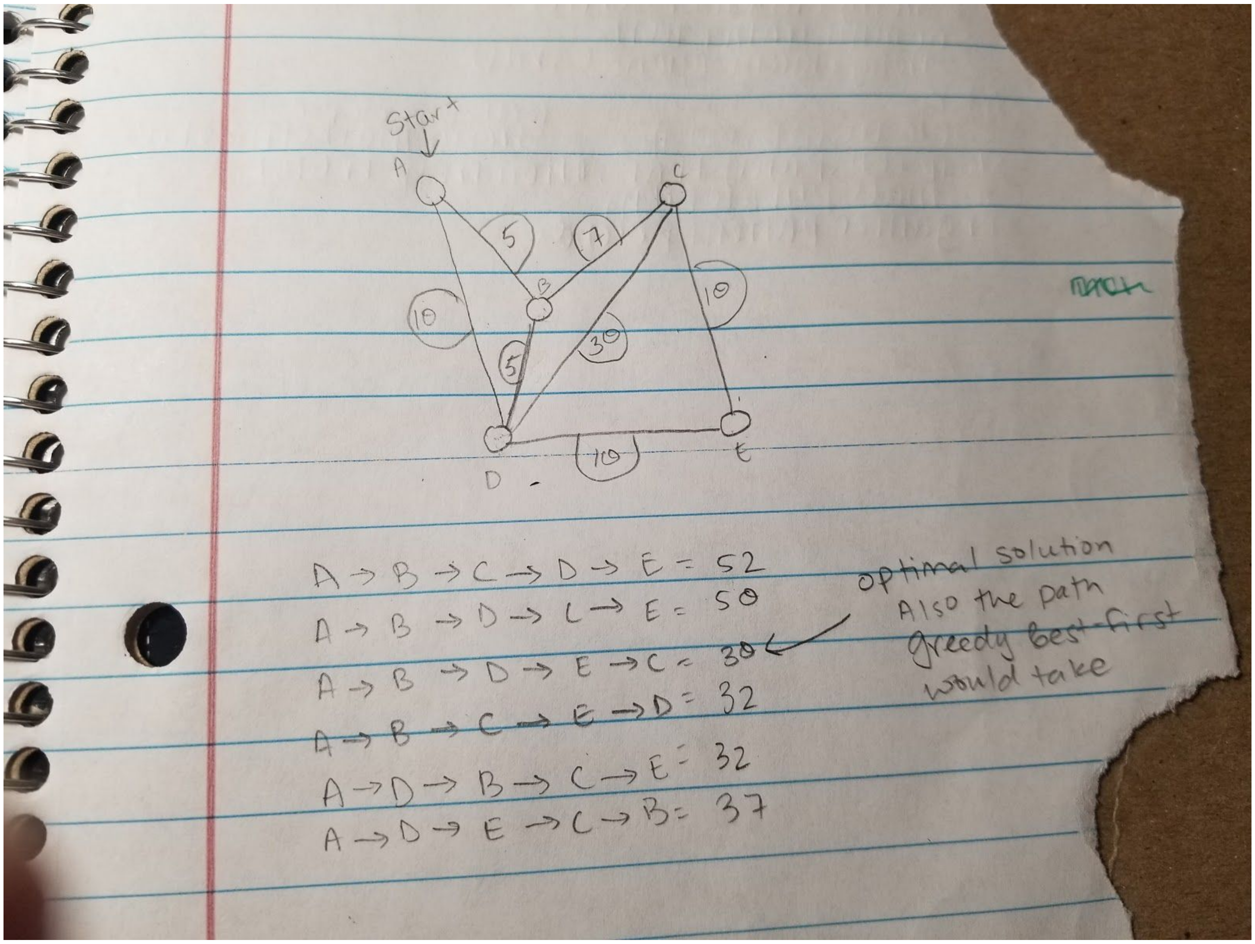
Nadya Postolaki
CSCI 4511W Homework 2

Written questions (75 points)
[15 points] You are given the following heuristic function for the Traveling SalesPerson Problem (TSP): h(n) = distance to the closest unvisited city.
Answer the following questions:

Is this heuristic admissible? Explain your reasoning.

Yes, because the definition of an admissible heuristic is one that never overestimates the cost to reach the goal. Since h(n) equals exactly the distance to the closest unvisited city, this satisfies the definition for an admissible heuristic. The distance to the closest unvisited city will never be more than the other unvisited cities.

Show an example of a TSP where the Greedy-Best-First algorithm with this heuristic finds an optimal solution.

