Heuristic Search

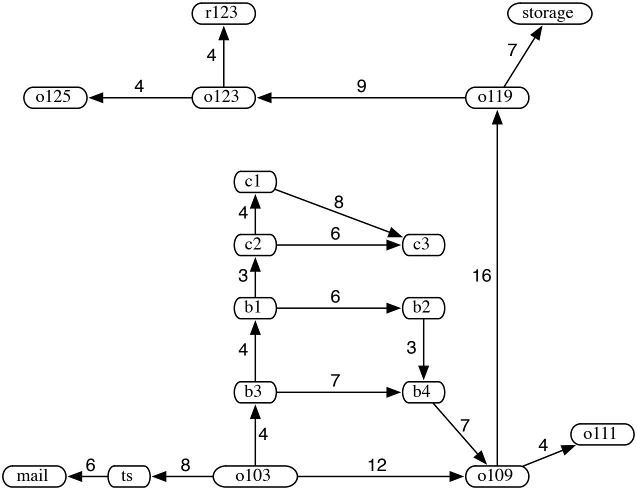
- Idea: don't ignore the goal when selecting paths.
- Often there is extra knowledge that can be used to guide the search: heuristics.
- $h(n)$ is an estimate of the cost of the shortest path from node n to a goal node.
- $h(n)$ uses only readily obtainable information (that is easy to compute) about a node.
- h can be extended to paths: $h(\langle n_0, \ldots, n_k \rangle) = h(n_k)$.
- The function $h(n)$ is an *underestimate* if $h(n)$ is less than or equal to the actual cost of a lowest-cost path from node n to a goal.

The heuristic function informs the search about the direction to a goal.
It provides an informed way to guess which neighbor leads to a goal.

Example Heuristic Functions

- If the nodes are points on a Euclidean plane and the cost is the distance, we can use the straight-line distance from n to the closest goal as the value of $h(n)$.
- If the nodes are locations and cost is time, we can use the distance to a goal divided by the maximum speed.
- If the goal is to collect all of the coins and not run out of fuel, the cost is an estimate of how many steps it will take to collect the rest of the coins, refuel when necessary, and return to goal position.

The Delivery Robot search space



There are three paths from o103 to r123.

Heuristics for the Delivery Robot

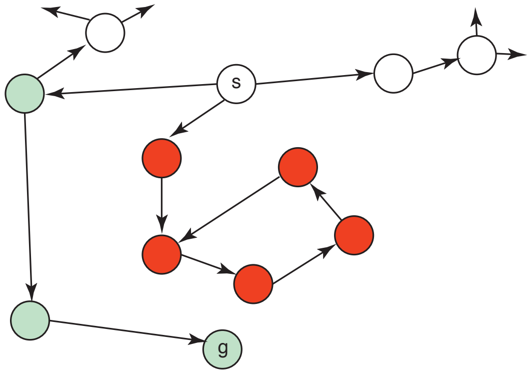
The straight-line distance in the world between the node and the goal position can be used as the heuristic function.
The examples that follow assume the following heuristic function:

$$
\begin{array}{rclrclrcl}
h(\text{mail}) & = & 26 & h(\text{ts}) & = & 23 & h(\text{o103}) & = & 21 \\
h(\text{o109}) & = & 24 & h(\text{o111}) & = & 27 & h(\text{o119}) & = & 11 \\
h(\text{o123}) & = & 4 & h(\text{o125}) & = & 6 & h(\text{r123}) & = & 0 \\
h(\text{b1}) & = & 13 & h(\text{b2}) & = & 15 & h(\text{b3}) & = & 17 \\
h(\text{b4}) & = & 18 & h(\text{c1}) & = & 6 & h(\text{c2}) & = & 10 \\
h(\text{c3}) & = & 12 & h(\text{storage}) & = & 12 & & &
\end{array}
$$

Best-first Search

- Idea: select the path whose end is closest to a goal according to the heuristic function.
- Best-first search selects a path on the frontier with minimal h-value.
- It treats the frontier as a priority queue ordered by h.

