Advance Discrete Planning (1) A* variants

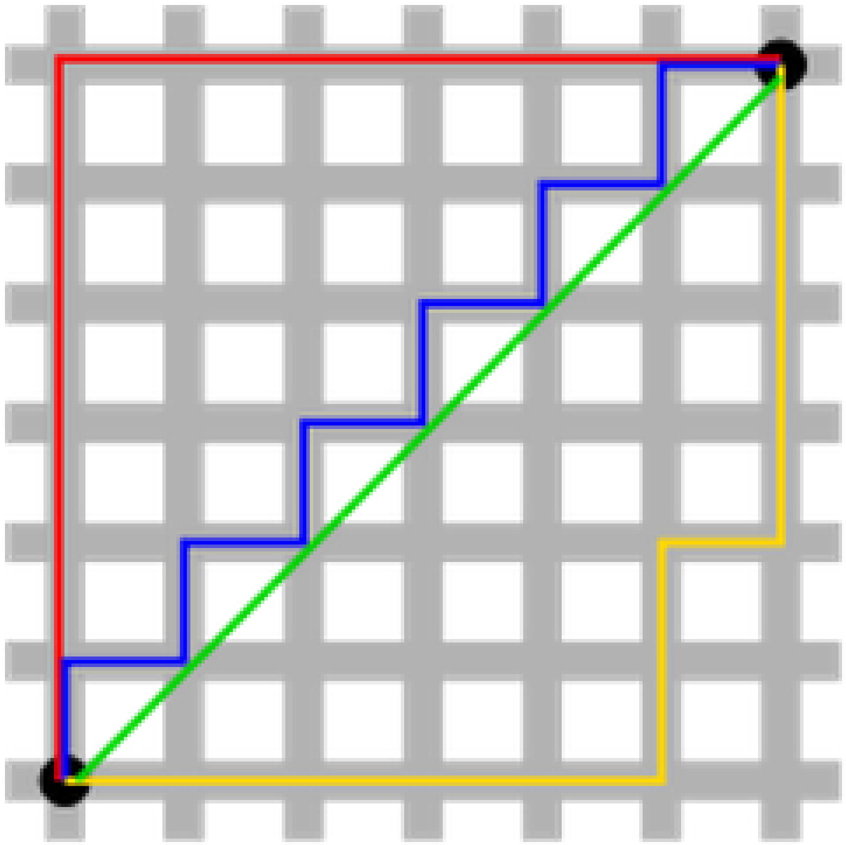
Jane Li

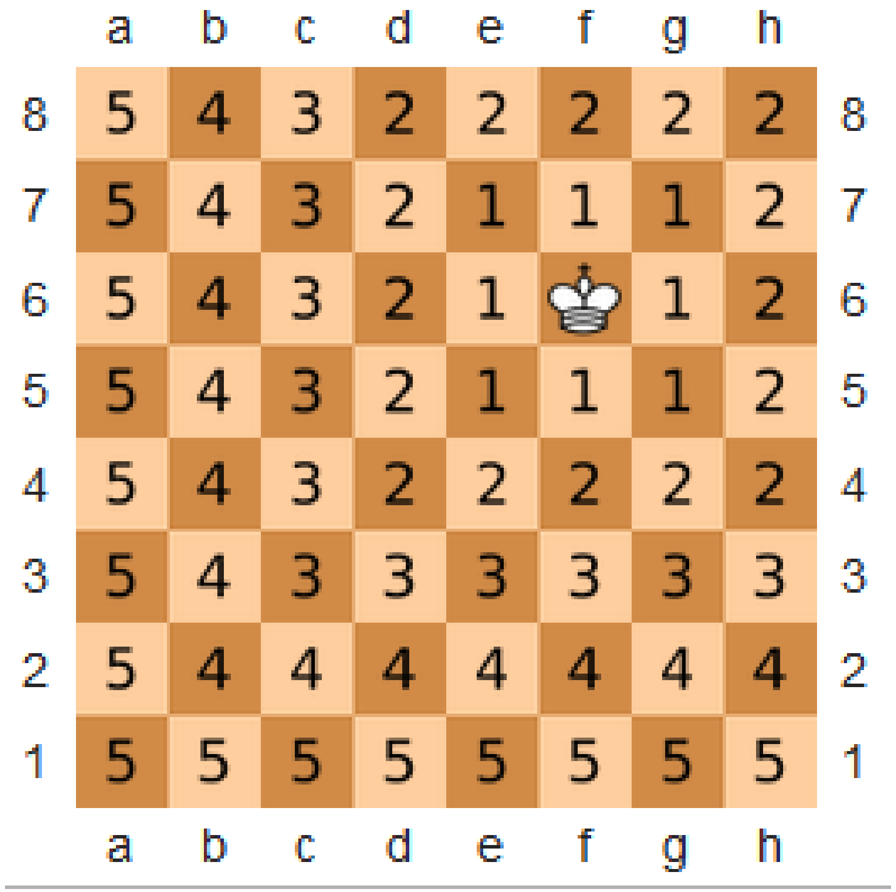
Assistant Professor
Mechanical Engineering Department, Robotic Engineering Program
Worcester Polytechnic Institute

Quiz (10 pts)

- (2 pts) How to measure distance in Cspace? List at least two metrics.
- (2 pts) Why do we need anytime search algorithm?
- (2 pts) Describe an example of abstract goal?
- (4 pts) What are the pros and cons of inflating the heuristic of a best-first search algorithm (like A*)?

Distance in C-space





Distance metrics

- L1-norm (Manhattan distance)

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

- L2-norm (Euclidian distance)

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

- L_∞-norm (chessboard distance)

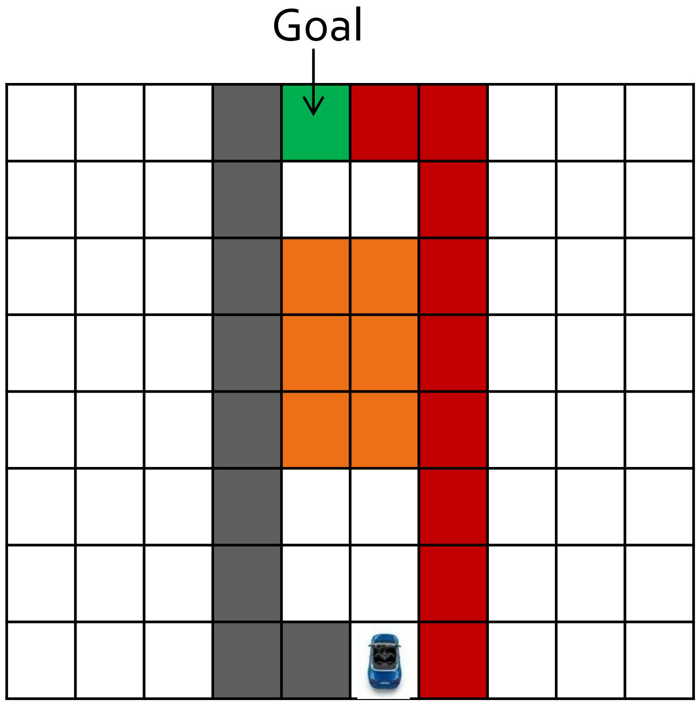
$$D_{\mathrm{Chebyshev}}(p, q) := \max_i(|p_i - q_i|).$$

How to search

- More issues
  - Do you need to search again, and again?
  - What if you search within limited amount of time?
  - What if your search may terminate all of sudden?

Goal Test

- Goals are most commonly specific cells you want to get to
- But they can be more abstract, too!
- Example Goals?
  - A state where X is visible
  - A state where the robot is contacting X
  - Topological goals



Admissibility

- h(x) must *never overestimate* the true cost-to-come
  - $h(x) < h^*(x)$, where $h^*(x)$ is the true cost
  - $h(x) > 0$ (so $h(G) = 0$ for goals G)
- If h(x) is *admissible*, A* will find the least-cost path!
- "Inflating" the heuristic
  - Faster search
  - Least-cost path is not guaranteed

Advanced Discrete Motion Planning

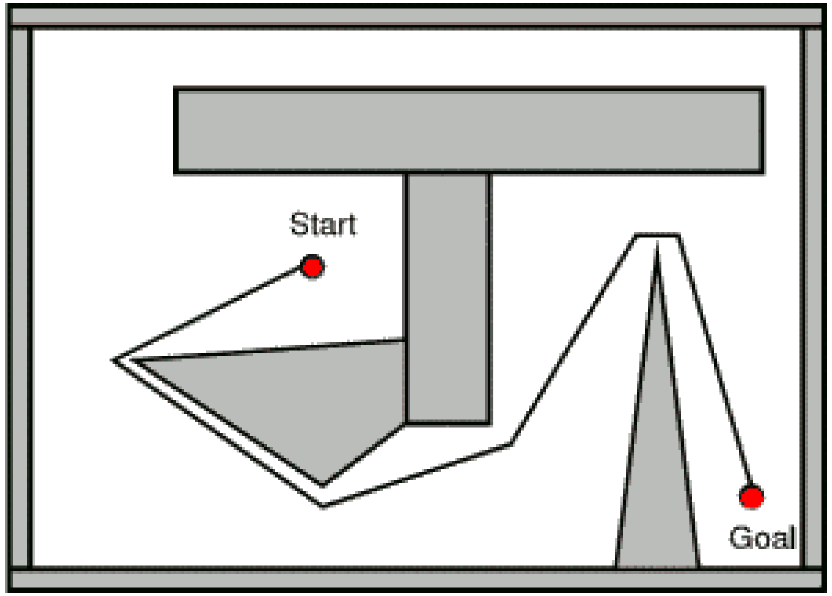
Overview of Advance Discrete Motion Planning

- More on search algorithms
  - Toward optimal and quicker solutions
  - Variants of A*
- Practical issues
  - Case study – Practical search techniques for autonomous driving
- Other advanced topics
  - Roadmaps for planning in dynamic environment
  - Learning search via imitation

