
1: Basics

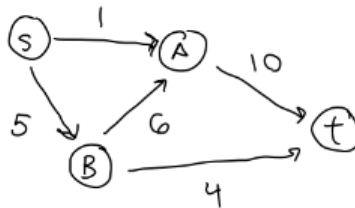
Fill in the definition of each of the following.

1. Traveling Salesman Problem:
2. A* Search:
3. Admissible heuristic:

Find the shortest path for the following graph using:

1. A* Search

state	$h(n)$
S	7
A	6
B	4
t	0



2: Practice Problems

1. In the A* search problem above, notice that the heuristic $h(u)$ was admissible because it always underestimates the $u \rightarrow t$ distance. Show (via an example) that a non-admissible heuristic may lead to A* terminating with the wrong 'shortest' $s \rightarrow t$ path. *Hint: start from the graph above, and make $h(B)$ an overestimate instead.*
2. At the very end of the A* lecture, we solved the Traveling Salesman Problem on a graph G by constructing a state graph S where each node corresponded to a partial tour in TSP, and then running A* shortest path on S .

However, to do this, we needed an admissible heuristic for $h(a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k)$. Prove that the heuristic

$$h(a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k) = \text{cost}(\hat{T}),$$

where $\hat{T}$ is an MST on all nodes except $a_2, \dots, a_{k-1}$ is admissible.

Helpful Videos:

- Dijkstra's algorithm in 3 minutes, Micheal Sambol, https://youtu.be/_lHSawdgXpI?si=aoGcZiufBrnkb_Ab
- A Star algorithm — Example — Informed search — Artificial intelligence — Lec-21 — Bhanu Priya, Education 4u, <https://youtu.be/PzEWHH2v3TE?si=62zaQuHsIdRDqWp2>

3: Solutions

Fill in the definition of each path finding algorithm.

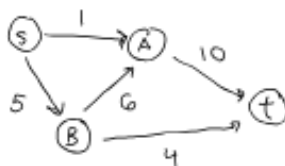
1. Traveling Salesman Problem: Given a graph with nodes and weighted edges, what is the shortest possible transversal that visits each node once, and then returns to the original city?
2. A* Search: This is used to find the shortest path between a starting s and end t node in a graph. We use the function $f(n) = g(n) + h(n)$, where $g(n)$ is the cost of the current path from s to n , and $h(n)$ is the heuristic, or estimated cost of the path from n to the end, t . $f(n)$ is the cost of the path so far to t .
3. Admissible heuristic: A heuristic is admissible if $h(n)$ is never larger than the true distance between n and t . We denote the true distance from n to t as $h^*(n)$.

Find the shortest path for the corresponding graph :

1. A* Search

A*

state	$h(n)$
S	7
A	6
B	4
t	0



1) First, calculate the value of $S \rightarrow A$ and $S \rightarrow B$. Store values in an array.
 $f(n) = g(n) + h(n)$

$$S \rightarrow A = \text{dist}(S, A) + h(A) \\ 1 + 6 = 7$$

$$S \rightarrow B = \text{dist}(S, B) + h(B) \\ 5 + 4 = 9$$

path	$f(n)$
$S \rightarrow A$	7
$S \rightarrow B$	9

2) Look at the lowest $f(n)$ in the array. explore the next node of that path update the array.

path	$f(n)$
$S \rightarrow A$	7 ← lowest
$S \rightarrow B$	9

$$S \rightarrow A \rightarrow t = \text{dist}(S-A-t) + h(t) \\ 11 + 0 = 11$$

path	$f(n)$
$S \rightarrow A$	7
$S \rightarrow B$	9
$S \rightarrow A \rightarrow t$	11

3) $S \rightarrow B$ has the next lowest value, so we explore that path and update the array.

$$S \rightarrow B \rightarrow A = \text{dist}(S-B-A) + h(A) \\ 11 + 6 = 17$$

$$S \rightarrow B \rightarrow t = \text{dist}(S-B-t) + h(t) \\ 9 + 0 = 9$$

path	$f(n)$
$S \rightarrow A$	7
$S \rightarrow B$	9
$S \rightarrow A \rightarrow t$	11
$S \rightarrow B \rightarrow A$	17
$S \rightarrow B \rightarrow t$	9

4) since $S \rightarrow B \rightarrow t$ is our smallest value, and it already ends in t , we are done. $S \rightarrow B \rightarrow t$ is the shortest path.



4: Practice Problems

1. In the A* example, let $h(B) = 10$. Then A* will of course start at s , but in the initial step, $f(A) = 1 + 6 = 7$,

while $f(B) = 5 + 10 = 15$. Thus, the next node to be removed from the heap will be A . After A is removed, $f(t) = 1 + 10 = 11$, but that's smaller than $f(B) = 15$, so we will remove t next. But since we have removed t , we can end the algorithm, with an $s \rightarrow t$ path of length 11. The true shortest path was of course $s \rightarrow B \rightarrow t$, but it wasn't found because $h(B) > h^*(B)$.

2. Notice that in order to finish the tour started by $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k$, we need an $a_k \rightarrow a_1$ path P that goes through all nodes except $a_2, a_3, \dots, a_{k-1}$ without repeating any nodes. But such a path is obviously also a spanning tree on all nodes except $a_2, a_3, \dots, a_{k-1}$ (because a line is also a tree, since it has no cycles). But $\hat{T}$ is a *minimum* spanning tree on those same nodes, and thus must have cost no greater than any other spanning tree. Thus, $\text{cost}(\hat{T}) \leq \text{cost}(P)$, so the heuristic is admissible.