed, the final step is to corner the sheep. Note that with this pattern in place, we cut the possible locations for the sheep's next move in half. Eventually, the sheep is pinned to the closest corner i.e. either of the corners from (0,0), (0,7), (7,0), (7,7). If the sheep is at (0,0), the game is over. Otherwise, the sheep is then guided to (0,0) from whichever corner it is in. This is done in finite number of steps by giving the sheep only one location at a time to move to, as shown in Figure 2.

## 2 Given the initial state in the above example, how many rounds do you need (on average) to corner the sheep? Answer as precisely as possible.

**Logical Reasoning :** For the given initial arrangement, we can visualize the set of movements shown in Figure 3

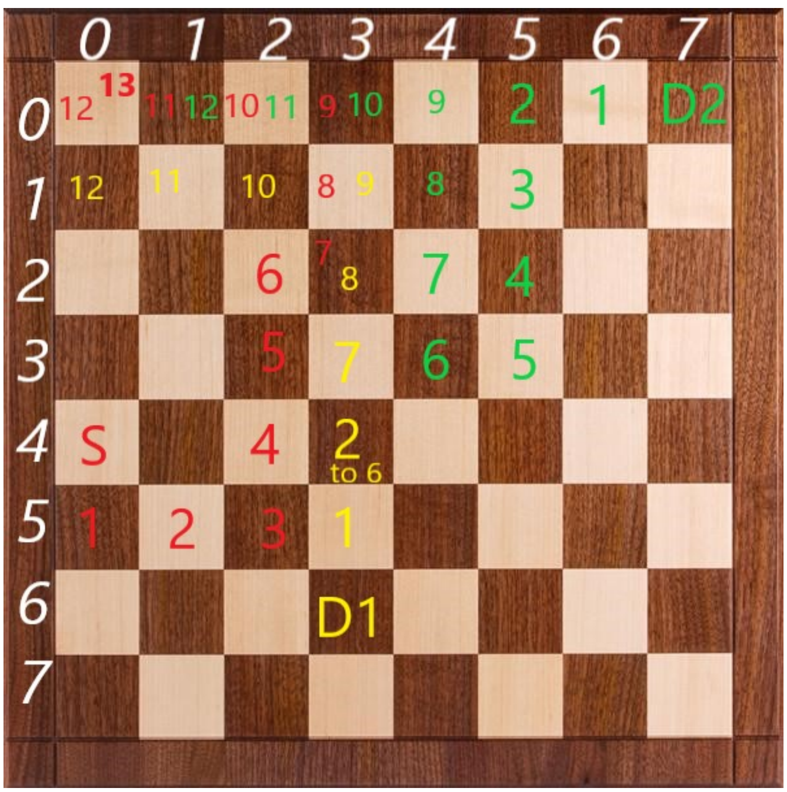


Figure 3: Set of movements for the given initial orientation

As we can see, the first six steps of both the dogs (indicated in 'Yellow' for Dog 1 and 'Green' for Dog 2) are aimed to bring the dogs in the diagonal orientation at locations (3,4) and (4,3). At the seventh step when the diagonal orientation has already been achieved, the dogs start following the sheep so as to minimize the Manhattan distance. Thus, we see that Dog 1 moves to (3,3)

and Dog 2 moves to (2,4) as the Sheep is at (2,3) and the Manhattan distance is best minimized along the chosen path. At this stage, Pattern 1 has been achieved (refer Figure 1).
After this, in the $8^{th}$ step, when the sheep moves to (1,3), Dog 1 moves to (2,3) and Dog 2 to (1,4)), maintaining the Manhattan distance.

After this, in the $9^{th}$ step, the sheep moves to (0,3) and the dogs follow along to (1,3) and (0,4). This continues till the sheep moves to (0,0) at the end of twelfth step and the dogs move to (0,1) and (1,0). As we can see that $9^{th}$ step onward, the sheep only has one feasible neighbour and hence is pinned to the corner in finite number of steps. Thus, at the $13^{th}$ step, the sheep has been cornered successfully.

**Program Output:**
On running the program for 500 iterations and calculating the mode of all the results obtained, we can infer the number of steps taken. The value so obtained is 13 (as shown in Figure 4), which matches our reasoning given above. Reason for taking 'mode' is that the mean gets skewed in certain iterations due to the following reasons:

1. Initially dogs are far away from the stipulated diagonal location.
2. After the dogs are in the diagonal orientation, the sheep keeps on moving back and forth causing the Manhattan distance from the either dogs to change. Due to this, the dogs keep flipping positions to ensure equal Manhattan distance.