Illustrative Graph — Best-first Search



The cost of an arc is its length.
The aim is to find the shortest path from s to g.
Heuristic function: the Euclidean distance to the goal g.
Because all of the nodes below s look good, a best-first search will cycle between them, never trying an alternate route from s.

Complexity of Best-first Search

- It uses space exponential in path length.
- It isn't guaranteed to find a solution, even if one exists.
- It doesn't always find the shortest path.

Heuristic Depth-first Search

- It's a way to use heuristic knowledge in depth-first search.
- Idea: order the neighbors of a node (by h) before adding them to the front of the frontier.
- It locally selects which subtree to develop, but still does depth-first search. It explores all paths from the node at the head of the frontier before exploring paths from the next node.
- Space is linear in path length. It isn't guaranteed to find a solution.

A^* Search

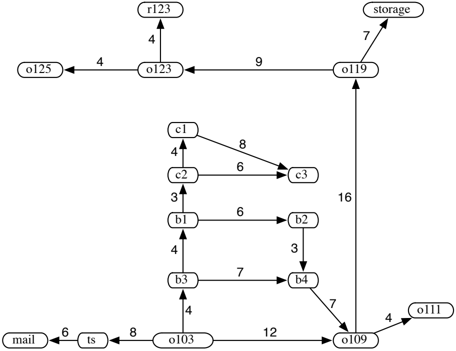
- A^* search uses both path cost and heuristic values
- $cost(p)$ is the cost of path p.
- $h(p)$ estimates the cost from the end of p to a goal.
- Let $f(p) = cost(p) + h(p)$. $f(p)$ estimates the total path cost of going from a start node to a goal via p.

$$\underbrace{\underbrace{start \xrightarrow{\text{path } p} n}_{cost(p)} \underbrace{\xrightarrow{\text{estimate}} goal}_{h(p)}}_{f(p)}$$

A^* Search Algorithm

- A^* is a mix of lowest-cost-first and best-first search.
- It treats the frontier as a priority queue ordered by $f(p)$.
- It always selects the node on the frontier with the lowest estimated distance from the start to a goal node constrained to go via that node.

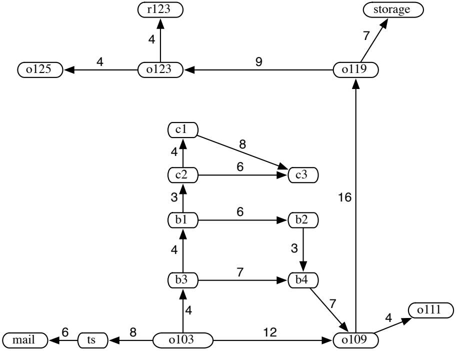
A^* for the Delivery Robot



$h(\text{mail})=26$	$h(\text{ts})=23$	$h(\text{o103})=21$
$h(\text{o109})=24$	$h(\text{o111})=27$	$h(\text{o119})=11$
$h(\text{o123})=4$	$h(\text{o125})=6$	$h(\text{r123})=0$
$h(\text{b1})=13$	$h(\text{b2})=15$	$h(\text{b3})=17$
$h(\text{b4})=18$	$h(\text{c1})=6$	$h(\text{c2})=10$
$h(\text{c3})=12$	$h(\text{storage})=12$	

Goal: from o103 to r123

$[o103_{21}]$,

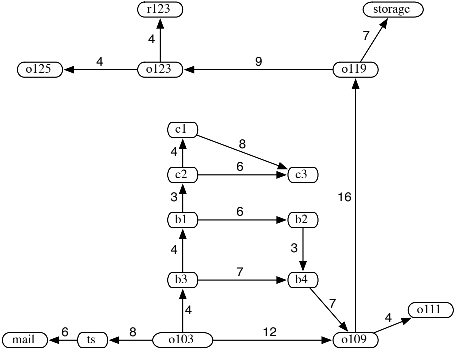


$h(\text{mail})=26$	$h(\text{ts})=23$	$h(\text{o103})=21$
$h(\text{o109})=24$	$h(\text{o111})=27$	$h(\text{o119})=11$
$h(\text{o123})=4$	$h(\text{o125})=6$	$h(\text{r123})=0$
$h(\text{b1})=13$	$h(\text{b2})=15$	$h(\text{b3})=17$
$h(\text{b4})=18$	$h(\text{c1})=6$	$h(\text{c2})=10$
$h(\text{c3})=12$	$h(\text{storage})=12$	

