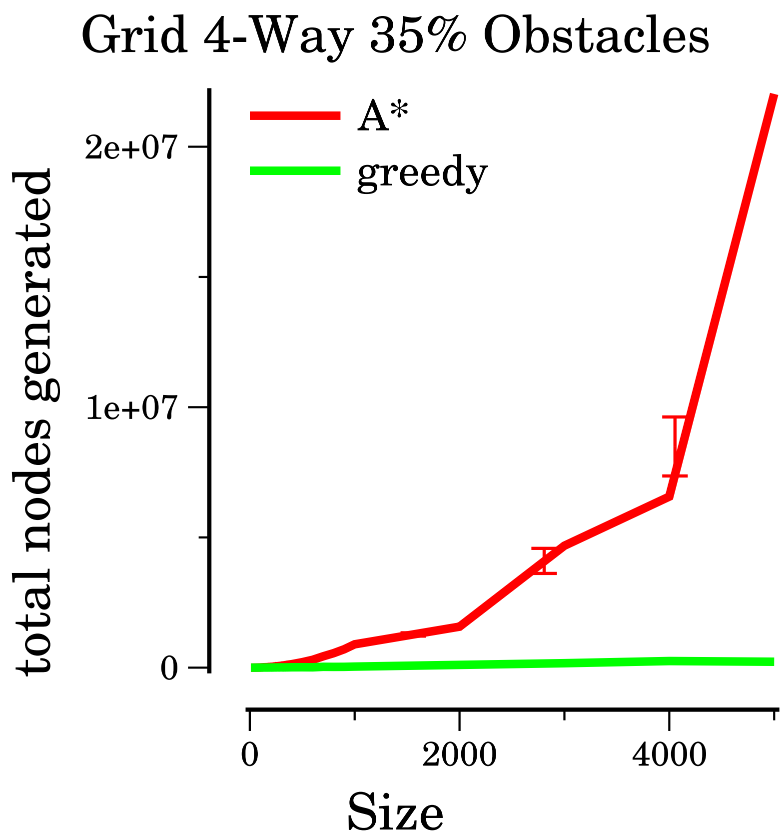
Finding Acceptable Solutions Faster Using Inadmissible Information

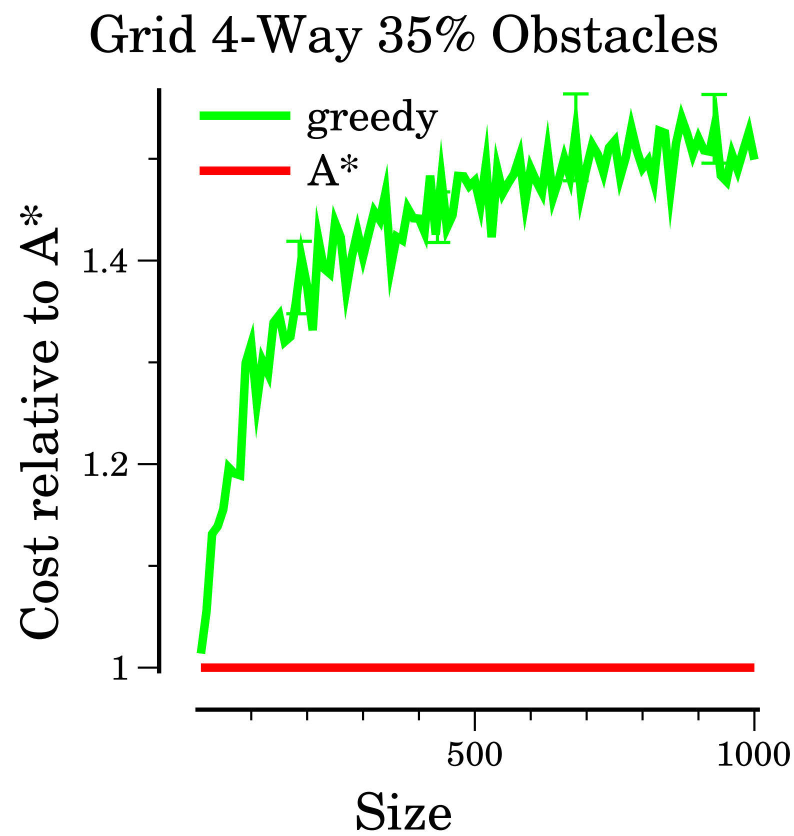
Jordan Thayer & Wheeler Ruml

University of New Hampshire

Supported by DARPA CSSG program (grant HR0011-09-1-0021) and NSF (grant IIS-0812141)

Bounded Suboptimal Heuristic Search





We want speed like this.