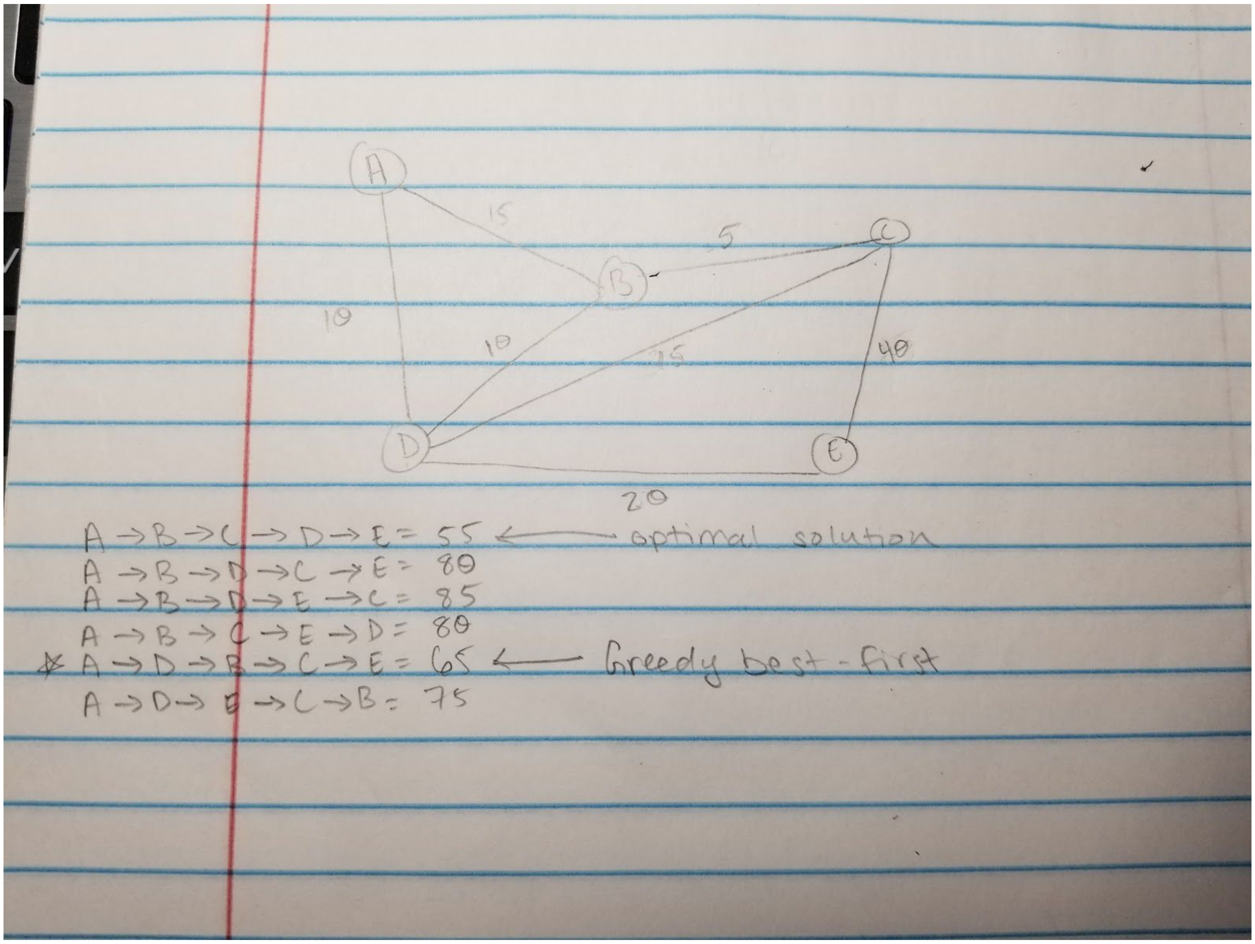


A → B → C → D → E = 52
A → B → D → C → E = 50
A → B → D → E → C = 30
A → B → C → E → D = 32
A → D → B → C → E = 32
A → D → E → C → B = 37

optimal solution
Also the path greedy best-first would take

{INSERT DRAWING HERE}

Show an example of a TSP where the Greedy-Best-First algorithm with this heuristic does not find an optimal solution.



A → B → C → D → E = 55 ← optimal solution

A → B → D → C → E = 80

A → B → D → E → C = 85

A → B → C → E → D = 80

☆ A → D → B → C → E = 65 ← Greedy best - first

A → D → E → C → B = 75

{INSERT DRAWING HERE}

Is Greedy-Best-First search with an admissible heuristic guaranteed to find an optimal solution?

No, since Greedy-Best-First algorithm exands nodes with minimal h(n), it only looks at the local costs and doesn't take into account the distance already traveled and cannot back-track to look for other paths that may be optimal, this is proven with the previous image where the heuristics of the closest unvisited city were expanded by greedy search and yet did not provide an optimal solution.