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}],$

A^* for the Delivery Robot

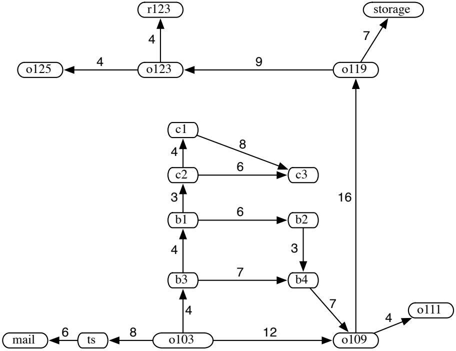


$h(\text{mail})=26$	$h(\text{ts})=23$	$h(\text{o103})=21$
$h(\text{o109})=24$	$h(\text{o111})=27$	$h(\text{o119})=11$
$h(\text{o123})=4$	$h(\text{o125})=6$	$h(\text{r123})=0$
$h(\text{b1})=13$	$h(\text{b2})=15$	$h(\text{b3})=17$
$h(\text{b4})=18$	$h(\text{c1})=6$	$h(\text{c2})=10$
$h(\text{c3})=12$	$h(\text{storage})=12$	

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}],$

A^* for the Delivery Robot

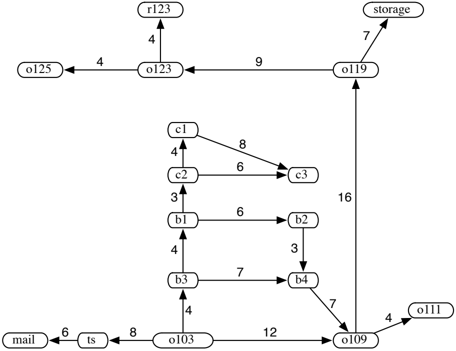


$h(\text{mail})=26$ $h(\text{ts})=23$ $h(\text{o103})=21$
$h(\text{o109})=24$ $h(\text{o111})=27$ $h(\text{o119})=11$
$h(\text{o123})=4$ $h(\text{o125})=6$ $h(\text{r123})=0$
$h(\text{b1})=13$ $h(\text{b2})=15$ $h(\text{b3})=17$
$h(\text{b4})=18$ $h(\text{c1})=6$ $h(\text{c2})=10$
$h(\text{c3})=12$ $h(\text{storage})=12$

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$

A^* for the Delivery Robot

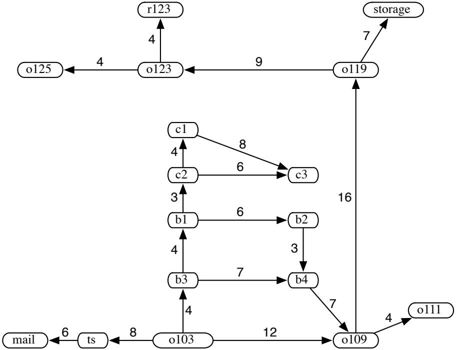


$h(\text{mail})=26$ $h(\text{ts})=23$ $h(\text{o103})=21$
$h(\text{o109})=24$ $h(\text{o111})=27$ $h(\text{o119})=11$
$h(\text{o123})=4$ $h(\text{o125})=6$ $h(\text{r123})=0$
$h(\text{b1})=13$ $h(\text{b2})=15$ $h(\text{b3})=17$
$h(\text{b4})=18$ $h(\text{c1})=6$ $h(\text{c2})=10$
$h(\text{c3})=12$ $h(\text{storage})=12$