We want cost like this.

Bounded Suboptimal Heuristic Search

- Guarantee the solution is within a factor w of optimal.

  Solution is w-admissible

- Find solutions as quickly as you can within the bound.

Bounded Suboptimal Heuristic Search

- Guarantee the solution is within a factor w of optimal.

  Solution is w-admissible

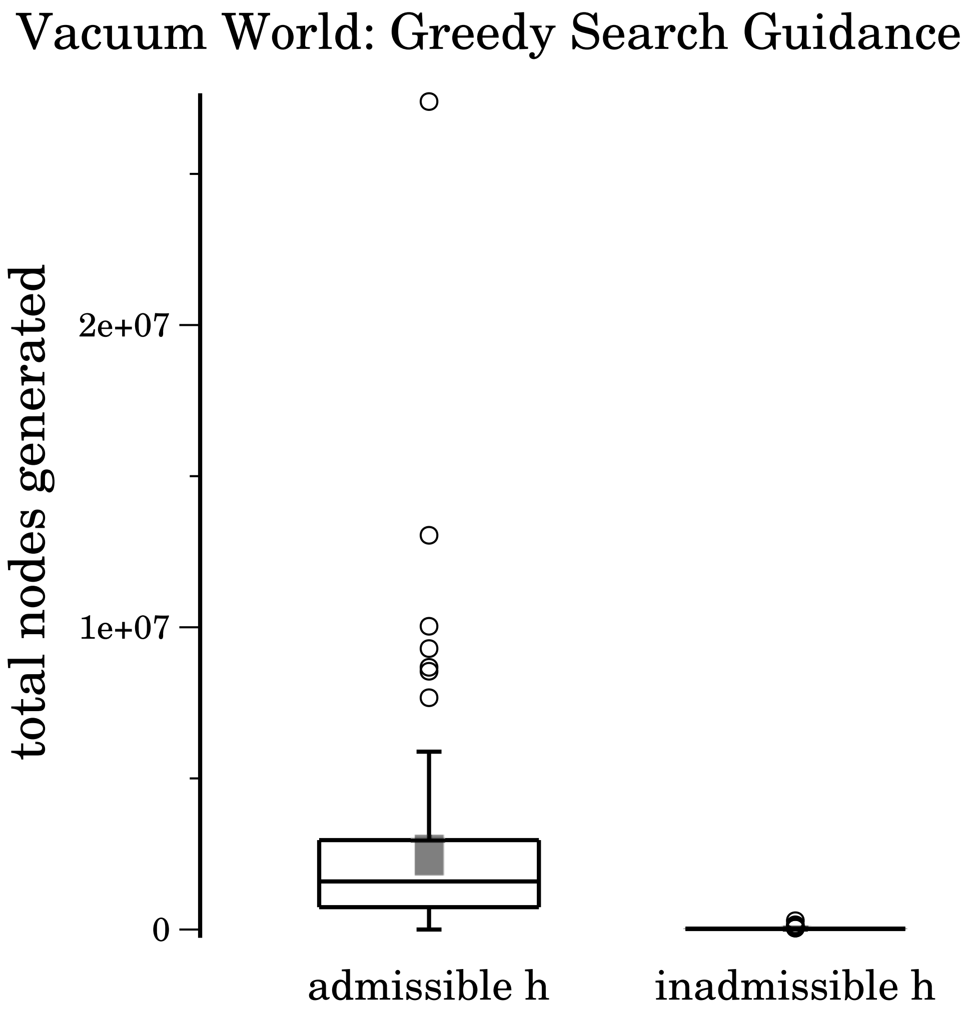
- Find solutions as quickly as you can within the bound.

