School of Computer Science & Software Engineering
The University of Western Australia
Crawley, Western Australia, 6009.
CRICOS Provider Code: 00126G

Algorithms (CITS3210)

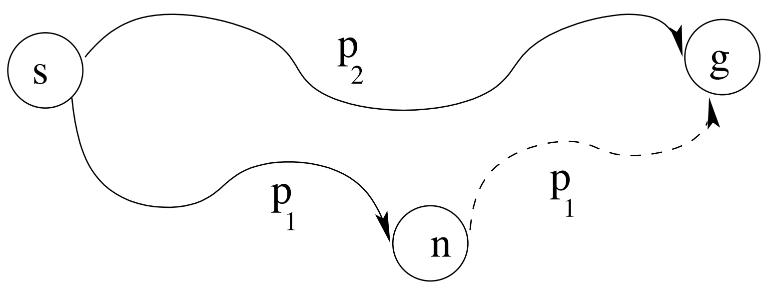
Addendum to Graphs Lecture Notes

See: A* Algorithm, page 127.

Since the $A*$ algorithm is not covered in the textbook, a more thorough proof given by Dr Gareth Lee, who taught Algorithms in 2004, is included here. It also acts as an example of how to create a proof by contradiction.

Theorem If $e(v,g) \leq \delta(v,g)$ then the A* procedure is guaranteed to produce an optimal path and we say function e is admissible.

Proof Consider the case of an A* procedure finding the shortest path from a source vertex s to a goal vertex g. In which case we must have just extracted g from the priority queue causing the procedure to terminate (see the code on page 125).



Let p_1 be the optimal path from s to g, such that

$$\text{length}(p_1) = \delta(s,g). \tag{1}$$

Let p_2 be a sub-optimal path from s to g, such that

$$\text{length}(p_2) > \delta(s,g). \tag{2}$$

Also recall the notation used in the previous lecture notes:

$d[u]$ is the distance to u along a path from s.

$key[u]$ is the priority value used to order u in the priority queue and

$$\begin{aligned} \text{key}[u] &= d[v] + w(v,u) + e(u,g), & (3)\\ &= d[u] + e(u,g). & (4) \end{aligned}$$

$\delta(u, v)$ is the shortest distance between vertices u and v in the graph.

$e(u, v)$ is the estimated distance from vertex u to v in the graph.

Proof by contradiction.

Consider a vertex n on path p_1.

If e is admissible, this implies,

$$\begin{aligned} e(n, g) &\leq \delta(n, g), & (5)\\ \Rightarrow d[n] + e(n, g) &\leq \delta(s, n) + \delta(n, g), & (6)\\ \Rightarrow d[n] + e(n, g) &\leq \delta(s, g), & (7)\\ \Rightarrow \text{key}[n] &\leq \delta(s, g), & (8) \end{aligned}$$

since we already know the distance from s to n along p_1 and therefore $d[n] = \delta(s, n)$ and because of the definition of key from Eq. (4).

But g has been extracted from the priority queue before n, which implies,

$$\begin{aligned} \text{key}[g] &\leq \text{key}[n], & (9)\\ \Rightarrow \text{key}[g] &\leq \delta(s, g), & (10) \end{aligned}$$

by considering Eqs. (9) and (8).

But g is the goal of this search so, by definition, $e(g, g) = 0$ and since

$$\begin{aligned} \text{key}[g] &= d[g] + e(g, g), & (11)\\ \Rightarrow \text{key}[g] &= d[g]. & (12) \end{aligned}$$

But $d[g]$ is the length to g along path p_2, so

$$\begin{aligned} \text{length}(p_2) &= d[g], & (13)\\ \Rightarrow \text{length}(p_2) &\leq \delta(s, g), & (14) \end{aligned}$$

by considering Eqs. (13), (12) and (10).

This final statement, Eq. (14), **contradicts** our original assertion (in Eq. (2)) and therefore we can see that, so long as e is admissible we must find the shortest possible path from s to g.