How to find it

$A^\star$ $\underbrace{\text{spend as little as possible}}_{\text{min cost}}$, $\underbrace{\text{hoping for the best}}_{h \text{ underestimates}}$

How to live

$A^\star$ $\underbrace{\text{spend as little as possible}}_{\text{min cost}}$, $\underbrace{\text{hoping for the best}}_{h\text{ underestimates}}$

☺ $\underbrace{\text{enjoy the ride}}_{\text{max reward}}$, $\underbrace{\text{mindful of where you're going}}_{\text{apply } h \text{ to arc}}$

How to live

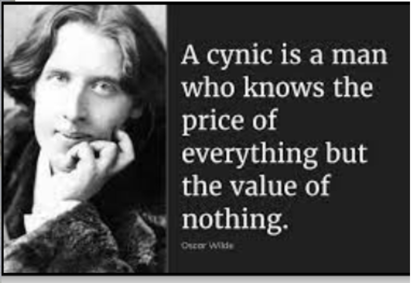
$A^\star$ $\underbrace{\text{spend as little as possible}}_{\text{min cost}}$, $\underbrace{\text{hoping for the best}}_{h\text{ underestimates}}$

☺ $\underbrace{\text{enjoy the ride}}_{\text{max reward}}$, $\underbrace{\text{mindful of where you're going}}_{\text{apply }h\text{ to arc}}$

How to live

$A^\star$ $\underbrace{\text{spend as little as possible}}_{\text{min cost}}$, $\underbrace{\text{hoping for the best}}_{h \text{ underestimates}}$

☺ $\underbrace{\text{enjoy the ride}}_{\text{max reward}}$, $\underbrace{\text{mindful of where you're going}}_{\text{apply } h \text{ to arc}}$

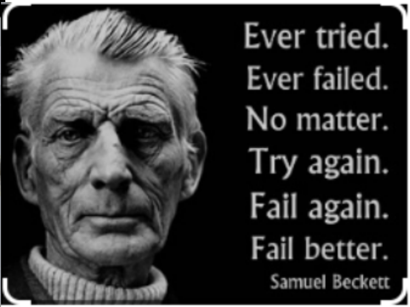

A cynic is a man who knows the price of everything but the value of nothing.

Oscar Wilde

How to live

$A^\star$ $\underbrace{\text{spend as little as possible,}}_{\text{min cost}}$ $\underbrace{\text{hoping for the best}}_{h\text{ underestimates}}$

☺ $\underbrace{\text{enjoy the ride,}}_{\text{max reward}}$ $\underbrace{\text{mindful of where you're going}}_{\text{apply }h\text{ to arc}}$



Ever tried.
Ever failed.
No matter.
Try again.
Fail again.
Fail better.

Samuel Beckett

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

- Arcs have costs that add up in a path

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

- Arcs have costs that add up in a path
- For arc-cost = 1, minimize sum $\rightsquigarrow$ breadth-first

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

- Arcs have costs that add up in a path
- For arc-cost = 1, minimize sum $\rightsquigarrow$ breadth-first
  maximize sum $\rightsquigarrow$ depth-first
- Minimize costs through costly exhaustive search:

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

- Arcs have costs that add up in a path
- For arc-cost $= 1$, minimize sum $\rightsquigarrow$ breadth-first
  maximize sum $\rightsquigarrow$ depth-first
- Minimize costs through costly exhaustive search:
  a tree with branching factor b has

$$1 + b + b^2 + b^3 + \cdots + b^n = \sum_{k=0}^{n} b^k = \frac{b^{n+1} - 1}{b - 1}$$

since $s_n + b^{n+1} = 1 + bs_n$

$$= 2^{n+1} - 1 \text{ for } b = 2$$

nodes of depth $\leq n$.

Arcs and goals at minimal costs

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

- Arcs have costs that add up in a path
- For arc-cost = 1, minimize sum $\rightsquigarrow$ breadth-first
  maximize sum $\rightsquigarrow$ depth-first
- Minimize costs through costly exhaustive search:
  a tree with branching factor b has

$$1 + b + b^2 + b^3 + \cdots + b^n \;=\; \sum_{k=0}^{n} b^k \;=\; \frac{b^{n+1}-1}{b-1}$$

since $s_n + b^{n+1} = 1 + bs_n$

$$= 2^{n+1} - 1 \text{ for } b = 2$$

  nodes of depth $\leq n$.
- FSM `accept` is depth-first:

  n arcs from [q0,[$a_1 \cdots a_n$]] to [q,[]] for final q.