A* Variants

Problem Statement

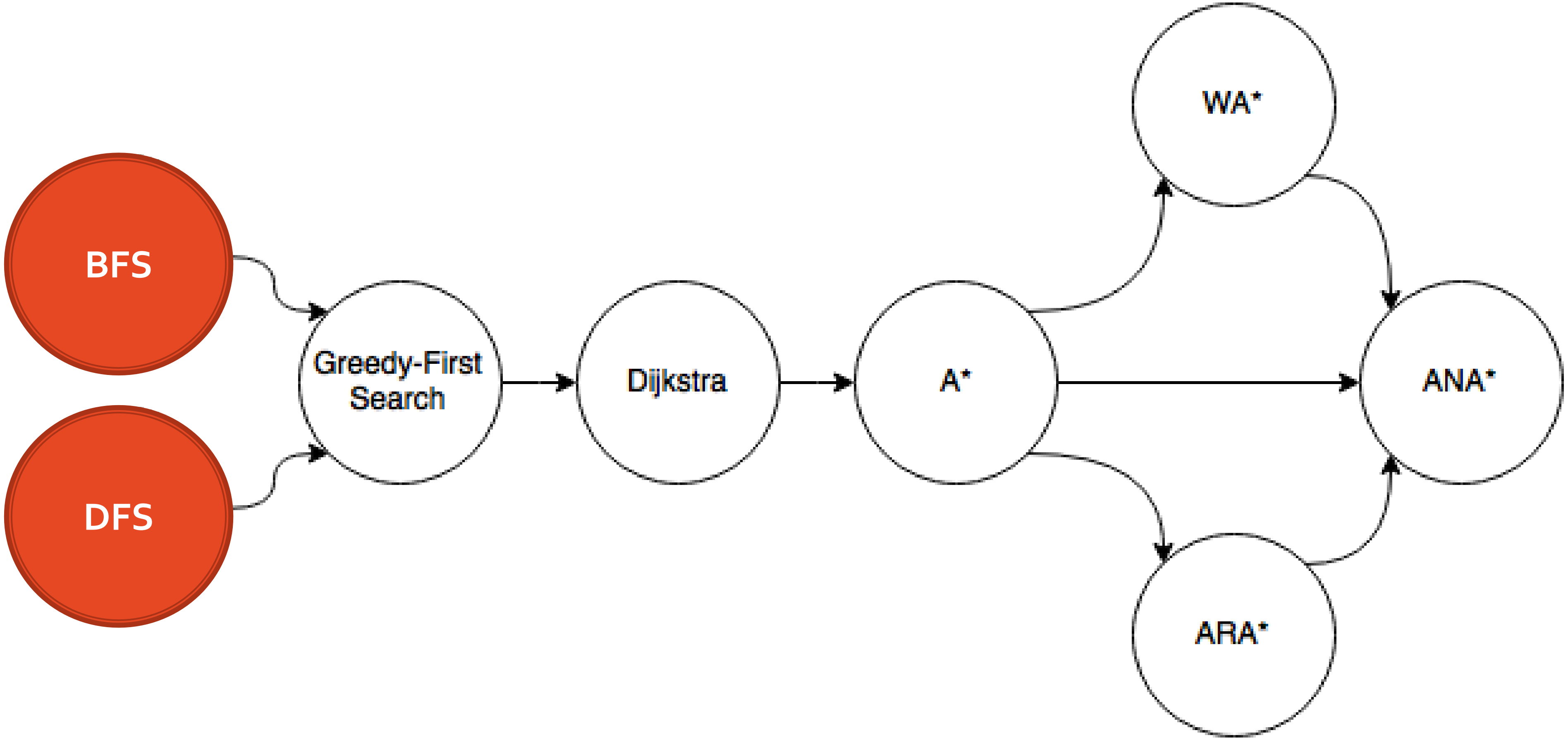
- Given a graph (G), find a path from start state to goal state.
- Parameters:
  - Optimal Path
  - Quicker Solution



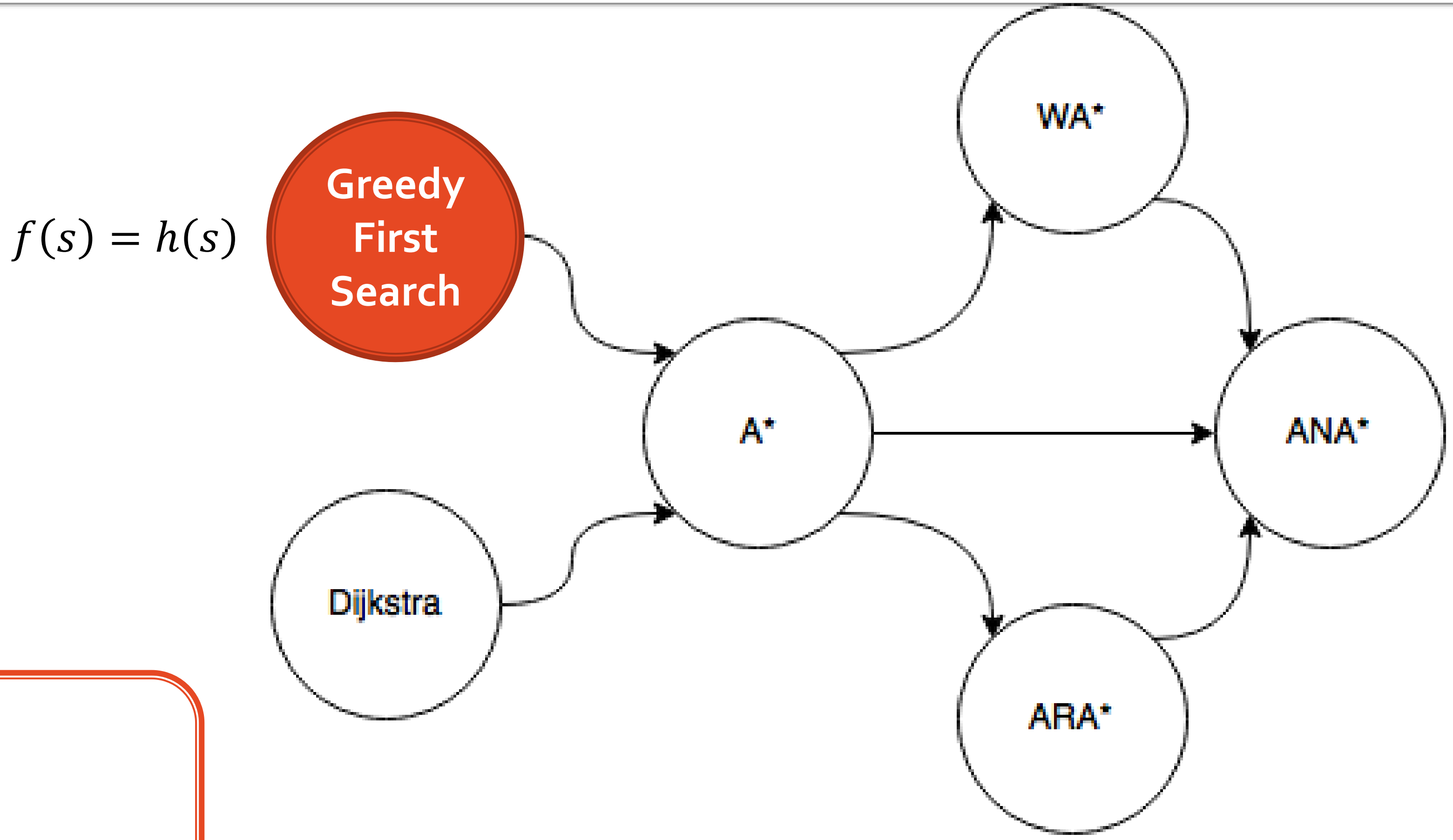
Current state-of-the-arts: ANA*

- Key idea:
  - Quickly find a sub-optimal solution
  - Improve it over time
- Features
  - Finds an initial solution faster
  - Spends less time in improving the solution
  - Improves solution sub-optimality bound in elegant way
  - Converges to optimal solution faster

