A* and heuristic

A* always requires a heuristic, it is defined using heuristic values for distances. A* in principle is just the ordinary **Dijkstra algorithm** using heuristic guesses for the distances.

The heuristic function should run fast, in O(1) at query time. Otherwise you won't have much benefit from it. As heuristic you can select **every function** h for which:

- h is **admissible**: h(u) <= dist(u, t) (never overestimate)
- h is **consistent**: h(u) <= cost(u, v) + h(v) (triangle inequality)

Straight-line heuristic

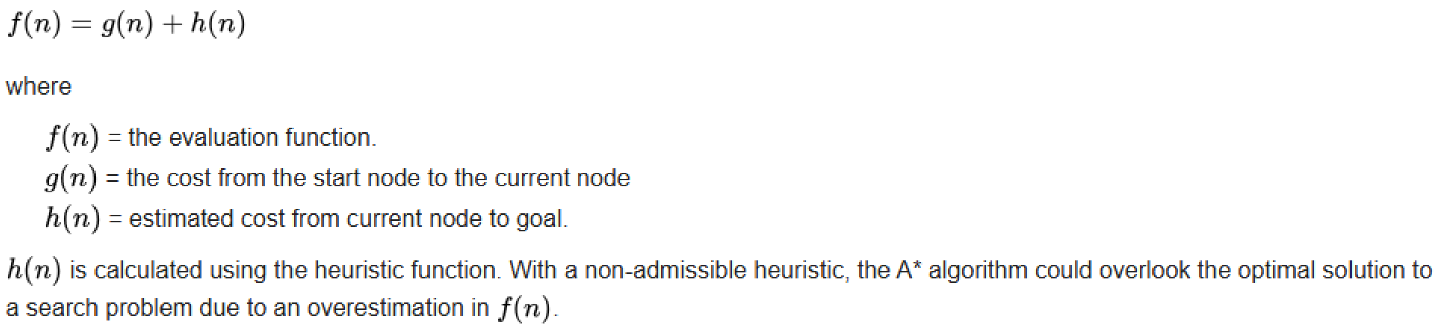
The straight-line distance (or as-the-crow-flies) is straightforward and easy to compute. For two nodes v, u you know the exact location, i.e. **Longitude** and **Latitude**.

You then compute the **straight-line** distance by defining h as the Euclidean distance

Admissible heuristic

In computer science, specifically in algorithms related to pathfinding, a heuristic function is said to be **admissible** if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path.[1] In other words, it should act as a lower bound.

It is related to the concept of consistent heuristics. While all consistent heuristics are admissible, not all admissible heuristics are consistent.



$$f(n) = g(n) + h(n)$$

where

$f(n)$ = the evaluation function.

$g(n)$ = the cost from the start node to the current node

$h(n)$ = estimated cost from current node to goal.

$h(n)$ is calculated using the heuristic function. With a non-admissible heuristic, the A* algorithm could overlook the optimal solution to a search problem due to an overestimation in $f(n)$.

Formulation

n is a node
h is a heuristic
$h(n)$ is cost indicated by h to reach a goal from n
$h^*(n)$ is the optimal cost to reach a goal from n

$h(n)$ is admissible if, $\forall n$

$$h(n) \leq h^*(n)$$

Construction An admissible heuristic can be derived from a relaxed version of the problem, or by information from pattern databases that store exact solutions to subproblems of the problem

Consistent heuristic

In the study of path-finding problems in artificial intelligence, a heuristic function is said to be **consistent**, or **monotone**, if its estimate is always less than or equal to the estimated distance from any neighbouring vertex to the goal, plus the cost of reaching that neighbour.

Formally, for every node *N* and each successor *P* of *N*, the estimated cost of reaching the goal from *N* is no greater than the step cost of getting to *P* plus the estimated cost of reaching the goal from *P*.

$h(N) \leq c(N,P) + h(P)$ and
$h(G) = 0.$

where

- *h* is the consistent heuristic function
- *N* is any node in the graph
- *P* is any descendant of *N*
- *G* is any goal node
- c(N,P) is the cost of reaching node P from N