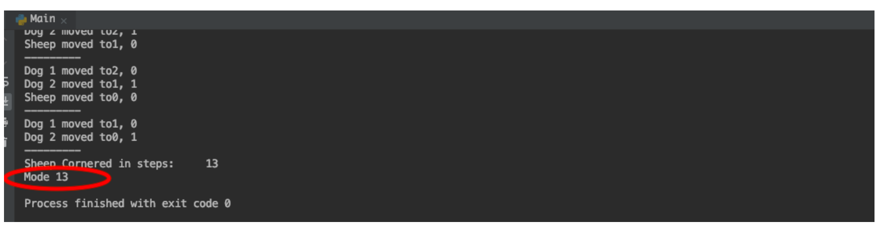


Figure 4: Program output for the number of steps taken to corner sheep for given location

# 3 What is the worst possible initial state (position of dog bots, sheep), and why? Justify mathematically.

With reference to Figure 5, if the Dog 1 is initially at (0,0), Dog 2 is at (7,0) and the sheep is at (7,7), it will take 7 steps for Dog 1 to reach the stipulated diagonal position (4,3) and 6 steps for Dog 2 to reach the stipulated diagonal position (3,4).

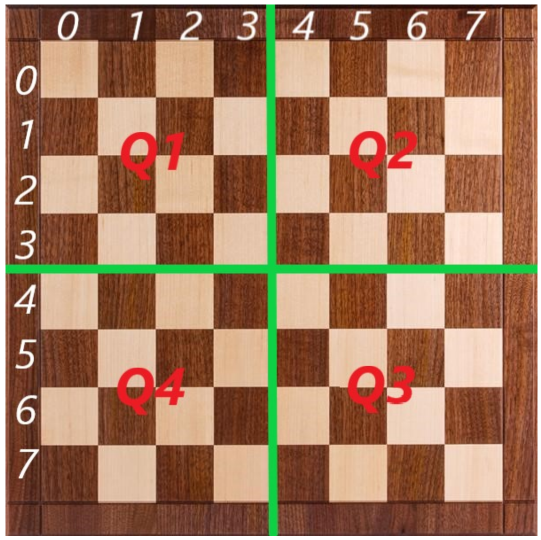


Figure 5: Grid Layout and Quadrant Representation

After these 7 steps, if the sheep is located in the lower region of Q3, the dogs in their diagonal orientation will follow it till the sheep is cornered to (7,7). After this, the sheep will have no option but to move along to edges to reach either (0,7) or (7,0) and ultimately (0,0). This action will result in 16 steps. Thus, the net steps taken so far will be $7 + 16 + x$ where, x is the number of steps taken after the $7^{th}$ step until the sheep is pinned to (7,7).

Ideally, this is the extreme case as a maximum of 7 steps are required for the dogs to reach the stipulated diagonals from any extremes and 16 is the maximum number of steps taken to pin the sheep to (0,0) from the most extreme end i.e. (7,7).

It is worth observing that the location of the sheep at unequal Manhattan distances from the either dogs causes the dogs to change the diagonal orientation (as explained in Question 1). If, in this case, the sheep keeps going back and forth between the current and previous location, the dogs will tend to change their orientation continuously at every such movement of the sheep.

Since the movement of the sheep is uniformly distributed among possible neighbours, it isn't expected that this sequence of movements will continue for a long time. However, when such a thing happens, the number of moves goes up, skewing up the mean considerably.

# 4 You are allowed to place your dog bots anywhere in the field at the start, then the sheep will be placed uniformly at random in one of the remaining unoccupied cells. Where should you place your dog bots initially, and why? Justify mathematically.

If given the liberty to place the Dogs anywhere in the field at the start, I would choose the location (4,3) and (3,4). The reason for this is the basis of my implementation.

This chosen location is roughly equidistant from all the corners of the grid. As a result, when we follow the strategy to move only in the direction which minimizes the Manhattan distance from the sheep, we not only follow the sheep but end up cutting the avenues for the sheep in that particular quadrant, making it easier to pin the sheep to any nearest corner.

It ensures minimum Manhattan distance to the sheep irrespective of the quadrant the sheep is in. In any other location, the dogs would be closer to the sheep only if the sheep happened to be near it. But, placing the dogs centrally would work no matter where the sheep is located. Thus, the optimal position to place the dogs is (3,4) and (4,3).

# 5 Do you think better strategies exist than the one you came up with? Justify.

We can generate a Neural Network relying on 'Reinforcement Learning' that will consistently contend against itself to explore, through trial and error, which actions are the best at winning rewards. It could be trained over multiple runs.

This neural can be regularized by including the number of steps in the error function. This would force the Neural Network to devise a strategy that will pin the sheep in the minimum number of steps.

The strategy used by my program was determined on the basis of few manual runs of the game and thus, I feel that a neural network trained on a huge number of runs might come up with a better strategy which reduces the number of steps taken to pin the sheep to goal state.