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}],$

A^* for the Delivery Robot

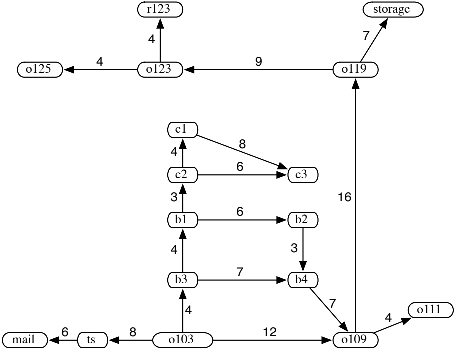


$h(\text{mail})=26$ $h(\text{ts})=23$ $h(\text{o103})=21$
$h(\text{o109})=24$ $h(\text{o111})=27$ $h(\text{o119})=11$
$h(\text{o123})=4$ $h(\text{o125})=6$ $h(\text{r123})=0$
$h(\text{b1})=13$ $h(\text{b2})=15$ $h(\text{b3})=17$
$h(\text{b4})=18$ $h(\text{c1})=6$ $h(\text{c2})=10$
$h(\text{c3})=12$ $h(\text{storage})=12$

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}], [b4_{29}, b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}],$

A^* for the Delivery Robot

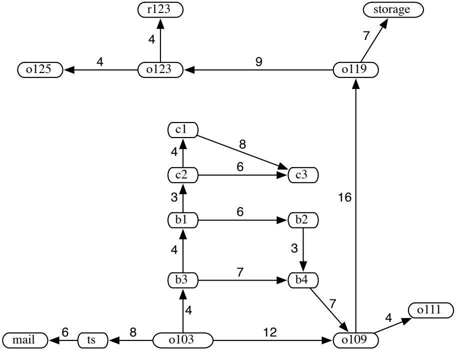


$h(\text{mail})=26$ $h(\text{ts})=23$ $h(\text{o103})=21$
$h(\text{o109})=24$ $h(\text{o111})=27$ $h(\text{o119})=11$
$h(\text{o123})=4$ $h(\text{o125})=6$ $h(\text{r123})=0$
$h(\text{b1})=13$ $h(\text{b2})=15$ $h(\text{b3})=17$
$h(\text{b4})=18$ $h(\text{c1})=6$ $h(\text{c2})=10$
$h(\text{c3})=12$ $h(\text{storage})=12$

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}], [b4_{29}, b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}],$
$[b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}, o109_{42}],$

A^* for the Delivery Robot

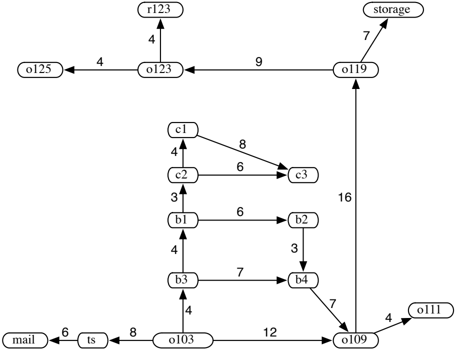


$h(\text{mail})=26$	$h(\text{ts})=23$	$h(\text{o103})=21$
$h(\text{o109})=24$	$h(\text{o111})=27$	$h(\text{o119})=11$
$h(\text{o123})=4$	$h(\text{o125})=6$	$h(\text{r123})=0$
$h(\text{b1})=13$	$h(\text{b2})=15$	$h(\text{b3})=17$
$h(\text{b4})=18$	$h(\text{c1})=6$	$h(\text{c2})=10$
$h(\text{c3})=12$	$h(\text{storage})=12$	

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}], [b4_{29}, b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}],$
$[b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}, o109_{42}], [c3_{29}, ts_{31}, c3_{35}, b4_{35}, o109_{36}, o109_{42}]$