Evolution of Search Algorithms



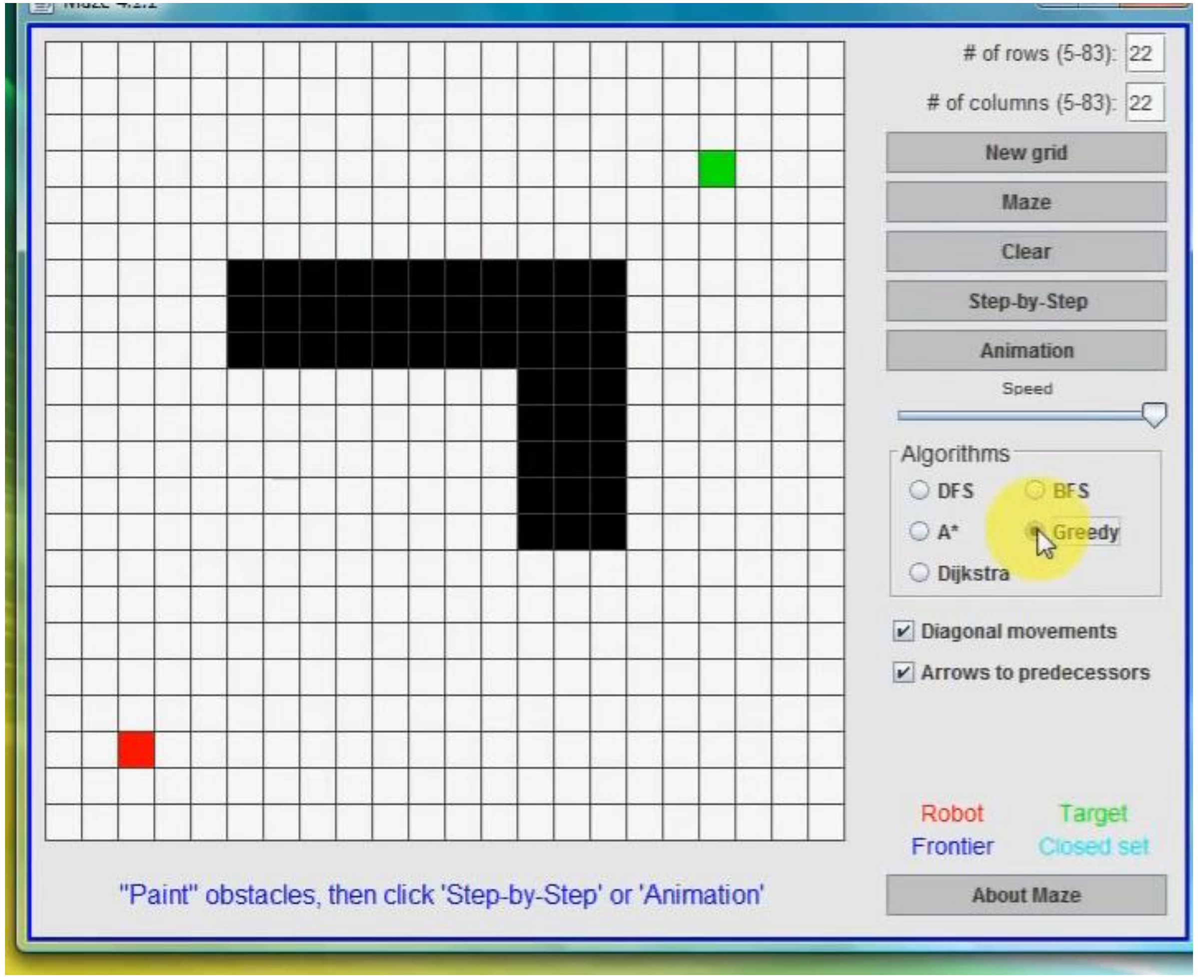
Greedy First



- Goal gets full attention
- Next expanded node has best estimate

Greedy Search Algorithm

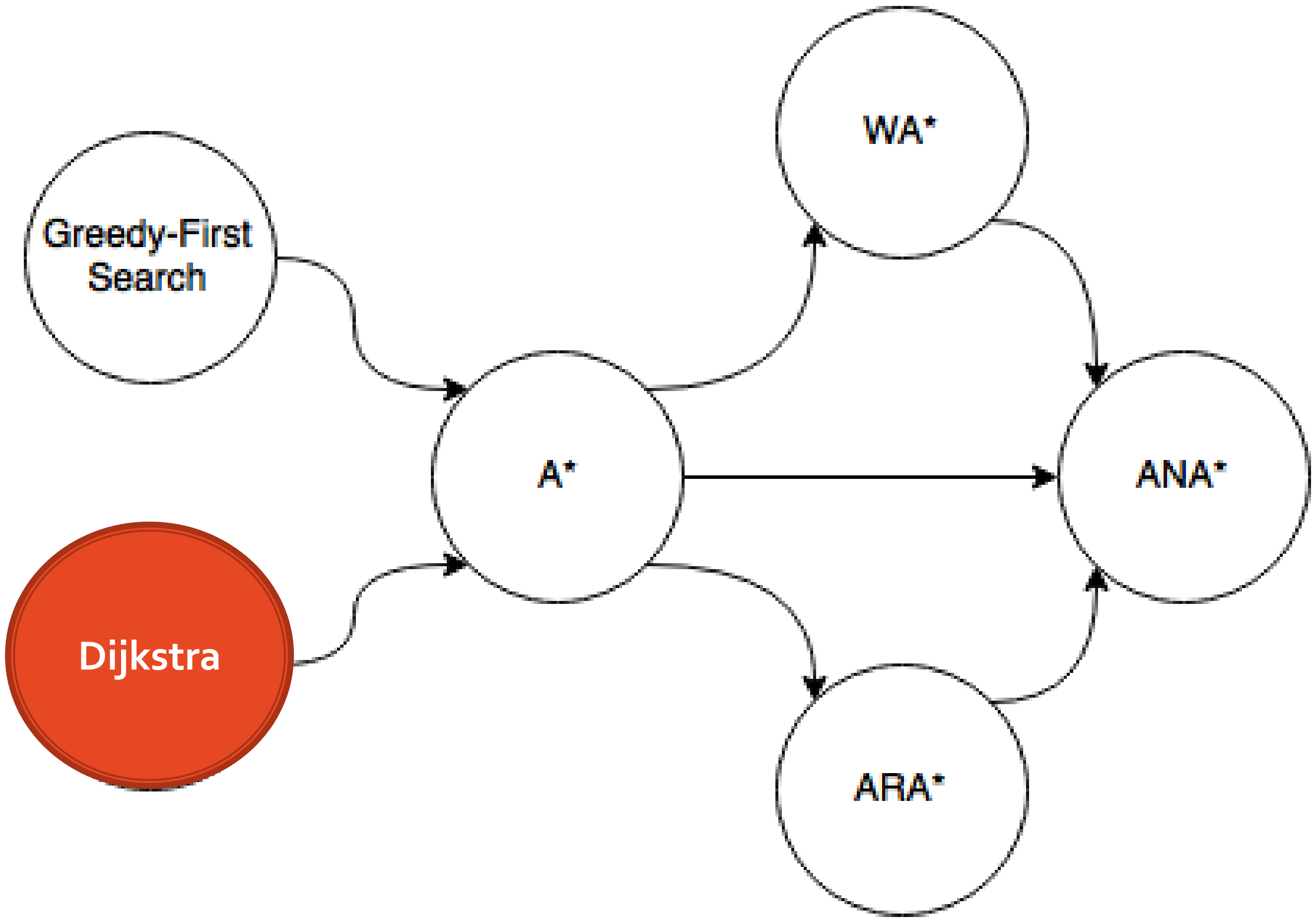
- Cost function
  - $f(s) = h(s)$



Dijkstra

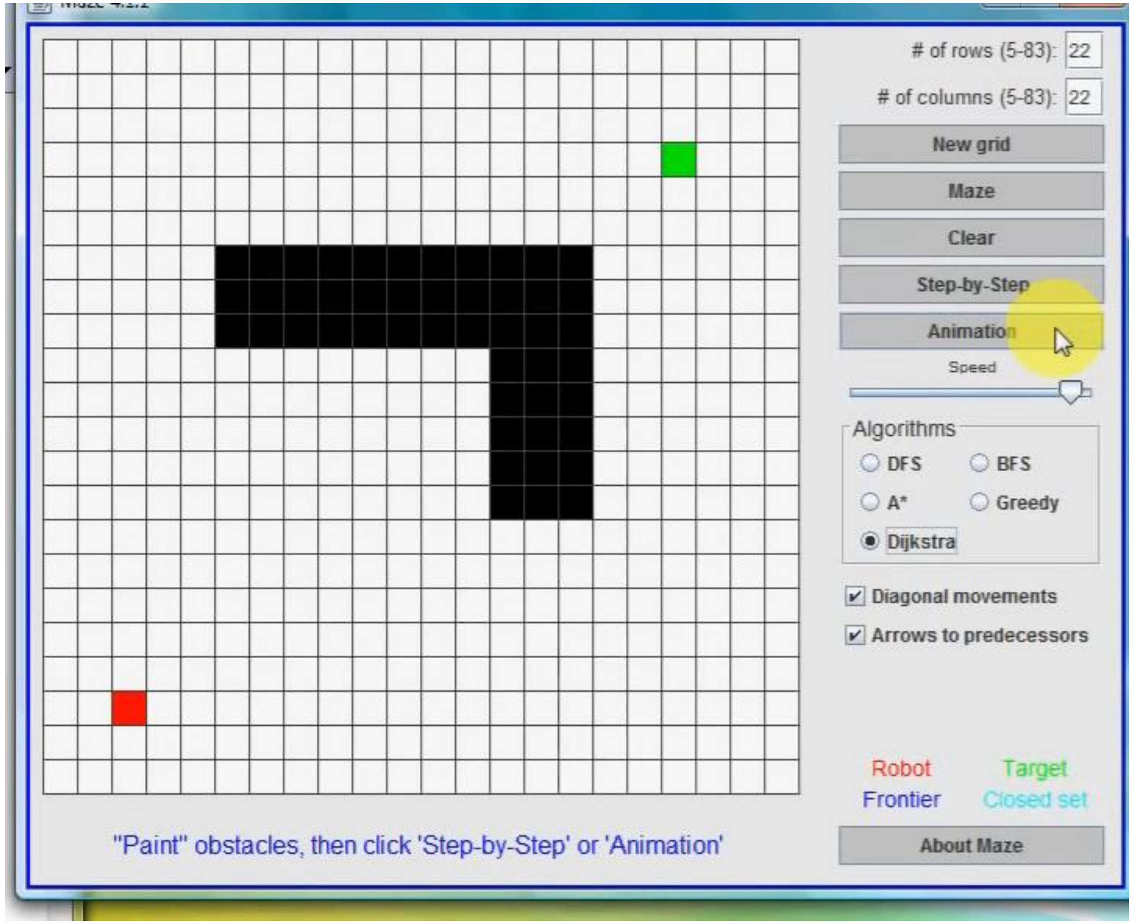
Optimality of Path to Goal

$f(s) = g(s)$