  Prolog also searches depth-first (for speed).

Estimating the cost of a path to a goal

$A^\star$: choose `Node` in frontier, minimizing

$$\text{cost}(\texttt{Start}\ldots\texttt{Node}) + h(\texttt{Node})$$

Estimating the cost of a path to a goal

$A^\star$: choose `Node` in frontier, minimizing

$$\text{cost}(\texttt{Start}\ldots\texttt{Node}) + h(\texttt{Node})$$

$A^\star$ tweaks breadth-first (searching exhaustively) assuming

(1) arc-costs > 0

(2) h under-estimates the cost of a path to a goal

and finite branching.

Estimating the cost of a path to a goal

$A^\star$: choose `Node` in frontier, minimizing

$$\text{cost}(\texttt{Start}\ldots\texttt{Node}) + h(\texttt{Node})$$

$A^\star$ tweaks breadth-first (searching exhaustively) assuming

(1) arc-costs > 0

(2) h under-estimates the cost of a path to a goal

and finite branching.

Min-cost becomes depth-first if arc-cost $= -1$ and $h = 0$

Estimating the cost of a path to a goal

$A^\star$: choose `Node` in frontier, minimizing

$$\text{cost}(\texttt{Start}\ldots\texttt{Node}) + h(\texttt{Node})$$

$A^\star$ tweaks breadth-first (searching exhaustively) assuming

(1) arc-costs > 0

(2) h under-estimates the cost of a path to a goal

and finite branching.

Min-cost becomes depth-first if arc-cost $= -1$ and $h = 0$, violating 2 of 3 conditions sufficient (together) for admissibility:

- for some fixed $\delta > 0$, arc-costs $> \delta$

Estimating the cost of a path to a goal

$A^\star$: choose `Node` in frontier, minimizing

$$\text{cost}(\texttt{Start}\ldots\texttt{Node}) + h(\texttt{Node})$$

$A^\star$ tweaks breadth-first (searching exhaustively) assuming

(1) arc-costs > 0

(2) h under-estimates the cost of a path to a goal

and finite branching.

Min-cost becomes depth-first if arc-cost $= -1$ and $h = 0$, violating 2 of 3 conditions sufficient (together) for admissibility:

- for some fixed $\delta > 0$, arc-costs $> \delta$
- h under-estimates cost of path to goal.

Rewarding exploration

Life is too short for timid, cost-driven search

Rewarding exploration

Life is too short for timid, cost-driven search

SHIFTING PERSPECTIVES:

- costly arc $\rightsquigarrow$ rewarding move

Rewarding exploration

Life is too short for timid, cost-driven search

SHIFTING PERSPECTIVES:

- costly arc $\rightsquigarrow$ rewarding move
- minimize costs $\rightsquigarrow$ maximize reward
  for reward $= -$cost,

$$\text{min cost} \;\approx\; \text{max reward}$$

Rewarding exploration

Life is too short for timid, cost-driven search

Shifting perspectives:

- costly arc $\leadsto$ rewarding move
- minimize costs $\leadsto$ maximize reward
  for reward $= -$cost,
  $$\text{min cost} \approx \text{max reward}$$
- let goal affect arc reward (goal-directed search),
  mixing destination (goal) with journey (arc)

Rewarding exploration

Life is too short for timid, cost-driven search

SHIFTING PERSPECTIVES:

- costly arc $\rightsquigarrow$ rewarding move
- minimize costs $\rightsquigarrow$ maximize reward
  for reward $= -$cost,
  $$\text{min cost} \approx \text{max reward}$$
- let goal affect arc reward (goal-directed search),
  mixing destination (goal) with journey (arc)
- frontier search from start $\rightsquigarrow$ back up from goal
  from branching factor $b = 2$ to future discount $b = \frac{1}{2}$

Rewarding exploration

Life is too short for timid, cost-driven search

Shifting perspectives:

- costly arc $\rightsquigarrow$ rewarding move
- minimize costs $\rightsquigarrow$ maximize reward
  for reward $= -$cost,
  $$\text{min cost} \approx \text{max reward}$$
- let goal affect arc reward (goal-directed search),
  mixing destination (goal) with journey (arc)
- frontier search from start $\rightsquigarrow$ back up from goal
  from branching factor $b = 2$ to future discount $b = \frac{1}{2}$
  approximate reward
  $$H = \lim_{n \to \infty} H_n$$
  by looking n steps ahead H_n
  — learning by incrementing n