A^* for the Delivery Robot

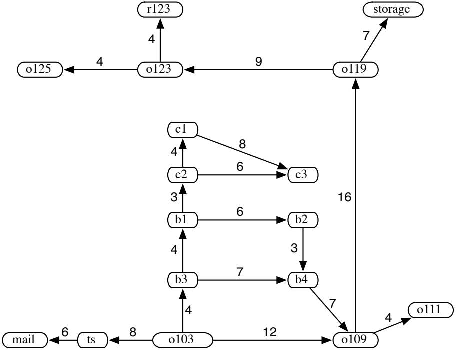


$h(\text{mail})=26$	$h(\text{ts})=23$	$h(\text{o103})=21$
$h(\text{o109})=24$	$h(\text{o111})=27$	$h(\text{o119})=11$
$h(\text{o123})=4$	$h(\text{o125})=6$	$h(\text{r123})=0$
$h(\text{b1})=13$	$h(\text{b2})=15$	$h(\text{b3})=17$
$h(\text{b4})=18$	$h(\text{c1})=6$	$h(\text{c2})=10$
$h(\text{c3})=12$	$h(\text{storage})=12$	

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}], [b4_{29}, b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}],$
$[b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}, o109_{42}], [c3_{29}, ts_{31}, c3_{35}, b4_{35}, o109_{36}, o109_{42}]$
$[ts_{31}, c3_{35}, b4_{35}, o109_{36}, o109_{42}]$

A^* for the Delivery Robot



$h(\text{mail})=26$ $h(\text{ts})=23$ $h(\text{o103})=21$
$h(\text{o109})=24$ $h(\text{o111})=27$ $h(\text{o119})=11$
$h(\text{o123})=4$ $h(\text{o125})=6$ $h(\text{r123})=0$
$h(\text{b1})=13$ $h(\text{b2})=15$ $h(\text{b3})=17$
$h(\text{b4})=18$ $h(\text{c1})=6$ $h(\text{c2})=10$
$h(\text{c3})=12$ $h(\text{storage})=12$

Goal: from o103 to r123

$[o103_{21}], [b3_{21}, ts_{31}, o109_{36}], [b1_{21}, b4_{29}, ts_{31}, o109_{36}], [c2_{21}, b4_{29}, b2_{29}, ts_{31}, o109_{36}],$
$[c1_{21}, b4_{29}, b2_{29}, c3_{29}, ts_{31}, o109_{36}], [b4_{29}, b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}],$
$[b2_{29}, c3_{29}, ts_{31}, c3_{35}, o109_{36}, o109_{42}], [c3_{29}, ts_{31}, c3_{35}, b4_{35}, o109_{36}, o109_{42}]$
$[ts_{31}, c3_{35}, b4_{35}, o109_{36}, o109_{42}]$

A lowest-cost path is eventually found. The algorithm is forced to try many different paths, because several of them temporarily seemed to have the lowest cost. It still does better than either lowest-cost-first search or best-first search.

Admissibility of A^*

If there is a solution, A^* always finds an optimal solution —the first path to a goal selected— if

- the branching factor is finite
- arc costs are bounded above zero (there is some $\epsilon > 0$ such that all of the arc costs are greater than ϵ), and
- $h(n)$ is an underestimate of the length of the shortest path from n to a goal node.

- If $h(n)$ is an underestimate of the path costs from node n to a goal node, then $f(p)$ is an *underestimate* of a path cost of going from a start node to a goal node via p.

Why is A^* admissible?

- If a path p to a goal is selected from a frontier, can there be a shorter path to a goal?
- Suppose path p' is on the frontier. Because p was chosen before p', and $h(p) = 0$:

$$cost(p) \leq cost(p') + h(p').$$

- Because h is an underestimate

$$cost(p') + h(p') \leq cost(p'')$$

  for any path p'' to a goal that extends p'
- So $cost(p) \leq cost(p'')$ for any other path p'' to a goal.

Why is A^* admissible?

- There is always an element of an optimal solution path on the frontier before a goal has been selected. This is because, in the abstract search algorithm, there is the initial part of every path to a goal.
- A^* halts, as the costs of the paths on the frontier keeps increasing, and will eventually exceed any finite number.