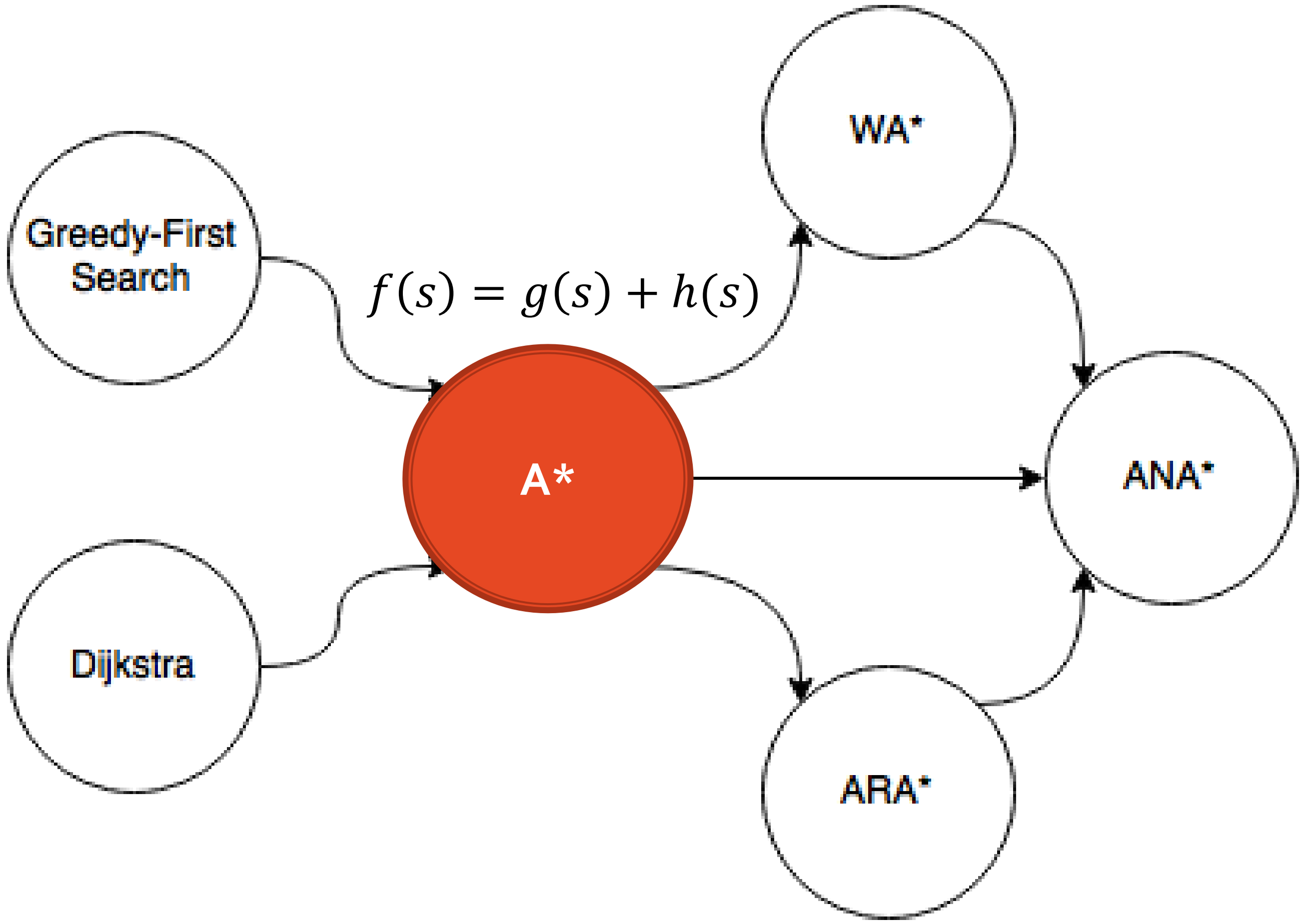
Dijsktra Algorithm

- Cost function
  - $f(s)=g(s)$



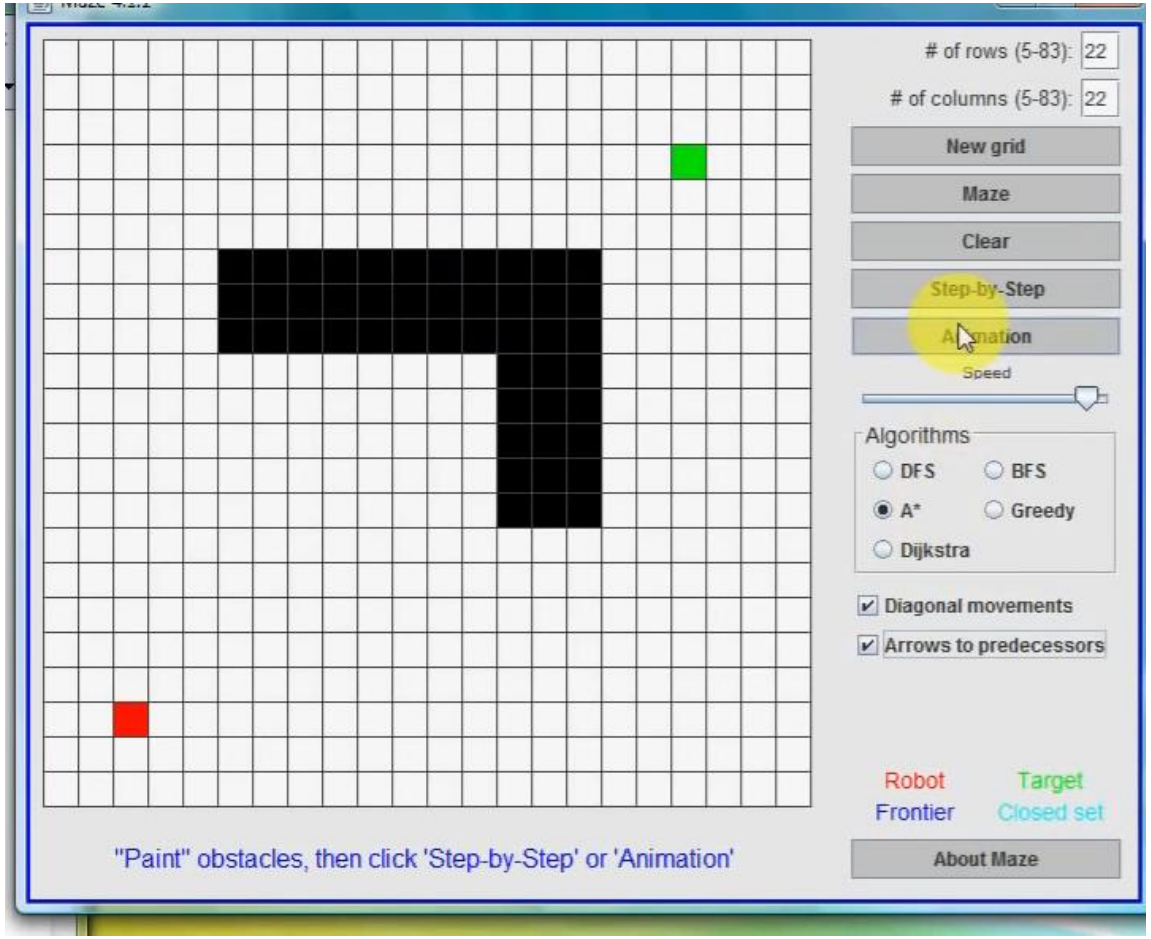
A*

- **Converges Quickly to Optimal Path**
- **Heuristic Functions**

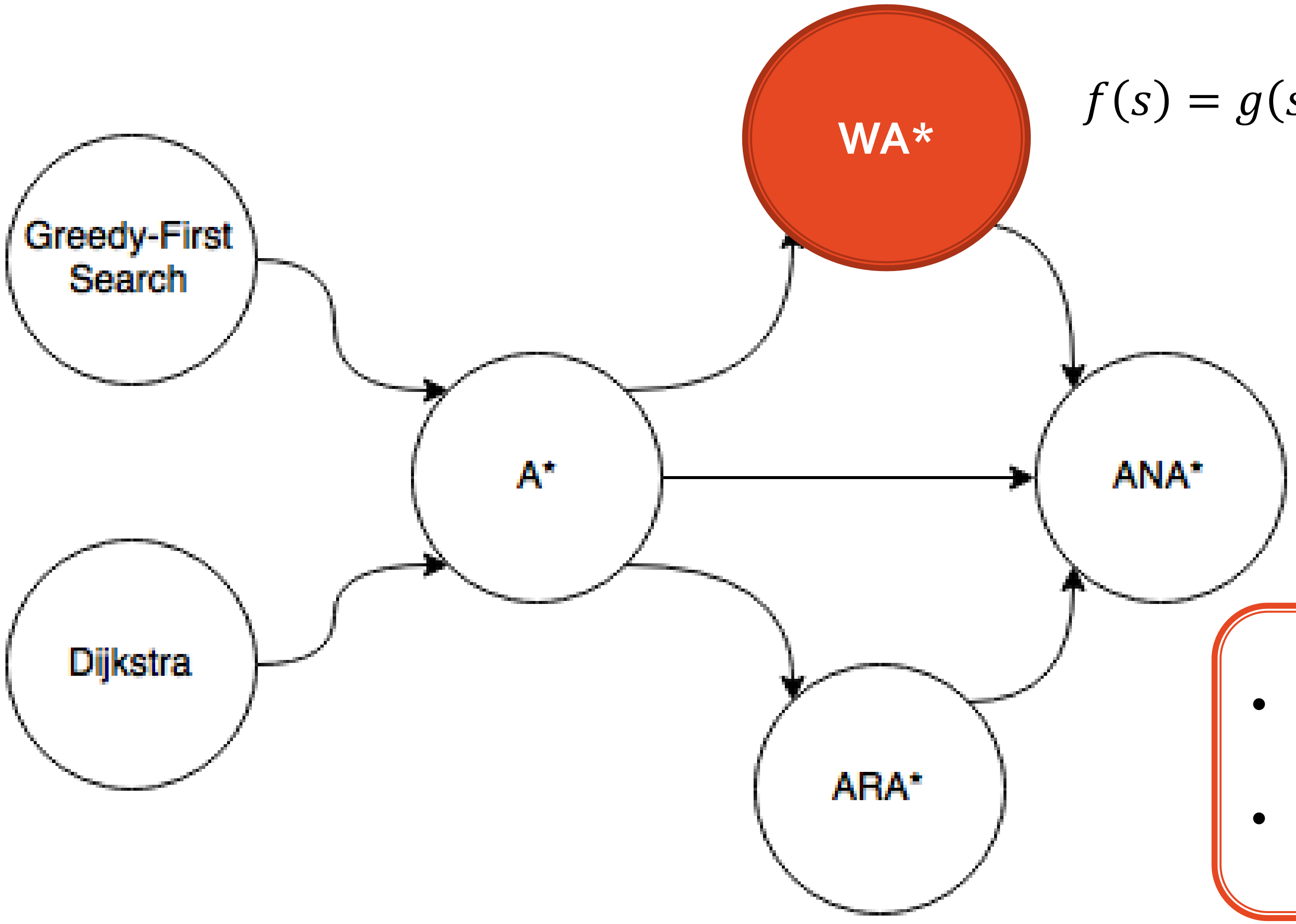


A* Algorithm

- A*: Cost function
  - $f(s)=g(s)+h(s)$
- A* is Optimal, if and only if heuristic is **Admissible**!