- Weighted A*
  Pohl, 1970
- Dynamically Weighted A*
  Pohl, 1973
- A^*_ϵ
  Pearl, 1982
- A_ϵ
  Ghallad & Allard, 1983
- AlphA*
  Reese, 1999
- Clamped Adaptive
  Thayer, Ruml, & Bitton 2008
- Optimistic Search
  Thayer & Ruml, 2008
- Revised Dynamically wA*
  Thayer & Ruml, 2009

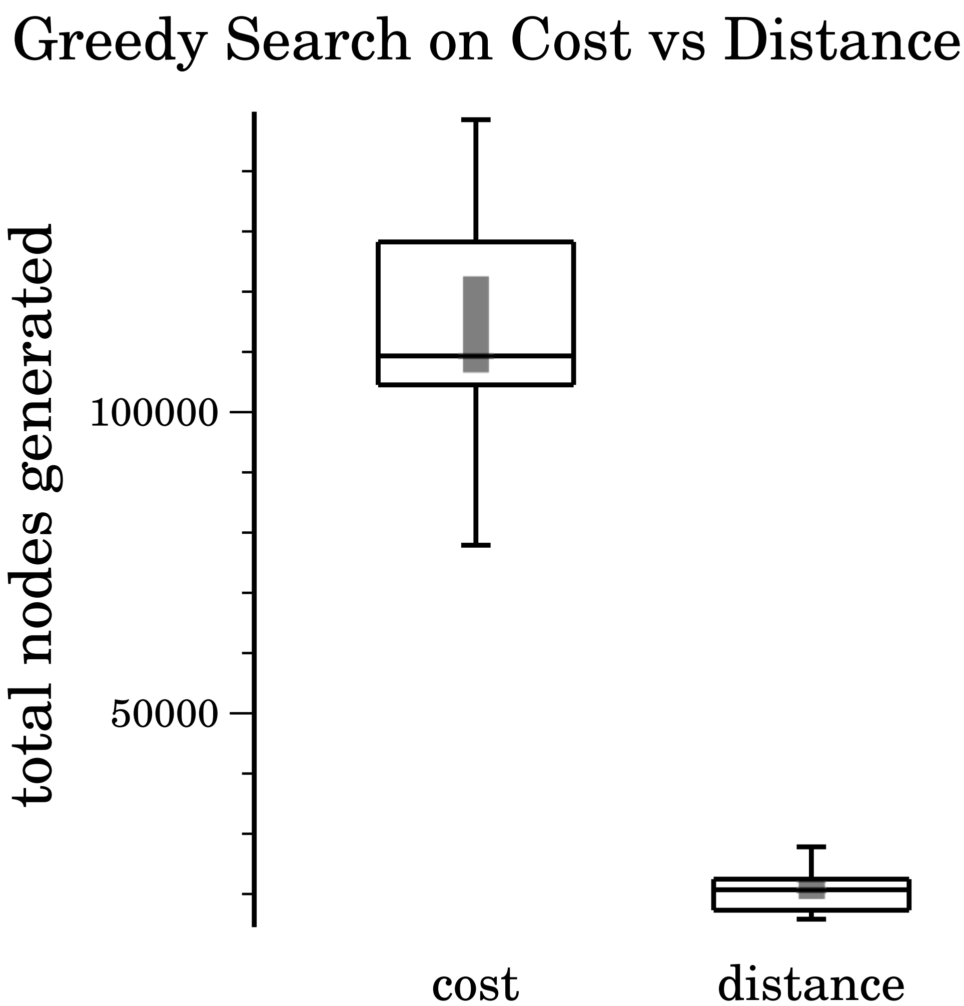
Outline

- Introduce Two Oportunities to Improve Bounded Suboptimal Search
  - Using Inadmissible Heuristics
  - Paying attention to differences in cost and distance
- Present EES, Which Exploits Them
- Show Selected Results

Inadmissible Estimates Outperform Admissible Estimates



Cost And Distance Are Different



We're Ignoring Useful Information

- Inadmissible estimates of cost provide better guidance.

- Search on distance is faster than search on cost.

We're Ignoring Useful Information

- Inadmissible estimates of cost provide better guidance.

  We can't use these without sacrificing bounds.

- Search on distance is faster than search on cost.

  Previous algorithms haven't effectively harnessed d.

We're Ignoring Useful Information

- Inadmissible estimates of cost provide better guidance.

  We can't use these without sacrificing bounds.

- Search on distance is faster than search on cost.

  Previous algorithms haven't effectively harnessed d.