Propose an admissible heuristics for the TSP, different from the heuristics given above.

h(n) = distance to the closest unvisited city + approx distance to travel all the unvisited cities

[10 points] Explain with your own words why any heuristic which is an optimal solution to a relaxed problem is admissible and consistent.

A relaxed problem is one with fewer restrictions on the actions. Per section 3.6 on page 105 the cost of an optimal solution to a relaced problem is an admissible heuristic for the the original problem, this is because the relaxed problem adds shortcuts to the original problem which means it underestimated the solution for the original problem, which satisfies the definition for an admissible heuristic. The heuristic is therefore consistent because it obeys the triangle inequality since it is an exact cost for the relaxed problem.

[10 points] What are the disadvantages of using a heuristic function for A* which is not consistent? Be precise and provide an example to support the points you are making.

A* requires extra bookkeeping to ensure optimality becuse the same node can be expanded more than once when using A*- Martelli's algorithm B (1977). Consistent heuristics always have the minimal g value when nodes are expanded by A* for the first time, whereas inconsistent heuristics might expand the nodes again when a shorter path has reached them.

[10 points] In A* the nodes that have been generated but not yet expanded are sorted according to the value of f(n) = g(n) + h(n), i.e. the sum of the cost from the start to node n plus the estimated cost from n to goal using admissible heuristic h(.). Nodes that have the same f(.) value are ordered arbitrarily.

Propose a domain independent criterion that could be used to order the nodes with the same f(.) value. Explain your answer.

Arranging the nodes by g(.) value going down, because if some nodes are only a percentage of the cost, we can determine whether the total cost is certain and how close we are to the goal. A lower percentage for g(.) means most of the cost is estimate whereas a higher percentage for g(.) means the cost is more concrete.

[15 points] Discuss the possible advantages of the following state-space search strategy: obtain by some method a path to a goal node and its associated cost f(Goal)=C. This cost is not necessarily minimal but it gives an upper bound on the minimal cost. Now use A* with an admissible h function and discard immediately any frontier nodes reached whose f values are greater than C.

Explain if the modified A* algorithm with this strategy is guaranteed to find an optimal solution if one exists or not. Be precise.

Since we know that this is a possible path and it is set as the upper bound on the minimal cost, we know that this strategy is guaranteed to find an optimal solution if it exists because any frontier nodes larger than C are discarded since they would clearly be greater than the upper bound. This is because since the heuristic is admissible, the true cost would not be overestimated.

Explain if the fact that the algorithm discards some of the frontier nodes (i.e. nodes in the frontier) means that fewer nodes are expanded. Be short but precise.

Since it compares the f values of the frontier nodes to the upper bound C, if the frontier nodes are larger than C they would get discarded and therefore not expanded since there is no need to look into the nodes when we know a more optimal solution exists that is less than or equal to the upper bound that was set.

Does this strategy reduce the total storage requirements? Explain your reasoning.

Yes because less nodes are expanded and therefore we don't need to backtrack along them, but only along the smaller costs to find the optimal solution.

[15 points] Answer these questions with your own words. Be precise.

Does the fact that A* is ``optimally efficient'' mean that A* will never expand more nodes than any other algorithm?

A* is optimally efficient for any given consistent heuristic because no other optimal algorithm is guaranteed to expand fewer nodes- this is also because any algorithm that does not expand all nodes with $f(n) < C^*$ runs the risk of missing the optimal solution.

Why 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*?

Because in search of the optimal solution, if any node is smaller than the anticipated value of f (it must be less than or equal to), it is in the best interest to expand the node to check whether an optimal solution exists. As soon as the f value is larger than anticipated, the node is discarded and we backtrack to the next node.

Is it true that all admissible heuristics are equal in the sense that A* will search the states in the same order no matter what the heuristic is?

No, because some heuristics will cause the search to have a uniform cost search making it breadth first search, whereas other heuristics may lead straight to the goal.

Programming questions (25 points)
For this question we will use the aima software for the TSP problem.
[5 points] Compare the performance of the following uninformed search algorithms:

breadth_first_tree_search,
breadth_first_graph_search,
depth_first_graph_search,
iterative_deepening_search,

depth_limited_search with a depth limit of 10,

on the Romanian map to solve these problems:
find a path from Arad to Bucharest
find a path from Oradea to Neamt
find a path from Eforie to Timisoara

To compare the performance, you can use the function compare_graph_searchers() in python or compare-search-algorithms in lisp.

[5 points] Add or modify the search functions to compute the cost of the solution found by each algorithm and print the cost for each algorithm when you compare the algorithms.

[15 points] Compute the average cost and standard deviation of the cost of the paths found by the different algorithms used when you compared their performance. Write the code so that it will work for any number of algorithms and will print the average and standard deviation at the bottom of the table printed by the comparison function.

You need to submit one or more files with your program and a file that shows the output. Do not modify the existing aima files, but create a new file with any changes you need.