Weighted A*



$$f(s) = g(s) + \epsilon \cdot h(s)$$

- **Controlled Speed of Convergence to Path**
- **Cost of Speed is Optimality!**

WA* Algorithm

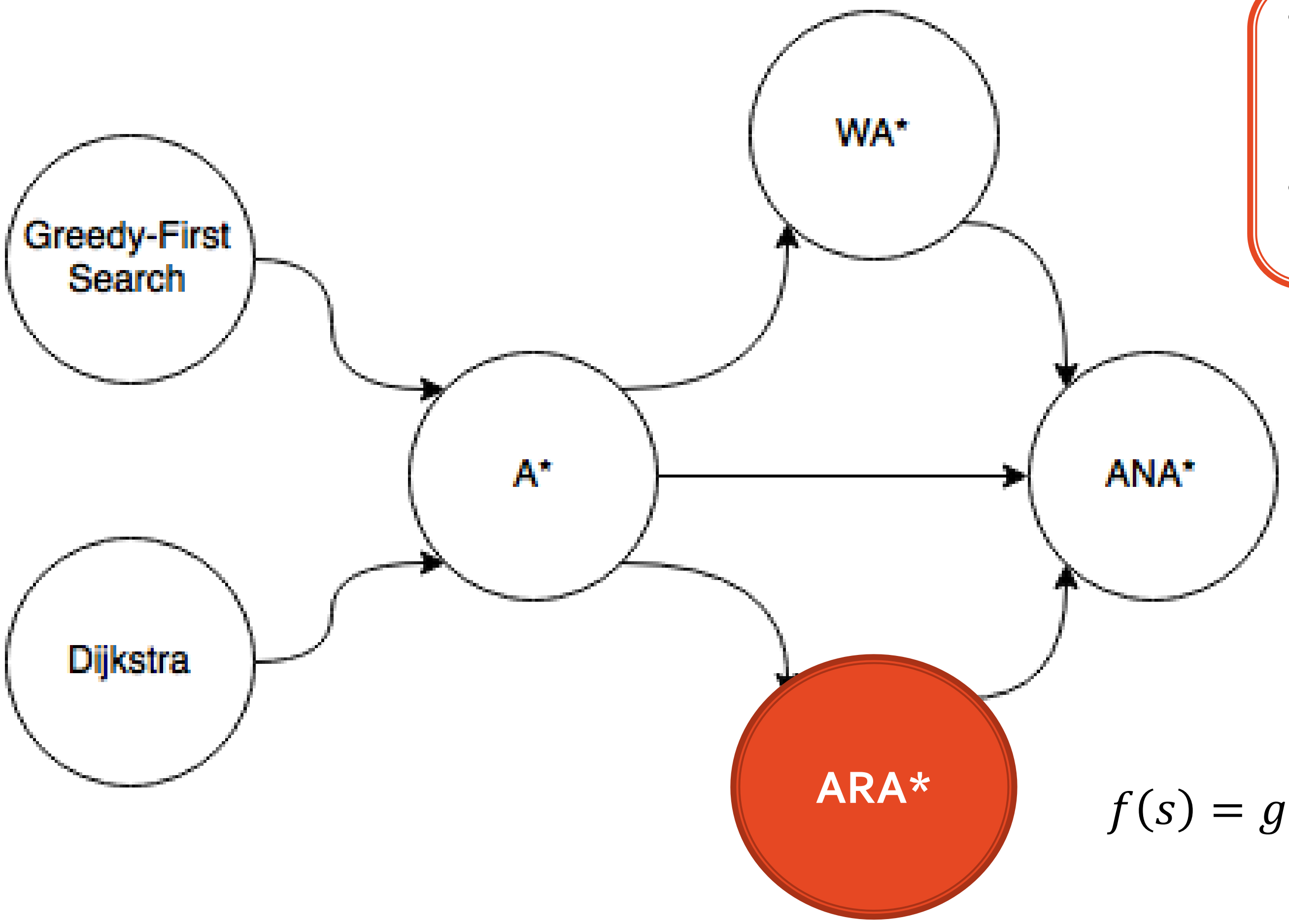
- WA* cost function

$$f(s) = g(s) + \epsilon \cdot h(s)$$

- WA* is optimal, if and only if $\epsilon \cdot h(s)$ is Admissible.
  - $\epsilon > 1$ may lead to sub-optimal, but faster solution

- Note: We have a parameter to set ϵ!!

ARA*



- Issue WA*: Terminates with Sub-Optimal Solution
- ARA*: Iteratively reduces ϵ till Optimal Solution

$$f(s) = g(s) + \epsilon \cdot h(s)$$

ARA* Algorithm

- ARA*

$$f(s) = g(s) + \epsilon \cdot h(s)$$
$$\epsilon_{new} = \epsilon_{old} - \Delta\epsilon$$

- Benefit
  - Can return anytime
  - No need to wait for Optimal Solution!
- Guarantee
  - ARA* Eventually Converges to Optimal Solution

Pseudo-Code Notation

- Start state = s_{start}
- Goal state = s_{goal}
- For each state ***s***,
  - g(s) = minimal cost from start to **s** $g(s_{start}) = 0,\ g(s) = \infty$
  - h(s) = heuristics
- Global variable **G** = cost of the current best solution

Pseudo-Code Notation

- ***OPEN*** queue = the open list
- Initially, ***OPEN*** only contains s_{start}
- Expend the node in ***OPEN*** with the minimal [something]

ARA* Algorithm

ARA*$(\varepsilon_0, \Delta\varepsilon)$

$G \leftarrow \infty$; $\varepsilon \leftarrow \varepsilon_0$; $OPEN \leftarrow \emptyset$; $\forall s : g(s) \leftarrow \infty$; $g(s_{\text{start}}) \leftarrow 0$
Insert s_{start} into $OPEN$ with key $f(s_{\text{start}})$
while $OPEN \neq \emptyset$ **do**
  IMPROVESOLUTION()
  Report current ε-suboptimal solution
  $\varepsilon \leftarrow \varepsilon - \Delta\varepsilon$
  Update keys $f(s)$ in $OPEN$ and prune if $g(s) + h(s) \geq G$

IMPROVESOLUTION()

while $OPEN \neq \emptyset$ **and** $\min_{s \in OPEN}\{f(s)\} \leq G$ **do**
  $s \leftarrow \arg\min_{s \in OPEN}\{f(s)\}$
  $OPEN \leftarrow OPEN \setminus \{s\}$
  if ISGOAL(s) **then**
    $G \leftarrow g(s)$
    return
  for each successor s' of s **do**
    if $g(s) + c(s, s') < g(s')$ **then**
      $g(s') \leftarrow g(s) + c(s, s')$
      pred$(s') \leftarrow s$
      if $g(s') + h(s') < G$ **then**
        Insert or update s' in $OPEN$ with key $f(s')$

Expand node s with best f(s)

Update cost of successors

- Note: We have another parameter to set $\Delta\epsilon$!!