- EES

  uses inadmissible estimates for guidance,
  admissible estimates for bounding

  takes advantage of cost and distance estimates
  without brittle behavior of previous approaches

Explicit Estimation Search

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

Explicit Estimation Search

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$
\begin{aligned}
f_{min} &= \text{node with least } f \\
best_{\widehat{f}} &= \text{node with best estimated cost} \\
best_{\widehat{d}} &= w\text{-admissible node nearest to goal}
\end{aligned}
$$

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned} f_{min} &= \text{node with least } f \\ &= \operatorname*{argmin}_{n \in open} f(n) = g(n) + h(n) \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned} f_{min} &= \text{node with least } f \\ &= \underset{n \in open}{\operatorname{argmin}} f(n) = g(n) + h(n) \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ &= \underset{n \in open}{\operatorname{argmin}} \widehat{f}(n) = g(n) + \widehat{h}(n) \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

Explicit Estimation Search

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned}
f_{min} &= \text{node with least } f \\
&= \operatorname*{argmin}_{n \in open} f(n) = g(n) + h(n) \\
best_{\widehat{f}} &= \text{node with best estimated cost} \\
&= \operatorname*{argmin}_{n \in open} \widehat{f}(n) = g(n) + \widehat{h}(n) \\
best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \\
&= \operatorname*{argmin}_{n \in open \wedge \widehat{f}(n) \leq w \cdot \widehat{f}(best_{\widehat{f}})} \widehat{d}(n)
\end{aligned}$$

Why This Expansion Order?

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Why This Expansion Order?

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Of all the nodes within the bound,
expand the one closest to a goal.

Why This Expansion Order?

Given:

h - An admissible estimate of cost to go
$\widehat{h}$ - A potentially inadmissible estimate of cost to go
$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Ensures $best_{\widehat{d}}$ is a high quality node.

Why This Expansion Order?

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Provides the suboptimality bounds.

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Explicit Estimation Search

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \end{cases}$$

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$
\begin{aligned}
f_{min} &= \text{node with least } f \\
best_{\widehat{f}} &= \text{node with best estimated cost} \\
best_{\widehat{d}} &= w\text{-admissible node nearest to goal}
\end{aligned}
$$

$$
selectNode = \begin{cases} best_{\widehat{d}} & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ & \text{if } \widehat{f}(best_{\widehat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}
$$

Given:

h - An admissible estimate of cost to go

$\widehat{h}$ - A potentially inadmissible estimate of cost to go

$\widehat{d}$ - A potentially inadmissible estimate of distance to go

$\widehat{f}(n) = g(n) + \widehat{h}(n)$

$$\begin{aligned} f_{min} &= \text{node with least } f \\ best_{\widehat{f}} &= \text{node with best estimated cost} \\ best_{\widehat{d}} &= w\text{-admissible node nearest to goal} \end{aligned}$$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if it is within the bound} \\ best_{\widehat{f}} & \text{if it is within the bound, but } best_{\widehat{d}} \text{ isn't} \\ f_{min} & \text{otherwise} \\ & \text{if } \widehat{f}(best_{\widehat{f}}) > w \cdot f(f_{min}) \\ & \wedge \widehat{f}(best_{\widehat{d}}) > w \cdot f(f_{min}) \end{cases}$$

Summary

- Inadmissible estimates of cost provide better guidance.

  We can't use these without sacrificing bounds.

- We can estimate the cost and the distance of a solution.

  Algorithms that use this information perform poorly.

