CSci 4511W: Intro To AI
Homework 2
Google Colab

1. **The A* algorithm sorts the nodes in the frontier by increasing value of f. When multiple nodes have the same value, the order in which they are sorted is arbitrary. Could you suggest a better way of sorting the nodes that have the same value? Explain your reasoning.**
   A better way of sorting the nodes that have the same value would be to prioritize nodes with a lower g(n) which is the cost from the start node to n. This is because a lower g-value means the node has reached its position with less cost, which may lead to a more optimal path. Additionally, we can use the heuristic value to prioritize the nodes (prioritize the nodes with lower h-value), as a lower h-value signifies that the node is estimated to be closer to the goal, reducing the number of nodes explored.

2. **You want to reduce the memory used by A*. You come up with the following idea: you keep in the queue only the N best nodes (i.e, the nodes with lower costs), for some positive value of N. When the queue is full and a new node has to be stored, the worst node is deleted from the queue and removed from consideration.**
   a. **If you use an admissible heuristic, will the modified algorithm find the optimal solution? Explain why (or why not). Are there any additional constraints? [hint: consider what can happen depending on the value of N].**
      No, with an admissible heuristic, the modified algorithm may not find the optimal solution. A* search algorithm guarantees optimality because it explores all possible optimal paths. If we remove nodes, we risk discarding a necessary step in the optimal path, especially if the N value is too small.

   b. **If you use a perfect heuristic, i.e. for all n h(n)=h*(n) , where h*(n) is the cost of the optimal path from n to goal, will the modified algorithm find the optimal solution?**
      Yes, with a perfect heuristic, the modified algorithm will find the optimal solution if h(n)=h*(n), because A* will expand the best possible path without backtracking. Since we effectively implement greedy best-first search with guaranteed correctness, removing worse nodes does not impact the result.

3. **Answer the following questions briefly but precisely:**
   a. **Why the cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem?**
      A relaxed problem is one where constraints are removed, making it easier than the original problem. Since the cost of an optimal solution to a relaxed problem is

always less than or equal to the cost of the optimal solution to the original problem, it never overestimates the true cost, making it admissible.

b. **Does the fact that A* is "optimally efficient" mean that A* will never expand more nodes than any other algorithm? Explain precisely.**
No, A* is optimally efficient in the sense that no other algorithm using the same heuristic function will expand fewer nodes while guaranteeing optimality. However, another algorithm using a better heuristic (closer to the true cost) could expand fewer nodes.

c. **Why is it that any node in the frontier with f(n) < f*(n) (the cost of the optimal solution path) will eventually be selected for expansion by A*?**
A* expands nodes in increasing order of f(n). If a node has f(n) < f*(n), it means the node has a lower estimated cost than the optimal path. Since A* explored nodes with the lowest f-values first, it must eventually expand that node before reaching the goal.

4. **This question is on genetic algorithms.**
   a. **What is the role of the fitness function used in genetic algorithms?**
   The fitness function evaluates how good a candidate solution is. It helps guide evolution by selecting individuals that perform better according to the problem's objective.

   b. **Could genetic algorithms be used without a fitness function? Why or why not?**
   No. Without a fitness function, there would be no way to determine which individuals are better suited for survival and reproduction. The algorithm would evolve randomly without direction, making it ineffective.

   c. **What happens if the crossover point is not the same for all the individuals of the population? Explain.**
   If crossover points vary, it increases the genetic diversity, potentially leading to a more effective search of the solution space. However, it can also introduce randomness that disrupts beneficial genetic structures, making convergence to an optimal solution slower or less predictable.

Note: Used Grammarly for sentence structure, and grammar purposes.