Pruning

- Remove the node, if

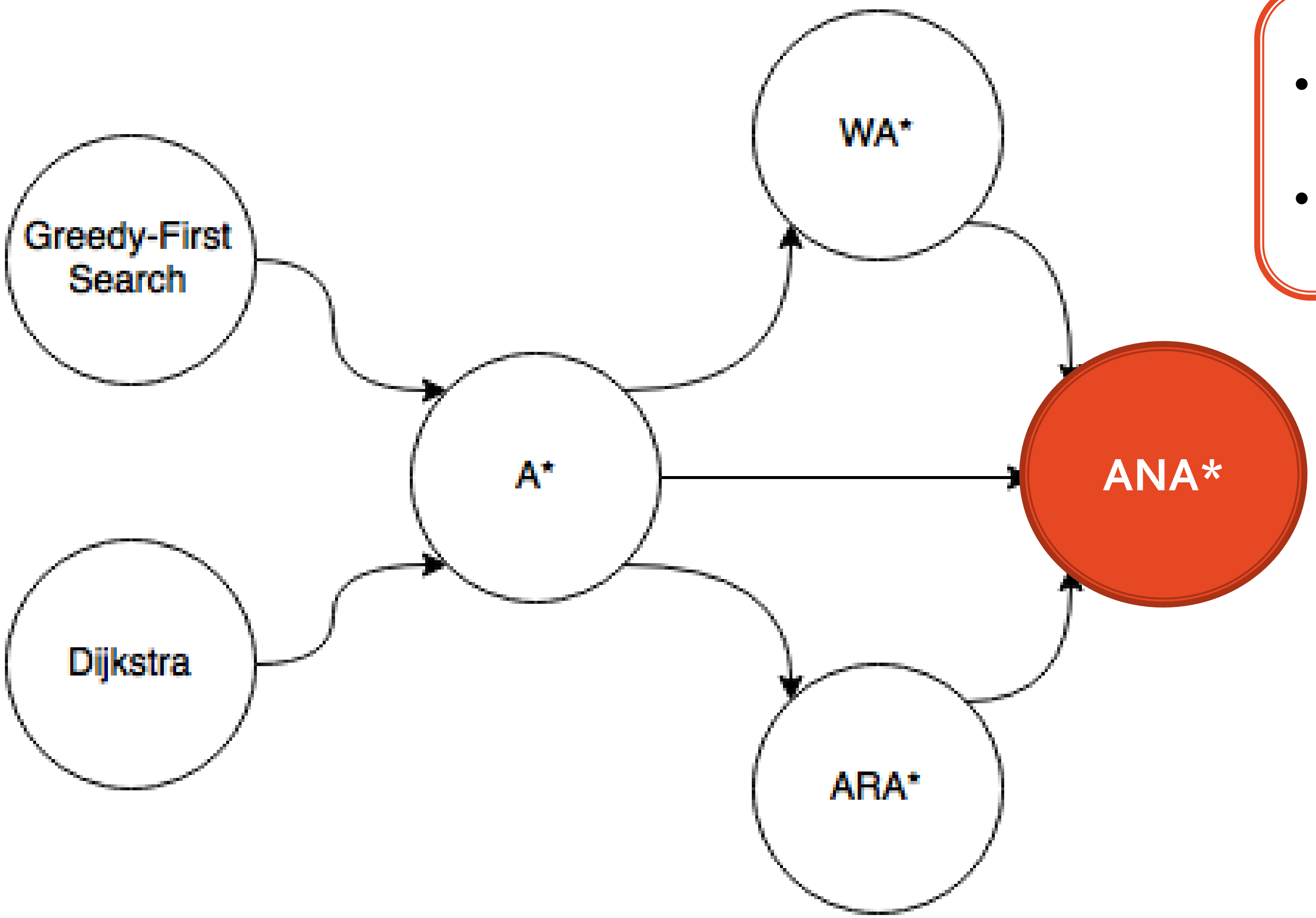
$$g(s) + h(s) > G$$

  - This node can never be a part of optimal path!

Ad-hoc Parameters!!

- We must select 2 parameters - ϵ, $\Delta\epsilon$
- If ϵ is too large and $\Delta\epsilon$ is small enough?
  - Convergence is slow
- Example: ϵ = 1000, $\Delta\epsilon$=1
  - $f(s)=g(s)+1000 \cdot h(s)$
  - $f(s)=g(s)+999 \cdot h(s)$
  - No change in path!
- Path is modified **iff** order relationship between $f(s)$ changes!

ANA*



- **No Parameters**
- **Elegant choice of $\Delta\epsilon$ => Selected On-the-fly**

$$f(s) = g(s) + \epsilon \cdot h(s)$$

ANA*

- Objective 1:
  - Converge quickly to initial solution
- Recall:
  - Greedy-First Search ⇒ Only heuristic counts!

$$f(s) = g(s) + \epsilon \cdot h(s)$$
$$\epsilon \to \infty$$

- This offers a quick initial solution

ANA* Algorithm

- Objective 2
  - Automate selection of $\Delta\epsilon$

- Define:

$$e(s) = \frac{G - g(s)}{h(s)}$$

$$\epsilon_{new} = \max_{\{s \in OPEN\}} e(s)$$

- G: is the best known cost of goal node.

ANA* Algorithm

- What does ϵ_{new} represent?

$$e(s) = \frac{G - g(s)}{h(s)}$$

- Intuitively – We explore the node with following properties
  - It has a large scope of optimality-improvement
  - It has a low heuristic value

ANA* Algorithm

ANA*()

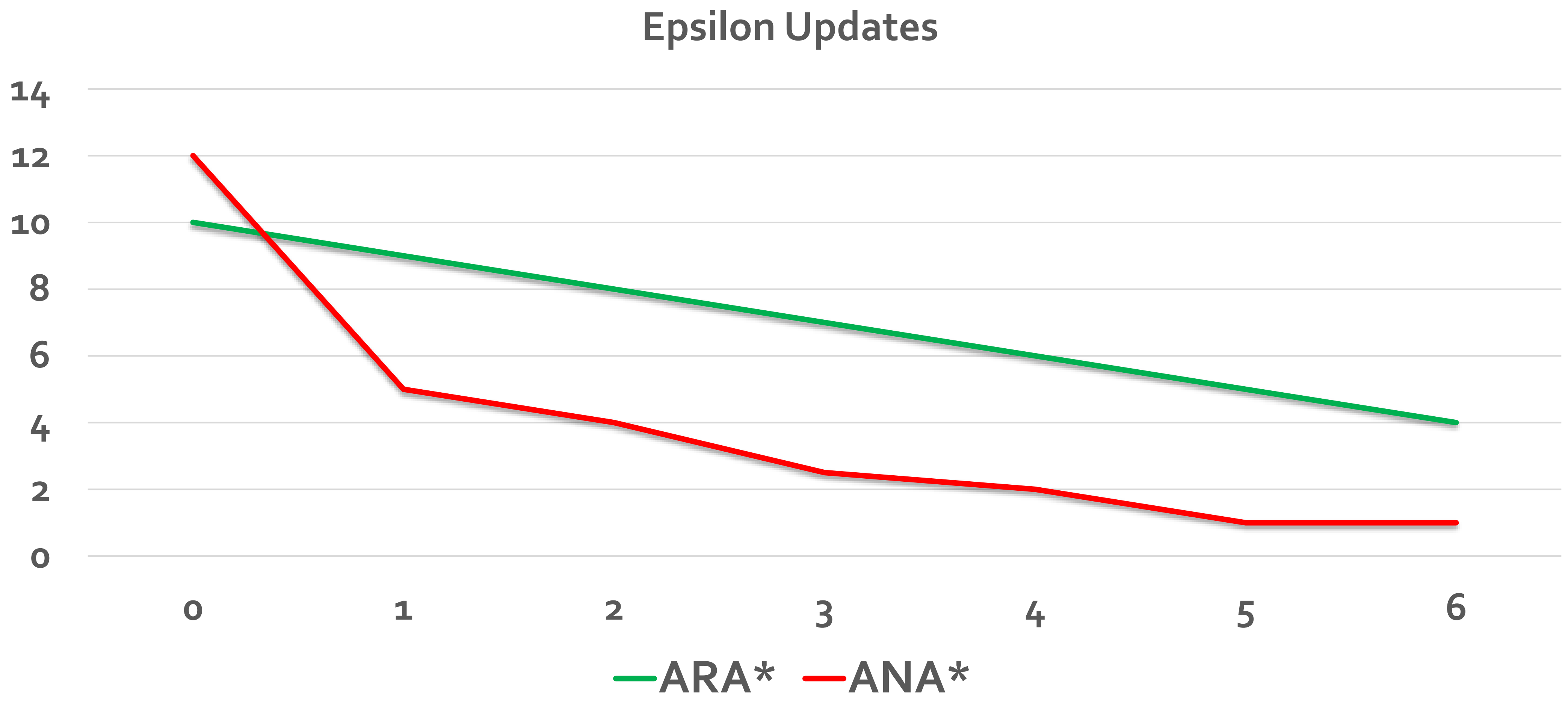
$G \leftarrow \infty$; $E \leftarrow \infty$; $OPEN \leftarrow \emptyset$; $\forall s : g(s) \leftarrow \infty$; $g(s_{start}) \leftarrow 0$
Insert s_{start} into $OPEN$ with key $e(s_{start})$
while $OPEN \neq \emptyset$ **do**
    IMPROVESOLUTION()
    Report current E-suboptimal solution
    Update keys $e(s)$ in $OPEN$ and prune if $g(s) + h(s) \geq G$

Note: No parameters!

IMPROVESOLUTION()

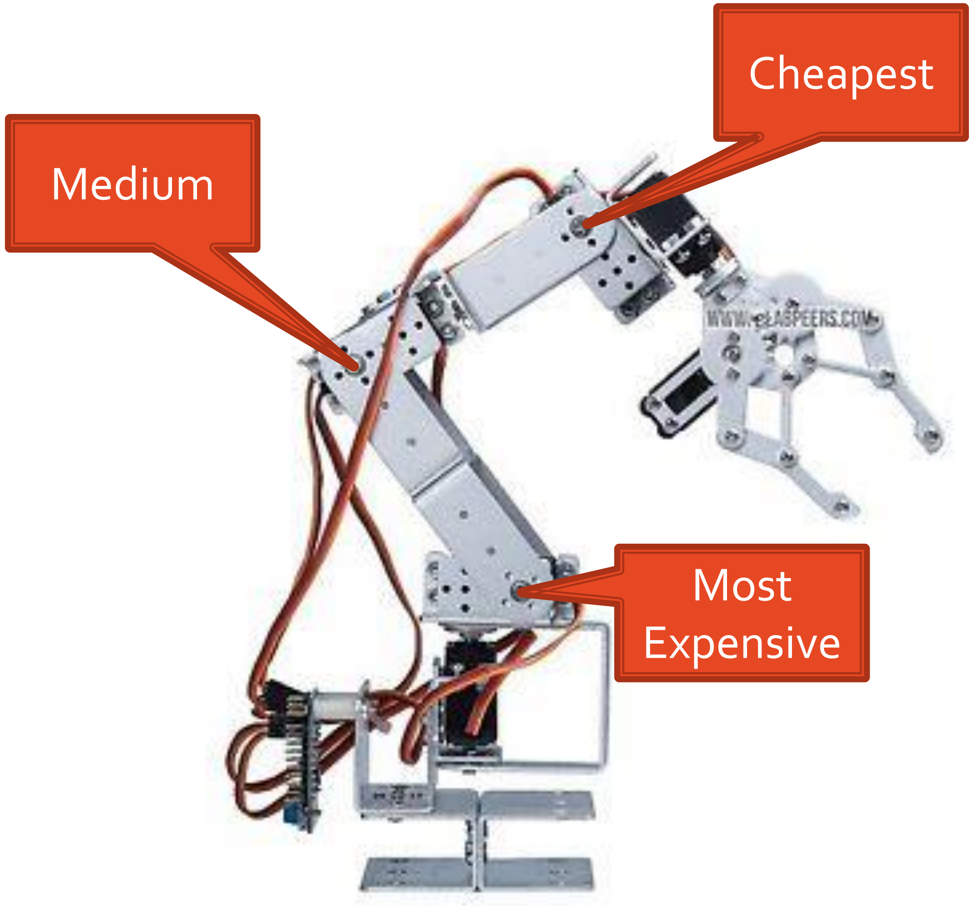
while $OPEN \neq \emptyset$ **do**
    $s \leftarrow \arg\max_{s \in OPEN} \{e(s)\}$
    $OPEN \leftarrow OPEN \setminus \{s\}$
    if $e(s) < E$ **then**
        $E \leftarrow e(s)$
    if ISGOAL(s) **then**
        $G \leftarrow g(s)$
        return
    for each successor s' of s **do**
        if $g(s) + c(s, s') < g(s')$ **then**
            $g(s') \leftarrow g(s) + c(s, s')$
            pred$(s') \leftarrow s$
            if $g(s') + h(s') < G$ **then**
                Insert or update s' in $OPEN$ with key $e(s')$