Summary

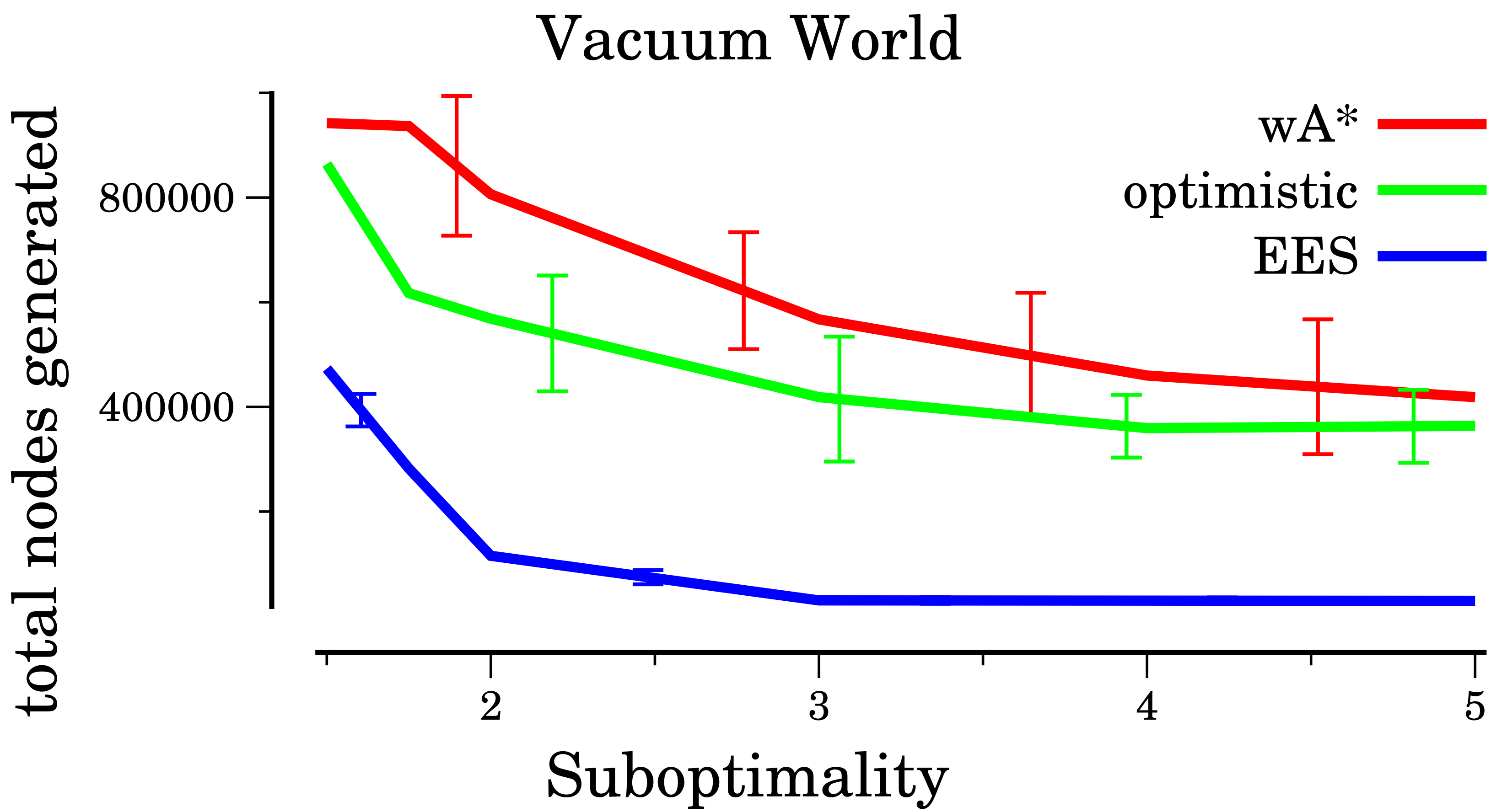
- Inadmissible estimates of cost provide better guidance.

  EES can use these without sacrificing quality bounds.

