Mini Project 1 Report

Michael Lukacsko
mlukacsko3@gatech.edu

1 AGENT DESIGN

The agent in this assignment is designed to solve the Sheep and Wolves problem using a Breadth First Search (BFS) approach. The problem is represented as a series of states, where each state consists of the number of sheep and wolves on the left side of the river, along with the position of the boat. The goal is to move all sheep and wolves to the right side of the river without ever allowing the wolves to outnumber the sheep on either side at any time. Using a BFS method for this problem ensures that all possible states are explored systematically, guaranteeing that the shortest solution is found.

State generation plays a key role in this process. As such, for any given state, a 'generate_moves' function evaluates all valid moves based on the current position of the boat and the number of sheep and wolves on the left and right sides of the river. If the boat is on the left side, the function generates potential moves such as transporting 1 or 2 sheep, 1 or 2 wolves, or 1 sheep and 1 wolf together to the right side. If the boat is on the right side, the agent evaluates the animals on the right side and generates moves to bring animals back to the left side. For each move, the agent checks if the number of sheep and wolves involved is valid and adheres to the problem constraints.

After generating a possible move, the agent tests the new state for validity. A valid state is one where the number of sheep and wolves on both sides of the river is within allowable bounds (i.e., non-negative and within the total number of animals) and where wolves never outnumber sheep on either side unless no sheep are present. If a state passes this test, it is added to the BFS queue for further exploration. Moreover, the agent tracks visited states to avoid revisiting them, ensuring that it doesn't fall into loops or explore redundant paths, thereby improving efficiency. This methodical approach allows the agent to systematically explore the state space, ensuring it finds the solution while adhering to the constraints of the problem.

2 AGENT PERFORMANCE

Referencing figure 1 below, it is clear the agent generally performs well, with a steady increase in the number of moves as the number of sheep and wolves increases. However, there are notable spikes in performance when there is only 1 wolf and a significantly larger number of sheep, such as in the cases of 250S-1W, and 500S-1W. It is most likely that these spikes occur because managing many sheep with only 1 wolf creates a more complex problem. As a result, the agent must ensure that the wolf is never left alone with more sheep than it can handle on either side of the river, which results in many back-and-forth trips. The constraints of the problem—only moving one or two animals at a time—further complicate the problem, requiring the agent to move small groups of sheep while also avoiding invalid states.

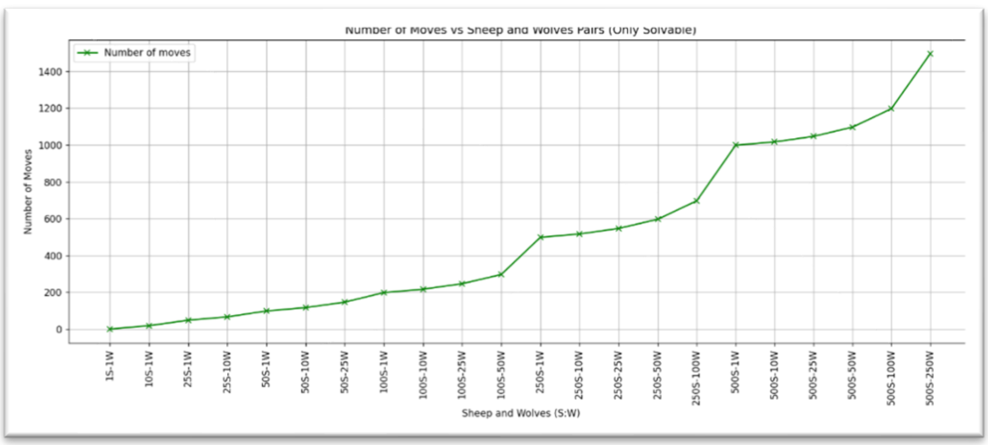

Figure 1—Agent Performance. Sheep to Wolves vs Number of Moves.