Epsilon-Convergence



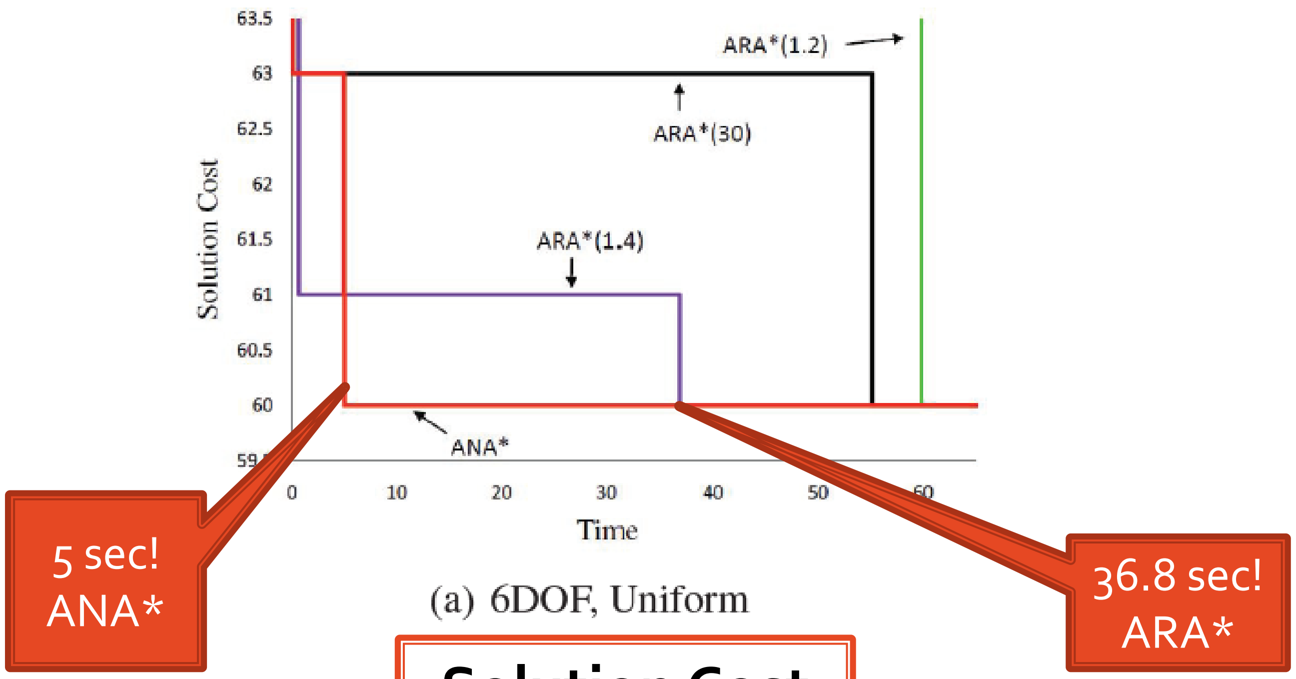
Benchmark Test 1: Robot Arm

- Experiment Setup
  - 6-DOF (and 20-DOF)Arms, fixed base
  - Changing Joint angle closer to base cost more energy
  - Heuristic = Euclidean 2-norm to goal



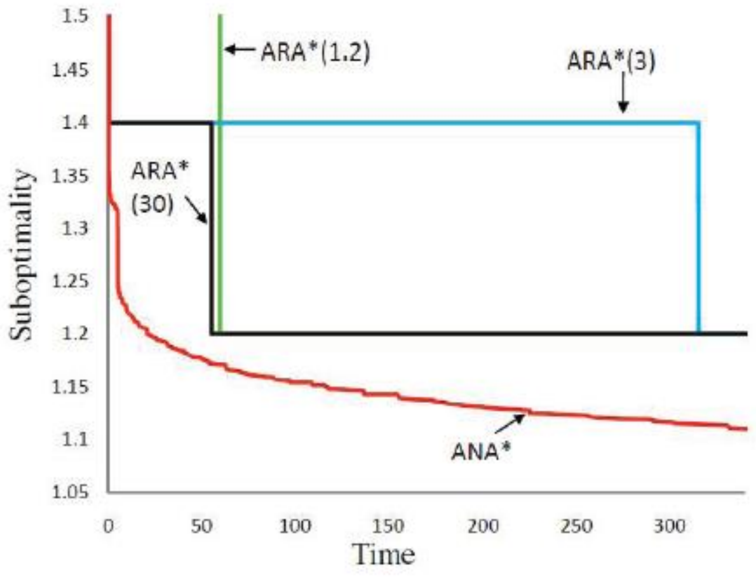
Benchmark Test 1: Robot Arm

$$\Delta\epsilon = 0.2$$



(a) 6DOF, Uniform

Solution Cost

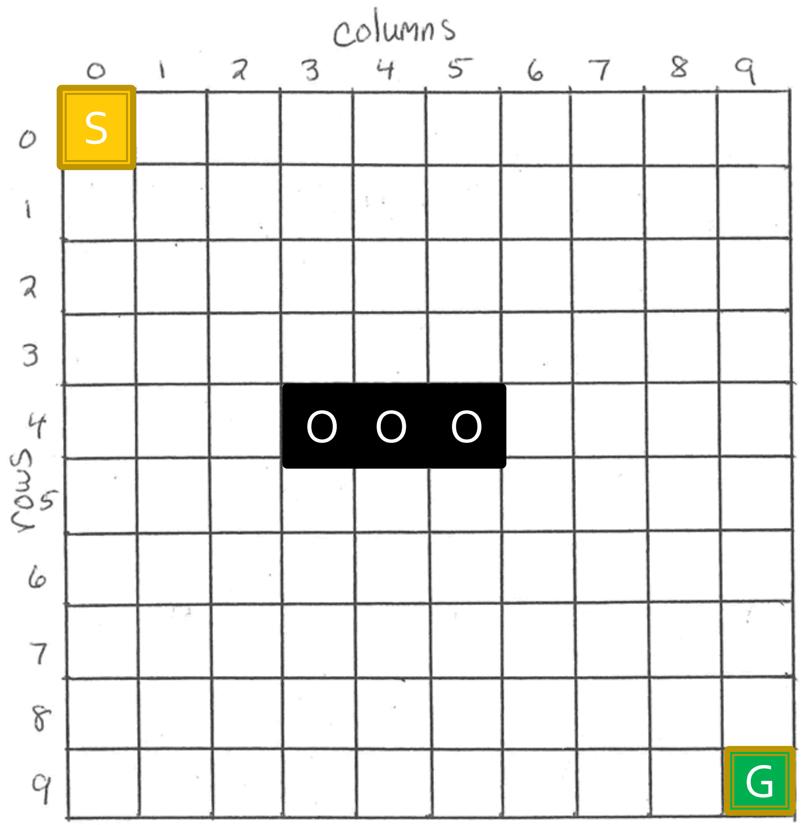


(b) 6DOF, Uniform

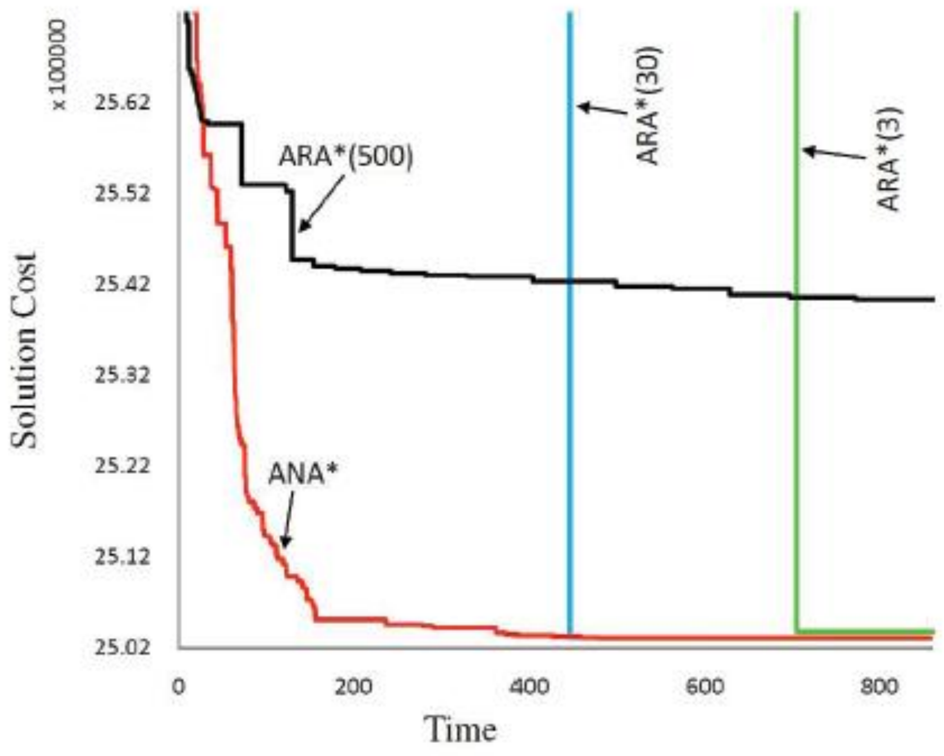
Sub-Optimality (ϵ)