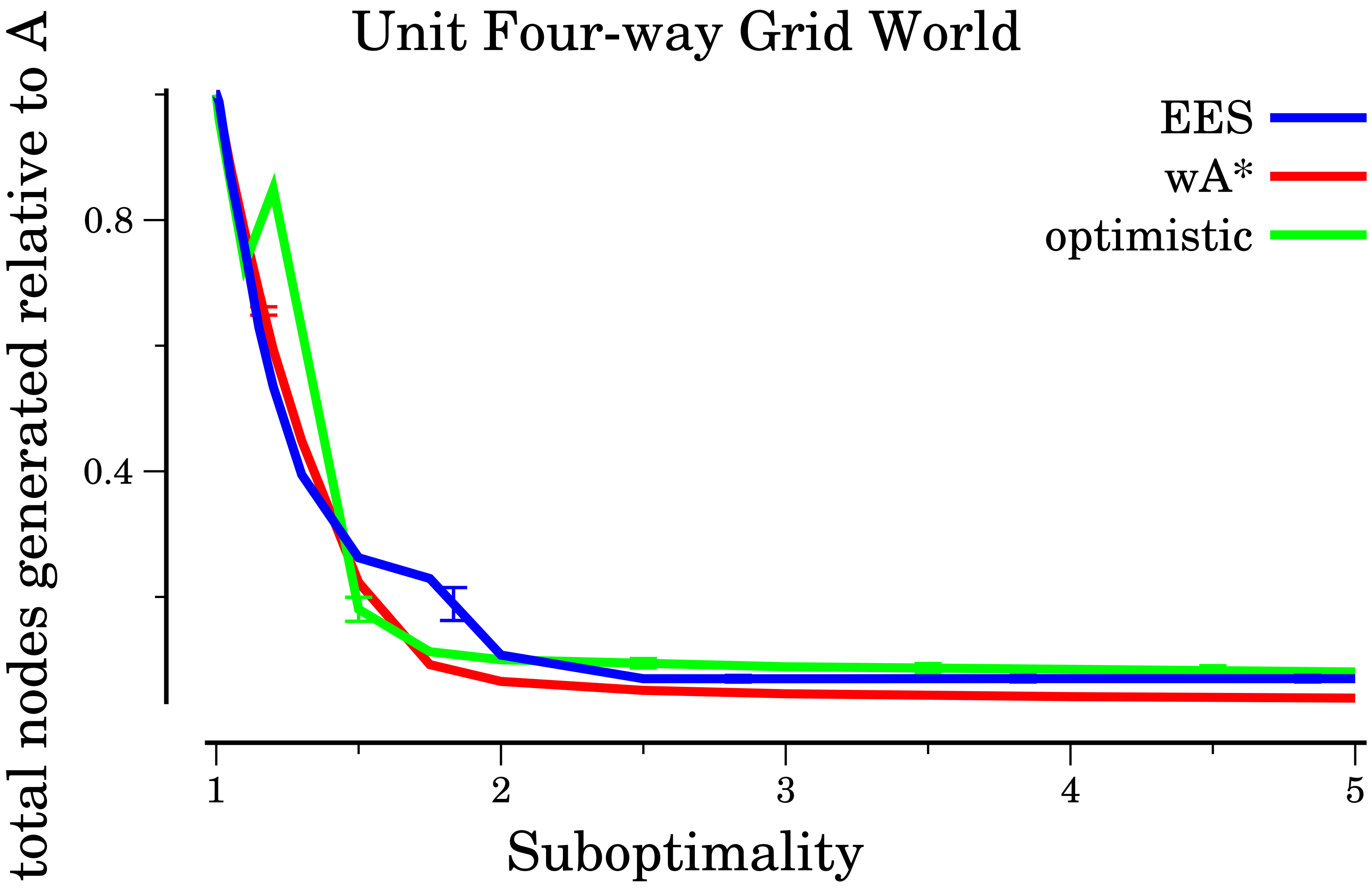
- We can estimate the cost and the distance of a solution.

  EES avoids the pitfalls of previous approaches.

Vacuums: Inadmissible Heuristics



Grids


Unit Four-way Grid World
total nodes generated relative to A
0.8
0.4
1
2
3
4
5
Suboptimality
EES
wA*
optimistic

Performance In Aggregate: CPU Relative to EES

Bound	1.5	1.75	2.	3.	4.	5.
optimistic	1.6	1.5	1.6	2.1	2.4	2.1
wA*	4.1	3.4	2.8	3.7	3.4	2.4
skeptical	2.6	4.7	4.9	5.1	11.4	13
A^*_ϵ	50	44	28	1.8	1.1	**0.6**
Clamped	8.3	10	11	67	85	85
AlphA*	120	140	180	280	300	310
rdwA*	370	310	240	100	84	120
A_ϵ	910	850	680	620	590	610

Numbers are average slowdown per domain, averaged across eight domains:
TSP (two variants), Grid Navigation (two variants), Dynamic Robot Path Planning, Vacuum Planning, Sliding Tiles Problem