3 AGENT EFFICIENCY

Figure 2 below visually confirms the agent's efficiency decreases as the number of sheep and wolves increases. For smaller numbers, the time taken to solve the problem remains very low and nearly constant, indicating that the agent can quickly process these cases with minimal computation. However, as the number of sheep and wolves rises, particularly in cases involving large numbers of sheep and wolves, the time taken begins to increase exponentially. The most significant increase occurs when the problem size reaches 500S-250W, where the time taken peaks sharply. This suggests that while the agent handles smaller problem instances efficiently, its performance degrades for larger, more complex cases due to the increased complexity in decision-making and the larger search space required to find a solution. Hence, as the number of animals rises, the agent needs more time to explore valid moves and avoid invalid states, leading to a noticeable decline in efficiency.

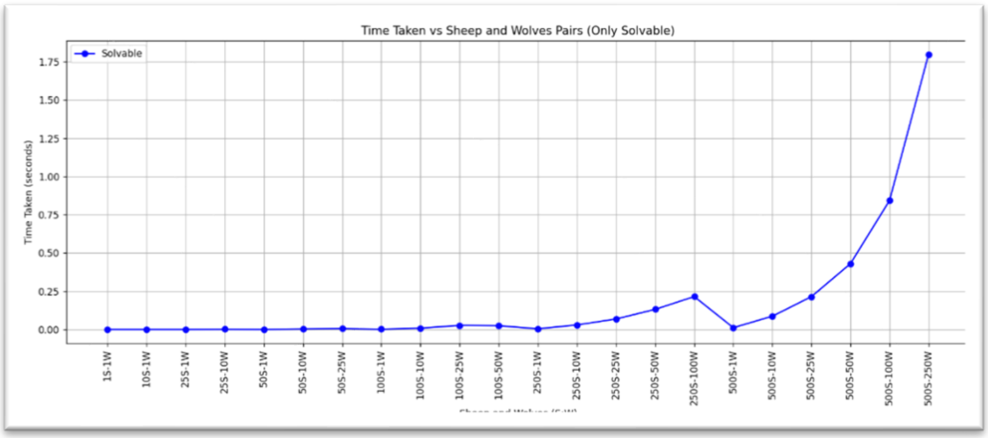

Figure 2 — Agent Efficiency. Sheep to Wolves vs Time to Solve

4 IMPROVING EFFICIENCY

By using BFS for this problem, it is guaranteed that the agent finds the shortest possible solution if one exists. In this BFS implementation, there are a few checks that help improve the efficiency:

1. **Avoid Revisiting States**: The agent uses a set to keep track of previously visited states. A look at this this set when looking at available states prevents the agent from revisiting the same state multiple times, which reduces redundant work and helps prune the search space. By avoiding loops in the search, the agent ensures that it doesn't waste time exploring already-known outcomes. This is particularly useful in problems with large numbers of sheep and wolves.

2. **Valid Move Generation**: The agent generates valid moves based on the current state of the boat and sheep and wolves. This limits the number of moves to only those that are logically possible. It also checks whether the move is valid by ensuring that wolves do not outnumber sheep on either side of the river. This ensures that only legal moves are explored, further improving efficiency by discarding invalid states early on.

5 AGENT VS HUMAN

The agent’s approach differs significantly from how a human might solve the problem for several reasons. Firstly, a human would likely rely on intuition and pattern recognition to identify moves that are valid by focusing on strategies pertaining to constraints such as ensuring that the wolves never outnumber the sheep on either side of the river and planning moves based on these limitations.

In contrast, the agent methodically explores every possible move to find the shortest solution. While this BFS approach guarantees finding the correct answer, it lacks the intuitive leaps a human would make. Another key difference is that the agent systematically checks the validity of each state, ensuring that no move violates the constraints. A human solving this problem would apply the rules more fluidly and without needing to explicitly check every state.

Because of the differences in the human and agent approach, it can be said that while the agent is thorough and ensures the correct solution, it approaches the problem differently from a human, who would likely solve the problem more intuitively and with fewer intermediate checks but may not always find the shortest path. The agent’s method is more systematic, while a human approach would rely on experience and intuition. This contrast demonstrates the strengths and weaknesses of both methods: the agent's systematic exploration guarantees optimal solutions but can be slower and computationally expensive, whereas the human's heuristic-based approach can solve problems quickly but may miss the most efficient path or make occasional errors in complex scenarios