Benchmark Test 2: Gridworld

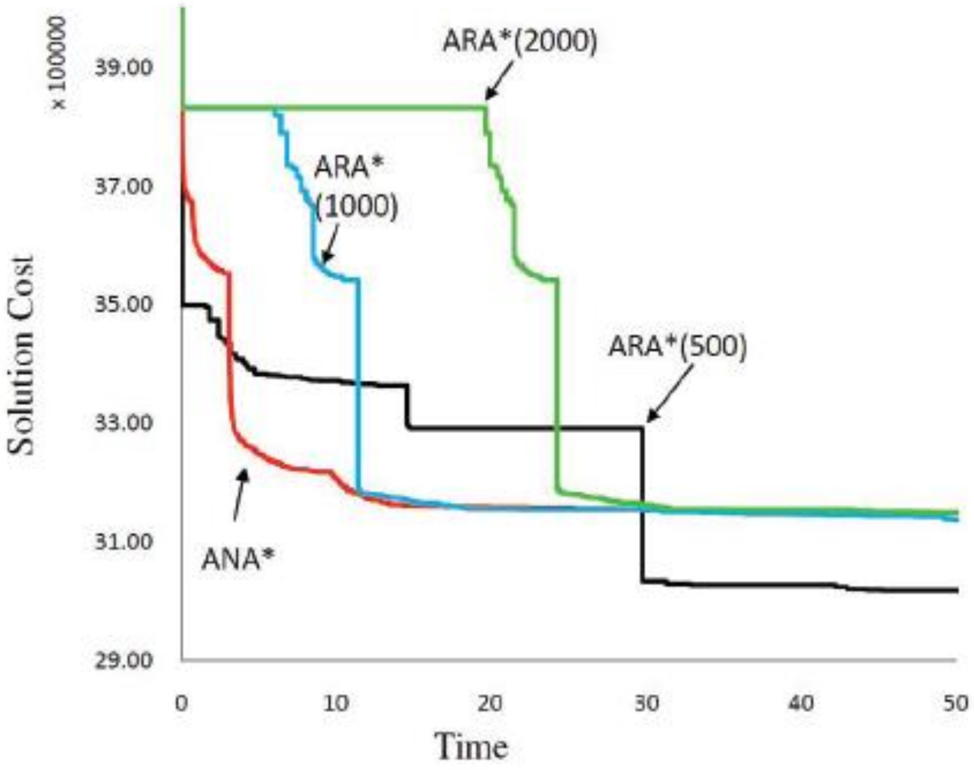
- Experiment Setup
  - 5000 x 5000 4-connected grid
  - Start at top-left corner
  - Goal at right-bottom
  - Obstacles are static



Benchmark Test 2: Gridworld



(a) Without obstacles



(b) With obstacles

Conclusion

- ANA* has superior properties among Anytime-Heuristic Search Algorithms

- ANA* uses dynamic ϵ-choosing mechanism, that removes need for ad-hoc parameter selection

Conclusion

- Advantages:
  - No parameters
  - Maximally greedy for Initial Solution
  - Maximally greedy to Improve Solution
  - Sub-optimality is reduced dynamically
- What ANA* can't do:
  - Not suitable for dynamic environments

Assignment Due Feb 16

- Implement ANA* and test on a grid search problem

ANA*()

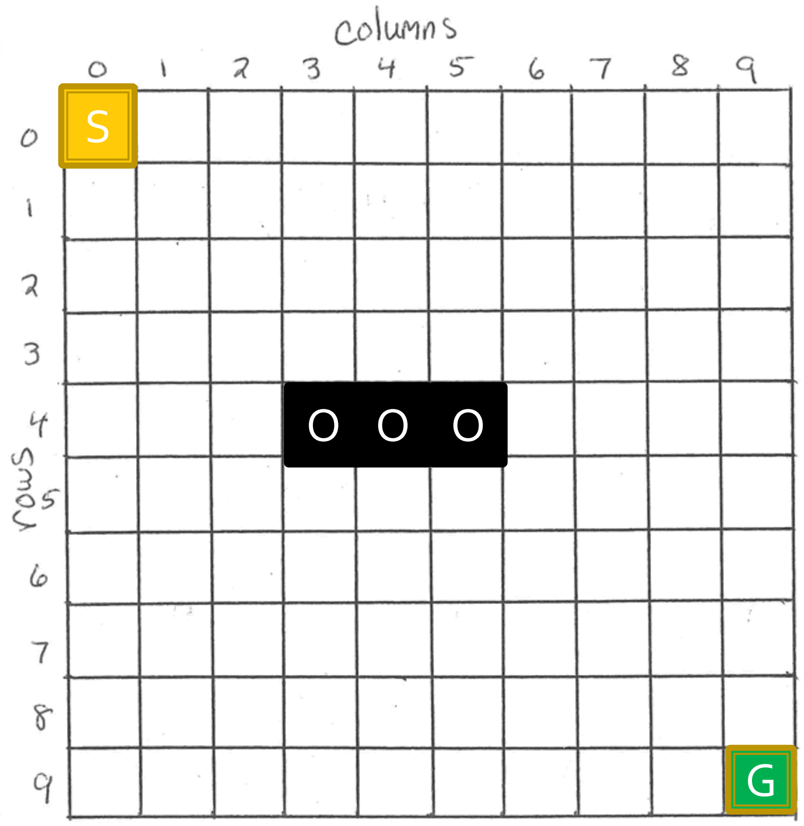
$G \leftarrow \infty$; $E \leftarrow \infty$; $OPEN \leftarrow \emptyset$; $\forall s : g(s) \leftarrow \infty$; $g(s_{\text{start}}) \leftarrow 0$
Insert s_{start} into $OPEN$ with key $e(s_{\text{start}})$
while $OPEN \neq \emptyset$ **do**
    ImproveSolution()
    Report current E-suboptimal solution
    Update keys $e(s)$ in $OPEN$ and prune if $g(s) + h(s) \geq G$

ImproveSolution()

while $OPEN \neq \emptyset$ **do**
    $s \leftarrow \arg\max_{s \in OPEN}\{e(s)\}$
    $OPEN \leftarrow OPEN \setminus \{s\}$
    if $e(s) < E$ **then**
        $E \leftarrow e(s)$
    if IsGoal(s) **then**
        $G \leftarrow g(s)$
        return
    for each successor s' of s **do**
        if $g(s) + c(s, s') < g(s')$ **then**
            $g(s') \leftarrow g(s) + c(s, s')$
            pred$(s') \leftarrow s$
            if $g(s') + h(s') < G$ **then**
                Insert or update s' in $OPEN$ with key $e(s')$

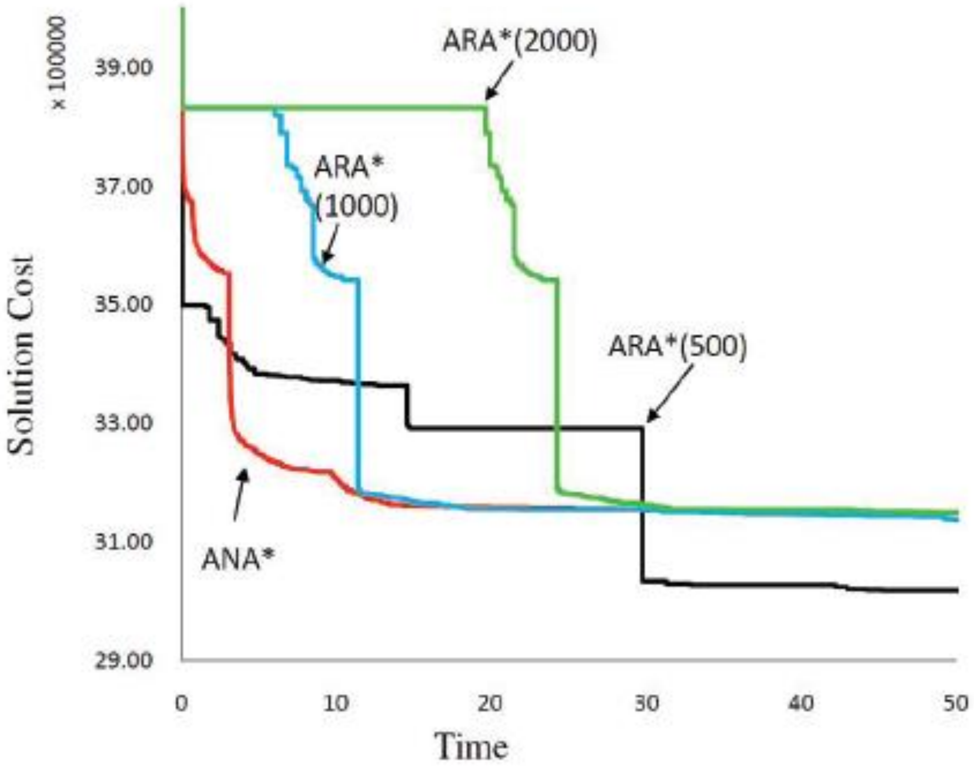
Assignment Due Feb 16

- Experiment Setup
  - 5000 x 5000, 4-connected grid
  - Start = top-left corner, Goal = right-bottom
  - Set your own static obstacles



Assignment Due Feb 16 by noon

- Submission include
  - Python code with problem setup and search using ANA*
  - Pdf report with the figure of solution cost vs Time



(b) With obstacles

Assignment Due Feb 16 by noon

- Refer to the paper on ANA* for more details
  - Van Den Berg, Jur, et al. "ANA*: anytime nonparametric A*." *Proceedings of Twenty-fifth AAAI Conference on Artificial Intelligence (AAAI-11)*. 2011.

- Acknowledgement to **Abhishek Kulkarni**
  - Current WPI graduate student
  - Best literature review presentation in RBE 550, 2017

Important!

- If the assignment description is not clear to you, **please** ask me and/or TA
  - After class
  - During office hour
  - On piazza

- We are willing to help as long as you ask!
  - If you don't ask, we could only assume everything is okay

Student talk
Spatiotemporal state lattice

End