General Performance: Nodes Generated Relative to EES

Bound	1.5	1.75	2.	3.	4.	5.
optimistic	3.1	2.4	2.5	3.3	3.4	3.2
wA*	6.6	5.5	4.5	5.5	5.0	4.0
skeptical	3.2	3.0	2.8	3.8	11	15
A^*_ϵ	58	44	17	1.8	1.1	**0.8**
Clamped	6.8	5.6	7.1	76	95	97
AlphA*	1.2	1.5	2.2	4.4	5.6	5.7
rdwA*	180	170	150	86	78	160
A_ϵ	1500	1400	1100	990	910	970

Numbers are average increase in nodes generated per domain, averaged across eight domains:
TSP (two variants), Grid Navigation (two variants), Dynamic Robot Path Planning, Vacuum Planning, Sliding Tiles Problem

General Performance: Algorithm Rankings (CPU)

	1^{st}	2^{nd}	3^{rd}	4^{th}	$> 4^{th}$
EES	2	3	3	0	**0**
Optimistic	3	1	2	1	1
Skeptical	1	3	1	0	3
A^*_ϵ	2	0	1	1	4
wA*	0	1	1	4	2
A_ϵ	0	0	0	0	8
AlphA*	0	0	0	0	8
Clamped	0	0	0	0	8
rdwA*	0	0	0	0	8

Rankings by CPU time consumed

Average Performance: Algorithm Rankings (Nodes)

	1^{st}	2^{nd}	3^{rd}	4^{th}	$> 4^{th}$
EES	5	3	0	0	**0**
Optimistic	1	0	4	1	2
Skeptical	0	2	3	1	2
A^*_ϵ	2	1	0	1	4
wA*	2	0	0	3	3
A_ϵ	0	0	0	0	8
AlphA*	0	0	0	0	8
Clamped	0	0	0	0	8
rdwA*	0	0	0	0	8

Rankings by nodes generated

Conclusions

- We can finally use inadmissible heuristics.
- We can benefit from using cost and distance information.
- EES provides

  robust behavior on a wide range of benchmarks.

  state of the art performance in several domains.

Additional Slides

Proof of Bounded Suboptimality

Assume:
$\widehat{f}(n) \geq f(n)$ and $\widehat{h}(goal) = 0$
$f(n) = \widehat{f}(n) = g(n)$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if } \widehat{f}(best_{\widehat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}$$

Proof of Bounded Suboptimality

Assume:

$\widehat{f}(n) \geq f(n)$ and $\widehat{h}(goal) = 0$

$f(n) = \widehat{f}(n) = g(n)$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if } \widehat{f}(best_{\widehat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}$$

$$w \cdot f(opt) \geq w \cdot f(f_{min})$$

$$w \cdot f(f_{min}) \geq \widehat{d}(best_{\widehat{d}})$$

$$\widehat{f}(best_{\widehat{d}}) \geq f(best_{\widehat{d}})$$

$$f(best_{\widehat{d}}) \geq g(best_{\widehat{d}})$$

Proof of Bounded Suboptimality

Assume:

$\widehat{f}(n) \geq f(n)$ and $\widehat{h}(goal) = 0$

$f(n) = \widehat{f}(n) = g(n)$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if } \widehat{f}(best_{\widehat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}$$

$$w \cdot f(opt) \geq w \cdot f(f_{min})$$

$$w \cdot f(f_{min}) \geq \widehat{f}(best_{\widehat{f}})$$

$$\widehat{f}(best_{\widehat{f}}) \geq f(best_{\widehat{f}})$$

$$f(best_{\widehat{f}}) \geq g(best_{\widehat{f}})$$

Proof of Bounded Suboptimality

Assume:

