CS7637: Knowledge-Based AI:

Mini-Project 1

Yaxin Yang

Yyang894@gatech.edu

***Abstract—*** The SemanticNetsAgent is designed to solve a classic river crossing problem involving sheep and wolves. The agent aims to move all the sheep and wolves from one side of the river to the other while following specific constraints.

# 1 How does the agent work?

The BlockWorldAgent uses a form of guided tree search to solve the Block World problem. It starts with an initial arrangement of blocks and uses Breadth-First Search (BFS) to explore possible moves to reach the goal arrangement. The agent generates possible sub-states from the current state, calculates a weight for each sub-state based on its closeness to the goal state, and prioritizes exploring the sub-state with the highest weight.

## 1.1 Generate & Test? Means-Ends Analysis?

The agent primarily uses a Generate & Test approach, where it generates possible sub-states (moves) and tests them by calculating a weight (score) indicating how close the sub-state is to the goal. The agent does not explicitly perform Means-Ends Analysis but implicitly does something similar by calculating weights for the sub-states to guide the search towards the goal.

# 2 PERFORMANCES

The agent can successfully solve the problem in many cases, especially when the number of blocks and the complexity of the arrangements are not too high. However, since it uses BFS and calculates weights for each generated sub-state, it may struggle with efficiency and memory usage in cases with a large number of blocks and complex arrangements, leading to a combinatorial explosion of possible states.

How efficient is your agent? How does its performance change as the number of blocks?

The agent's efficiency is dependent on the number of blocks and the complexity of the arrangements. As the number of blocks increases, the number of possible states and moves grows exponentially, leading to increased computation time and memory usage. The agent's performance can degrade significantly with an increase in problem size.

Does your agent do anything particularly clever to try to arrive at an answer more efficiently?

The agent calculates weights for each generated sub-state based on its closeness to the goal state and prioritizes exploring the most promising sub-state. This strategy helps in guiding the search towards the goal and can be considered a heuristic approach, making the agent more efficient compared to a naive BFS.

How does your agent compare to a human? Does your agent solve the problem the same way you would?

The agent uses a systematic approach to explore possible moves and prioritizes them based on calculated weights, which is a logical and methodical strategy. Humans might use a similar strategy of evaluating and prioritizing moves, but they also leverage intuition, spatial reasoning, and pattern recognition, which the agent lacks. While the agent’s approach is more exhaustive and deterministic, a human might solve the problem more creatively and potentially more efficiently for certain complex arrangements.

# 3 EFFICIENCIES

The efficiency of the agent decreases as the number of animals increases. The time complexity of BFS is O (V + E), where V is the number of states, and E is the number of transitions. Thus, as the number of animals increases, the state space grows exponentially, making the algorithm slower.

As the number of animals (sheep and wolves) rises, the number of possible states and edges grows exponentially. This not only increases the time needed to find a solution but also the memory requirements to keep track of visited states.

The BFS queue may become extremely large, causing significant memory usage, and possibly leading to memory overflow errors in systems with limited resources. In addition, storing visited states also becomes computationally expensive as the state space grows, adding to both time and memory requirements.

# 4 Clever

The most straightforward optimization in the agent is the "early exit" strategy. As soon as a goal state is encountered, the BFS algorithm terminates, avoiding unnecessary exploration of the remaining state space. This helps in reaching a solution faster when the goal state is reachable before exploring all possible states. The agent tests for state validity as soon as it generates a new state. This ensures that only valid states are appended to the BFS queue and reduces the number of states that need to be processed.

# 5 COMPARISONS TO HUMAN

## 5.1 Speed and Efficiency

The most obvious advantage of the agent is speed. Once the algorithm is initiated, it can examine states and find a solution much faster than a human can manually analyze the problem. A computer can process the BFS algorithm in a matter of seconds, whereas a human might take minutes or even longer to manually solve the same problem, especially as the number of animals increases.

## 5.2 Algorithmic Versus Intuitive

The agent takes a systematic, algorithmic approach to solve the problem, explore all possible states in a breadth-first manner. In contrast, humans typically employ a more intuitive, heuristic-based strategy, using their understanding of the problem to eliminate unlikely moves and focusing on the most promising paths. This might lead to faster solutions in some cases but could also result in missing the optimal path if the intuition is incorrect.

## 5.3 Error-prone

Humans are more likely to make errors in complex and large tasks, especially when they involve a lot of state tracking and mental computation. The agent, once programmed correctly, is not susceptible to such errors and will consistently produce the correct output for any given input. Just image when sheep and wolf are both in 3 digits. Can humans provide the number of movements in minutes or seconds?

## 5.4 Insight and Understanding

While the agent can find a solution, it doesn't understand the problem in the way a human does. A person can often provide a rationale for why a particular solution is more efficient or why a certain strategy won't work, which can be important for teaching or explaining the problem to others.