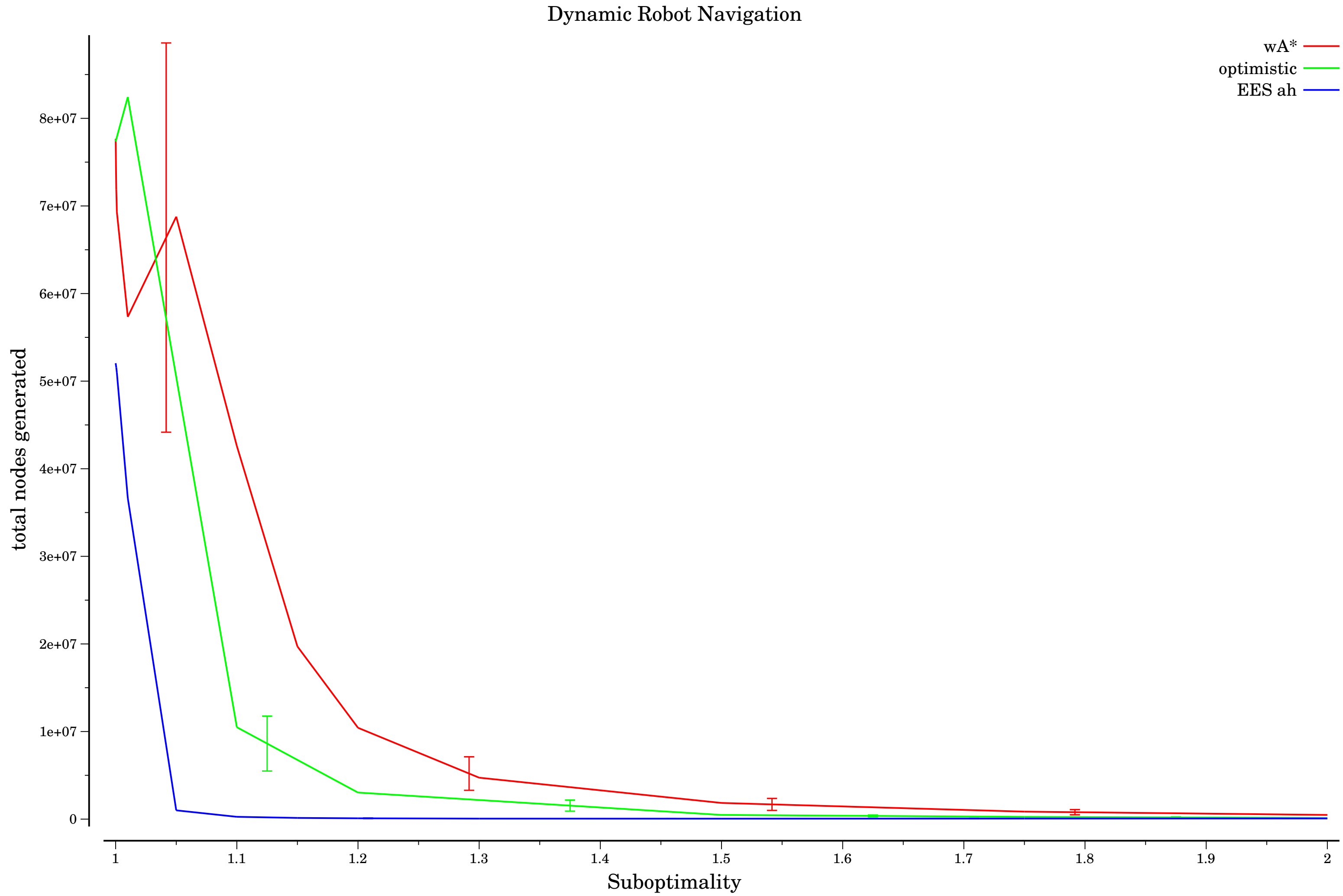
$\widehat{f}(n) \geq f(n)$ and $\widehat{h}(goal) = 0$

$f(n) = \widehat{f}(n) = g(n)$

$$selectNode = \begin{cases} best_{\widehat{d}} & \text{if } \widehat{f}(best_{\widehat{d}}) \leq w \cdot f(f_{min}) \\ best_{\widehat{f}} & \text{if } \widehat{f}(best_{\widehat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}$$

$$w \cdot f(opt) \geq \ w \cdot f(f_{min})$$

Robot Navigation: Inadmissible Heuristics



Strict vs. Loose Approaches to Quality Bounds

Loose: Optimistic Search

- Run weighted A^* with weight $(bound - 1) \cdot 2 + 1$
- Expand node with lowest f value after a solution is found.

  Continue until $w \cdot f_{min} > f(sol)$

  This 'clean up' guarantees solution quality.

Strict: EES

$$selectNode = \begin{cases} best_{\hat{d}} & \text{if } \widehat{f}(best_{\hat{d}}) \leq w \cdot f(f_{min}) \\ best_{\hat{f}} & \text{if } \widehat{f}(best_{\hat{f}}) \leq w \cdot f(f_{min}) \\ f_{min} & \text{otherwise} \end{cases}$$